HAL open science

Information Extraction for Editing and By-Example Synthesis in Expressive Rendering

Pierre-Edouard Landes

To cite this version:

Pierre-Edouard Landes. Information Extraction for Editing and By-Example Synthesis in Expressive Rendering. Mathematics [math]. Université Joseph-Fourier - Grenoble I, 2011. English. NNT : . tel-00637651v1

HAL Id: tel-00637651

https://theses.hal.science/tel-00637651v1

Submitted on 2 Nov 2011 (v1), last revised 10 Apr 2012 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques & Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Pierre-Édouard LANDES

Thèse dirigée par **François X. SILLION**
et codirigée par **Cyril SOLER**

préparée au sein du **Laboratoire Jean KUNTZMANN**
et de l'**École Doctorale Mathématiques, Sciences & Technologies de l'Information, Informatique**

Extraction d'information pour l'édition et la synthèse par l'exemple en rendu expressif

Thèse soutenue publiquement le **17 février 2011**,
devant le jury composé de :

M. Karol MYSZKOWSKI
Professeur, *Max-Planck-Institut für Informatik*, Président
M. Doug DECARLO
Professeur, *Rutgers University*, Rapporteur
M. Pascal GUITTON
Professeur, Université de Bordeaux I, Rapporteur
M. François X. SILLION
Directeur de recherche, INRIA Grenoble - Rhône-Alpes, Directeur de thèse
M. Cyril SOLER
Chargé de recherche, INRIA Grenoble - Rhône-Alpes, Co-Directeur de thèse

GRENOBLE UNIVERSITIES

DOCTORAL SCHOOL MSTII

MATHEMATIQUES, SCIENCES ET TECHNOLOGIES DE L'INFORMATION, INFORMATIQUE

P H D T H E S I S

to obtain the title of

PhD of Computer Science

of the University of Grenoble

Specialty : MATHEMATICS & COMPUTER SCIENCE

defended by

Pierre-Edouard LANDES

Information Extraction for Editing and By-Example Synthesis in Expressive Rendering

Extraction d'information pour l'ï¿½dition et la synthï¿½se par l'exemple en rendu expressif

prepared at INRIA Grenoble - Rhône-Alpes, ARTIS team

Jury

President	Karol MYSZKOWSKI	-	Max-Planck-Institut fr Informatik
Reviewers	Doug DECARLO	-	Rutgers University
	Pascal GUITTON	-	University of Bordeaux 1
Advisors	François-X. SILLION	-	INRIA Grenoble - Rhône-Alpes
	Cyril SOLER	-	INRIA Grenoble - Rhône-Alpes

Contents

1 Introduction **1**

I Texture Synthesis by Example **7**

2 State of the Art in Raster Texture Analysis and Synthesis **15**
2.1 A Glimpse at Procedural Texturing . . . 15
2.2 Example-Based Texture Generation . . . 17
2.2.1 Texture Characterization Using Tonal Statistics . . . 17
2.2.2 Textures as Markov Random Field Samples . . . 20
2.2.3 Constrained Texture Models . . . 23
2.2.4 Texture Synthesis by Multi-Scale Statistics Transfer . . . 24
2.2.5 Statistical Transfer for Example-Driven Procedural Texturing . . . 26
2.2.6 Synthesis by Non-Parametric Sampling . . . 27
2.2.6.1 Non-Parametric Pixel-Based Synthesis . . . 28
2.2.6.2 From Local to Global Optimization . . . 30
2.2.6.3 Breaking the Output Pixels' Cyclic Dependencies . . . 30
2.2.6.4 Coercing Pixel Matching Through Image Coherence . . . 32
2.2.6.5 Non-Parametric Patch-Based Synthesis . . . 33
2.2.7 From Pixel Patches to Texture Tessellation . . . 37
2.2.8 Automation *versus* Artistic Control . . . 39
2.2.8.1 Constrained Synthesis . . . 40
2.2.8.2 User-Guided Synthesis . . . 40
2.2.8.3 Altered Synthesis . . . 42
2.2.8.4 Consistent Texture Editing . . . 43

3 Pattern Recognition for the Automatic Synthesis of Raster Shape Arrangements **45**
3.1 Introduction . . . 45
3.1.1 Intuition Behind Our Texture Analysis Approach . . . 45
3.1.2 Overview of Our Shape Extraction Technique . . . 45
3.2 Constructing the Region Similarity Map . . . 46
3.2.1 Visual Similarity Detection . . . 46
3.2.2 Repetitive Shape Extraction . . . 47
3.3 Recovering Pattern Classes . . . 49
3.3.1 Theoretical Analysis . . . 49
3.3.2 Proposed Heuristic . . . 50
3.3.3 Pattern Extraction . . . 51
3.4 Controlled Shape Pasting for Raster Arrangement Generation . . . 51
3.4.1 Re-Synthesis Results . . . 51
3.4.2 Discussion . . . 53
3.4.3 Comparison to Related Work . . . 53
3.4.4 Shortcomings . . . 55
3.4.5 Computation Costs . . . 55
3.5 Conclusions . . . 56

4 Spatial Modeling for the Automatic Synthesis of Vector Element Arrangements **57**
4.1 Introduction . . . 57
4.1.1 Related Work . . . 59
4.1.1.1 Pixel-based Texture Synthesis . . . 59
4.1.1.2 Patch-based Texture Synthesis . . . 59
4.1.1.3 Extension to Vector Primitives . . . 60
4.1.1.4 Line Appearance Encoding . . . 61
4.1.2 Contributions . . . 61
4.2 Appearance-based Element Categorization . . . 61
4.2.1 Stroke-based Element Description . . . 62
4.2.2 Detection of Meaningful Feature Modes . . . 62
4.3 Statistical Arrangement Modeling . . . 64
4.3.1 Multi-type Point Process Model . . . 64
4.3.2 Estimation of the Model Parameters . . . 65
4.3.2.1 Hard-core Distances $h_{m,m'}$ Between Category Pairs . . . 65
4.3.2.2 Trend Distances $r_{m,m'}$ Between Category Pairs . . . 66
4.3.2.3 Interaction Strengths $\gamma_{m,m'}$ and Category Occurrence Probabilities d_m . . . 66
4.3.3 Synthesis by Markov chain Monte-Carlo . . . 66
4.4 Results and Discussion . . . 67
4.4.1 Experimental Results . . . 67
4.4.2 Comparisons With Related Work . . . 69
4.4.3 Limitations and Future Work . . . 71
4.5 Conclusion . . . 72
4.6 Future Work and Perspectives . . . 73
4.6.1 Improving the Arrangement Analysis . . . 73
4.6.2 Improving Synthesis . . . 74

II Perceptually-Accurate Image Desaturation and Enhancement **77**

5 State of the Art in Grey-Scale Conversion **81**
5.1 Mapping Colors to Grey Values . . . 81
5.2 Grey-Scale Conversion by the Integration of the Color Contrasts . . . 83
5.3 Color-to-Texture Approaches . . . 85
5.4 Extension to Videos . . . 86

6 Apparent Grey-Scale, A Fast Conversion for Images and Videos **87**
6.1 Influence of Chroma on our Perception of Brightness . . . 87
6.1.1 Defining of the Helmholtz-Kohlrausch Effect . . . 87
6.1.2 Unveiling the Helmholtz-Kohlrausch Effect . . . 88
6.1.3 Explaining the Helmholtz-Kohlrausch Effect . . . 89
6.1.4 Overcoming the Helmholtz-Kohlrausch Effect . . . 90
6.1.5 Embedding the Helmholtz-Kohlrausch Effect in our Grey-Scale Conversion . . . 93
6.2 Chromatic Edges to the Rescue . . . 95
6.2.1 The Chicken and the Egg: Contrasts and Contours . . . 97
6.2.1.1 Global Contrasts for Enhancing Range . . . 98
6.2.1.2 Local Contrasts for Enhancing Contours . . . 99
6.2.1.3 Contours for Enhancing Global Contrasts . . . 101
6.2.2 Restoring Lost Contrasts through Illusory Brightness . . . 106
6.3 Additional Results and Discussion . . . 111
6.4 Conclusions . . . 119

III Artistic Line Rendering by Example 123

7 State of the Art in Line Rendering 133
7.1 Tracing Lines in 2d Image Space . . . 133
7.1.1 Exploiting the Projected Scene Depth . . . 133
7.1.2 Processing more General Geometric Information . . . 134
7.1.3 Finding Edges in Shaded Images . . . 136
7.1.4 Image-Space Approaches: The Pros and Cons . . . 137
7.2 Tracking Lines in 3d Object Space . . . 140
7.2.1 Automatic Detection of View-Independent Lines . . . 140
7.2.1.1 Parabolic Curves . . . 140
7.2.1.2 Ridges and valleys . . . 141
7.2.1.3 Demarcating Curves . . . 143
7.2.1.4 Relief Edges . . . 144
7.2.2 Automatic Detection of View-Dependent Lines . . . 144
7.2.2.1 Silhouette and Contour Curves . . . 146
7.2.2.2 Tweaking the Vantage Point . . . 150
7.2.2.3 Towards a View-Aware Version of Differential Geometry . . . 153
7.2.2.4 Capturing Illumination Variations . . . 154
7.2.3 Object-Space Approaches: The Pros and Cons . . . 155

8 Machine Learning for Interactive Line Rendering 157
8.1 Remaining Challenges in Line Rendering . . . 157
8.1.1 Automatic Line Drawings: Could Less be More? . . . 157
8.1.2 Automatic Line Drawings: a Visual Comparison . . . 159
8.1.3 Putting the Artist Back into the Loop . . . 164
8.1.4 Our Proposal . . . 165
8.2 Line Drawing Creation as a Predictive Process . . . 166
8.2.1 Describing Strokes by their Geometrical Footprint . . . 166
8.2.1.1 Selection of the Stroke Features . . . 167
8.2.1.2 Retrieval of the Stroke Features . . . 169
8.2.2 Learning the User's Style with ADABOOST . . . 173
8.2.3 Classification Results and Validation of our Approach . . . 177
8.2.4 Guiding the Creation of New Strokes by the Inferred User's Style . . . 186
8.3 Ongoing Research and Closing Remarks . . . 190

9 Conclusion 195
9.1 Contributions and Perspectives . . . 195
9.1.1 Pattern Recognition for the Automatic Synthesis of Raster Shape Arrangements . . . 196
9.1.2 Spatial Modeling for the Automatic Synthesis of Vector Element Arrangements . . . 196
9.1.3 Apparent Grey-Scale: A Fast Conversion for Images and Videos . . . 197
9.1.4 Machine Learning for Interactive Line Rendering . . . 198
9.2 Discussions . . . 199
9.2.1 From Example-Based Synthesis to Artistic Style Capture . . . 199
9.2.2 Augmenting Rather than Automating . . . 201
9.2.3 "It's Clever, But is it Art?" . . . 202

Appendices 205

A Survey of Color Specification Systems **207**
A.1 When Light Meets the Eye... 207
A.2 Characterization of the Eye's Brightness Responsivity 208
A.3 Exploiting Experimental Perceived Brightness 209
A.4 Specification of Colors and Principal Color Spaces 211
A.4.1 Color Mixing Models 211
A.4.1.1 Additive Color Mixing and Grassmann's Laws 211
A.4.1.2 CIE 1931 RGB Specification System 211
A.4.1.3 CIE 1931 XYZ Specification System 212
A.4.1.4 Refining upon the CIE 1931 XYZ Specification 214
A.4.2 Color Appearance Models 215
A.4.2.1 Early Stages of Colorimetry and First Models 215
A.4.2.2 Making it Complete while Keeping it Simple 215
A.4.2.3 About the Importance Attached to Perception 218
A.4.3 Towards Perceptual Uniformness 219
A.4.3.1 Munsell Color System 219
A.4.3.2 CIE Uniform Color Spaces 220
A.4.4 Towards Aesthetical Uniformness 224
A.4.5 Surround Effects and Eye Adaptation 226

B Differential Geometry: Concepts and Notations **229**
B.1 A Detour to Flatland 229
B.2 Back to Sphereland 230
B.3 Differential Geometry in Practice 233
B.3.1 Curve Sampling 233
B.3.2 Patch-Fitting 234
B.3.3 Averaging 235

C French Summary **237**

Bibliography **253**

CHAPTER 1

Introduction

"Taming the Computer's Appealing Transcendental Charm"[1]

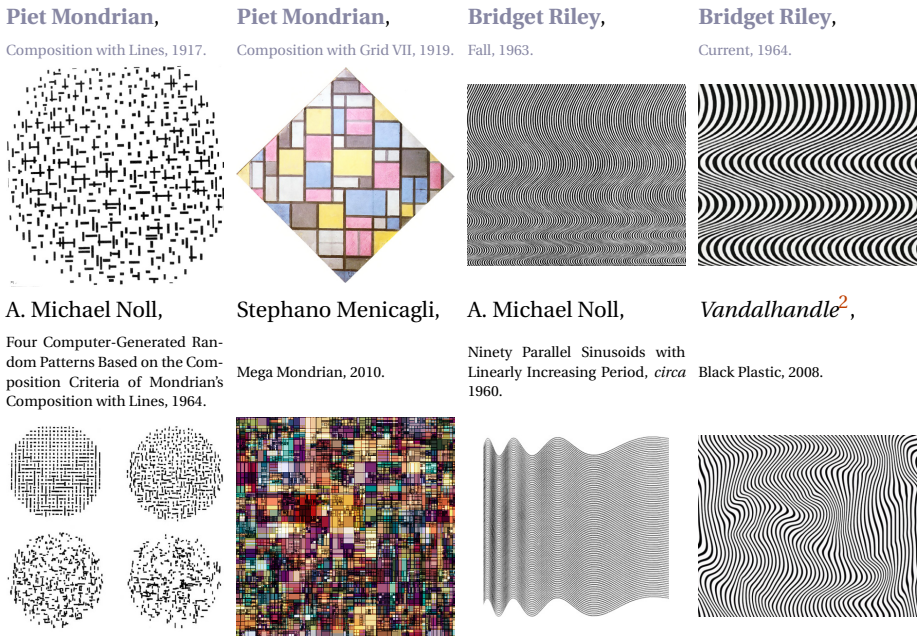
The time when the digital artist community mostly comprised inspired engineers and playful computer scientists sure seems far away. Also long gone is the time when computers were reserved to a limited *intelligentsia* accessing the resources of university laboratories. Who could have thought that the leap from Noll's computer-assisted imitations of Mondrian's or Riley's paintings to the myriad of digital creations from nowadays' online artist communities would take less than half a century?

Looking back at the advent of computer-assisted artistic creations is all the more fascinating from the perspective of today's habits where computers have become such malleable, ubiquitous tools that resorting to them is almost like a reflex. However, back in the sixties, mentalities were certainly not as computer-oriented. The recourse to computers for the production of images of sole aesthetical value was an artistic commitment of isolated individuals such as former *Bell Labs* researcher A. Michael Noll whose experiments constitute the earliest forms of *digital art*. But the progressive penetration of computers in the artistic production is nowhere limited to visual art, and even back at its beginnings, concerned other art forms such as music, poetry, dance, sculpture and architecture. Only after the resounding success of the *Cybernetic Serendipity* (the first art exhibition to ever incorporate computer-generated artworks) at the London Institute of Contemporary Arts in 1968 did such creations start enjoying recognition among the artists' community, entering art galleries and museums, and being rated on the art market.

But the enthusiastic agitation surrounding computers was mostly fueled by the vision of the machine more as an actor, than a mere tool. Hiroshi Kawano notably wrote in 1975 that "computer art is the art of computer as

[1] Extract from the Manifest of the Computer Technic Group, a group of Japanese art students formed in 1966 that pioneered the use of computers in art: *"We will tame the computer's appealing transcendental charm and restrain it from serving established power. This stance is the way to solve complicated problems in the machine society."*

[2] *DeviantArt* pseudonym.



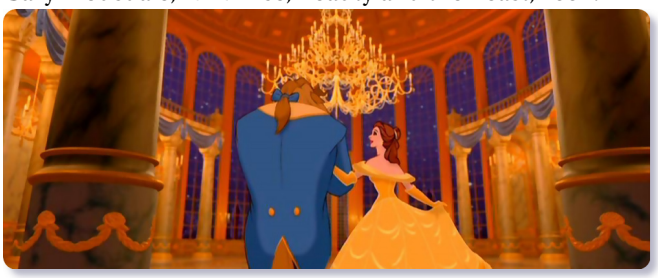
1.1 From Earliest to Newest Forms of Digital Art. Sporting clear lines and vivid solid colors, often tricking our mind with the repetition of geometric shapes, artworks from Cubist painters and Opt artists lend themselves particularly well to being represented through algorithmic procedures, and soon became the prime targets for computer scientists who managed via their programs to create comparable visuals.

an artificial intelligence", and claimed that "therefore, as long as art as an algorithmic procedure, a computer should have a its own artistic behavior". This concept of the simulation by the computer of human art still stimulates the production of contemporary artists, but for a vast majority, has been superseded by the apprehension of the computer as an apparatus whose memory and processing skills facilitate the act of *production*, but does not encroach the act of *conception*.

Looking Back at Today

Leaving the intellectual appeal of computers as artificially intelligent and creative forms, we can still only admire the almost indispensable role computers now fill in almost all stages of production. The fact such a reversal of situation occurred in the lapse of time of several decades is also quite astonishing. From toys for visual experimentation to guarantors of the feasibility and financial viability of most modern projects, computers changed status, and are almost systematically made use of for imagery generation or enhancement.

Gary Trousdale, Kirk Wise, Beauty and the Beast, 1991.



May it be in movies, animations or illustrations, computers made their way in most forms of traditional media, progressively succeeding in not only to embracing but extending them. Indeed, while they represented a valuable help via their automation of the most tedious tasks invoked by the creation of complex imagery, it would be unfair and inaccurate to restrict the impact of computers on traditional techniques to the sole removal of the irreversibility of the gestures of artists over their physical creations (notably with the addition of the *undo* and *save* key features). More than that, the digitalization of traditional techniques urges artists to confront themselves to the frontiers of these new virtual media, domains where neither visual intricacy nor execution complexity represents a bound. In the case of cell animations, computers made possible the automation of the inking and coloring processes, and provide good first guesses for the creation of in-between frames. But they also lifted some of the constraints imposed by the physical medium. Computer-assisted animations can now boast an infinite number of animation layers [3], smooth camera motions worthy of live-action movies, and exquisitely detailed and lit backgrounds. The famous ballroom scene from Disney's 1991 classics *The Beauty and the Beast* ranks among the first harmonious alliances of computer-generated elements with hand-drawn characters.

But the most striking aspect of the computer revolution reveals itself in the cinematographic industry, and is all the more palpable in recent years with the release of high-budget productions putting emphasis on their computer-crafted effects. Either for the creation from scratch of alternative universes reaching stunning levels of believability (*cf.* Figure 1.2), or for the embellishment of reality with subtle story-driven touches (*cf.* Figure 1.3), computer-based visual effects now take the center stage in the production of an ever-increasing number of movies.

A distinguishing characteristic of computer-assisted visual production is its ever-evolving nature. Computers and software are in constant evolution which directly reflects the activism of research in computer graphics and the voluntarism of the entertainment industry. Rarely a medium for artistic creation proved so versatile and dynamic by nature. It is easily witnessed by the drastic changes in the visual prowess executed today and seem barely related to similar creations from only one or two decades ago (*cf.* Figure 1.4). This is especially conspicuous in video games which in terms of production and direction, represent the quintessential form of computer art. Constantly bound by the limits of current techniques and hardware capacities, video games graphics and presentation are direct indicators of the artists' mastery of the digital medium (*cf.* Figure 1.5).

[3] The physical superposition of cell sheets formerly progressively prevented the good traversal of light.

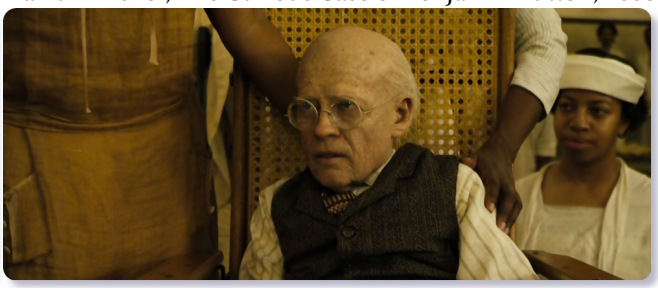
Dylan Cole, Concept art for Cameron's Avatar, 2009.

Tim Burton, Alice in Wonderland, 2010.

1.2 Computer Effects for Substituting Reality. James Cameron's Avatar is likely to make history not only for breaking most box office records, but also for the cinematographic change of direction it represents. Bringing special effects to higher levels, it is particularly successful in the believability of its computer-enhanced actors and virtual environments. At a more modest scale, Burton also intensively uses computer effects for infusing life to Carroll's universe.

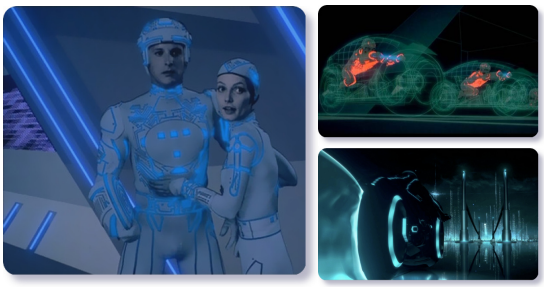
Clint Eastwood, The Changeling, 2008.

David Flincher, The Curious Case of Benjamin Button, 2008.



1.3 Computer Effects for Extending Reality. A less visible but even more impressive use of computer effects is the seamless addition of virtual settings or characters that still remain unnoticed within natural scenes. Recent examples include the virtually restored 1920s Los Angeles in Eastwood's Changeling, or Flincher's rejuvenating Benjamin Button.

Steven Lisberger,
Tron, 1982.

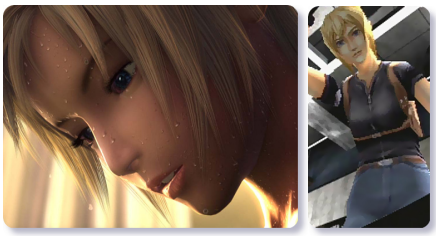


Joseph Kosinski,
Tron Legacy, 2010.

1.4 Evolution of Computer-Generated Visual Effects in Movies. Pioneering the immersion of human actors in virtual environments, Tron impressed by the strangeness of its cold albeit stylish visuals which culminated in the famous light bike race sequence. Its sequel is to lack its predecessor's originality, but will surely similarly impress.

Parasite Eve,
Squaresoft, Playstation, 1998.

Parasite Eve II,
Square PDD 5, Playstation, 1999.



Parasite Eve 3rd Birthday,
Square Enix & Hexadrive, PSP, 2010.

1.5 Evolution of Computer-Generated Visual Effects in Video Games. Finer models, more convincing facial expressions, more detailed textures, better settings and dramatic lighting, nothing seems too good for the lovely and monster stalked Aya Brea, never-aging muse of the Parasite Eve franchise.

The Reasons for the Success of Computers

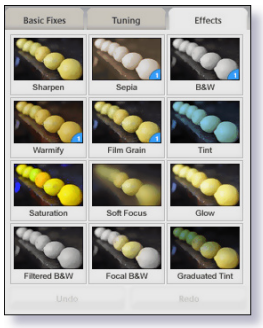
But one could wonder about the causes behind such a drastic change in art production, and the reasons that permitted such a leap of faith in favor of computers and their added value to the creation process. Machines' ever-increasing computational power left aside, a natural explanation is the maturing research in computer graphics proposing high-quality techniques for the modeling of complex 3d scenes, their fine-tuned animation, and believable rendering. We are here mostly interested in that last aspect of the visual creation pipeline.

Research dedicated to the rendition of three dimensional objects introduced in a remarkably short period of time increasingly refined algorithms and models for the faithful representation of reality and most visual phenomena, and proposed a wide spectrum of elaborated methods ranging from accurate simulation to credible imitations.

1.6 Ivan Sutherland's *Sketchpad* Console Controlled by a Light Pen and Command Button Box.

In the case of two-dimensional imagery, the advent of computers is often associated with the sudden success of *virtual canvases*. Such methods digitalizing the act of painting and removing most of the inconveniences caused by its physical counterpart, have quickly met with success since the eighties. In 1983, first Macintosh computers grew popular among artists notably thanks to their exclusive ground-breaking art-oriented software such as *Adobe Illustrator* introduced in 1986, and followed by *Adobe Photoshop* in 1990. Even today, these programs remain among the top software for digital illustration, painting and photo retouching. The apparition of graphics tablets (whose ancestor is presented in Figure 1.6) completed the digitalization of the whole painting process which turns out almost as intuitive as its real-world incarnation while extending its execution.

Virtual canvases actually profit more from their intuitiveness than from the complexity of the drawing procedures they propose. They enable artists to forget about the tools they are using while naturally helping them to give shape to their wildest imagination, for the digital medium does not suffer any physical restriction anymore. The question of a method's intuitiveness is thus important as this example would hint that controllability and apparent simplicity condition usefulness, and facilitate the adoption by artists. At the light of the sheer amount of work and computer-created content in nowadays' productions, the expedience of the involved techniques may be as important as their underlying theoretical complexity.

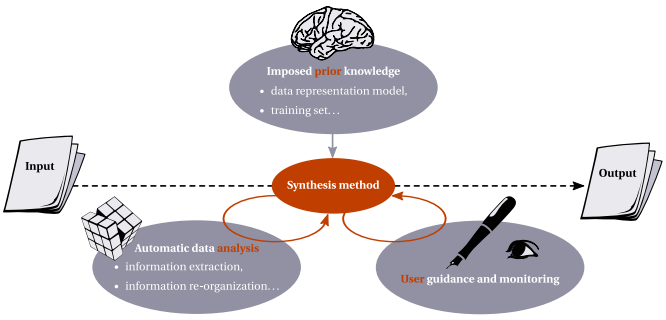


1.7 Effects Proposed by Google's *Picasa*.

At a more modest scale, with the arising of internet as a mass-media and computers and digital cameras being within the reach of a majority of people, the computational representation of images is also becoming more and more natural to everyone. As a direct consequence, the resort to computer-assisted edition of images' content by casual users (from retouching blemishes from photos via disarmingly simple tools such as Google's *Picasa* to more elaborated tweaking using softwares such as the free *GNU Image Manipulation Program*) is not the prerogative of professional artists anymore. Again, illusory simplicity and ease of use are the key features industry now counts upon in order to win over an ever-increasing audience since the democratization of the Web, and base the reputation of their services and products amidst nowadays' free supply.

Thesis

However from a researcher's perspective, one should not confuse a method's controllability with its complete surrender to user guidance. Finding a good trade-off between the two positions is at the core of the present thesis which aches for the identification of balance between automation and the machine's incapacity to guess its users' intent.



1.8 The Information Source Triangle. Mentioned throughout the thesis, this triangle presents at its extremities the three possible means of information gathering example-based approaches have at their disposal: they can either extract information directly from the inputs, understand them through the application of prior knowledge, or process users' annotations and indications.

Up until now were mostly presented methods that enable the creation of digital content from scratch. With geometric models exhibiting an ever-increasing complexity, and renderings perpetually gaining fineness and intricacy, one cannot expect artists to manually craft everything from the ground up. The present thesis tackles this concern and specifically dwells on the case of *example-driven synthesis*. Example-based techniques here designate methods being provided an input they either have to imitate or enhance. For that aim, the relevant information needs to be identified and processed, which represents the main challenge such methods must face to partially alleviate users' work. Example-based approaches are also particularly interesting and promising in the sense that they truly aim at proposing maybe the easiest and most intuitive use metaphor: only being passed an illustrative example of the desired output as argument, such methods aim at identifying its distinguishing characteristics, and accordingly replicate these as if generalizing the generative process from which the example stems. The actual challenge of such approaches lies in the deciphering of the actual information embedded in the input. From our point of view, three main sources of information can fuel their analysis stage as illustrated in Figure 1.8:

- the automatic extraction of the input's relevant information followed by its reorganization and processing allowing further synthesis;
- the resort to imposed prior knowledge and the expectation the input's visual features respect a set of known assumptions;
- and lastly, the intervention of the user along the course of the method's execution.

Finding the appropriate balance between these three aspects is no easy task, and of course heavily application dependent. But if successful in that sense, by-example methods can constitute remarkably efficient tools that can take over artists for the accomplishment of repetitive, cumbersome or tedious tasks. Our conceptualization of example-based methods' work flow begs the question of the users' role who need high-level control while not having to resort to hazardous parameter tweaking or to entirely replace the method itself.

For better considering this question, we notably took great interest in putting this issue to test in the context of expressive rendering synthesis applications. This focus indeed sheds a particular light to the problem since expressive rendering is notably characterized by the originality of the quality assessment of its results. Inattentive to the quest of photo-realism that long drove research in computer graphics, it grants more care to subjective criteria such as an output's artistic value, eye-pleasantness or informative readability. While realism can obviously be confronted against reality, algorithmic complexity analyzed, and performances recorded, expressiveness is hard to evaluate and as hard to infuse to methods claiming to preserve or enhance it. The integration of the user's feedback in their execution is therefore even more crucial than for more generally purposed synthesis methods.

The research presented herein propose different analysis schemes in various application scenarios, each of them granting a different emphasis to the three previously mentioned pillars of content creation by example. All presented approaches also aim at broadening their analysis and information management techniques, and at not staying riveted to the algorithmic tools commonly encountered in Computer Graphics. We believe its interdisciplinary nature is one of this thesis' main appeal.

Contributions and Organization

Follows the outline of the document which is divided into three main parts, each of them alluding to the question of proper information processing in distinct sub-domains of the expressive rendering field. Initially, my thesis was to dwell on the question of style capture in non photo-realistic rendering and intuitively to strive for the discovery and automation of the "hidden artistic procedure logic on human art" as Hiroshi Kawano theorized back in 1975. But during its course, we progressively nuanced our ambitions and instead consider more realistic and useful alternatives to complete automation in a wider range of applications as initially intended. I am glad it turned out that way, as close to all primary aspects of artistic graphical creation have been approached from textures and patterns, to colors and contrasts, to lines and shape, always at the light of our knowledge of human perception and under multi-disciplinary vantage points.

Part I transposes our interrogations to the case of texture analysis and synthesis. This work is characterized by a strong will of automation, restricting the user's intervention to the maximum, and assumes a form closer to object detection as one usually finds in computer vision than in conventional texture synthesis methods as detailed in Chapter 3. Directly following work presented in Chapter 4 explore vector-based arrangement synthesis by example. Once again, pronounced emphasis is given to automation but the main difference with respect to the previous technique comes from the improved importance attached to prior knowledge that appears via the fitting of a specifically chosen statistical model.

Part II tackles the daily encountered albeit overlooked issue of color to grey-scale conversion of images and videos. This time automation voluntary takes a back seat as our objectives for perceptual accuracy can only be handled by resorting to prior knowledge by way of complex color specifications, brightness predictors and entoptic illusions. Users are also given a fairly limited role, and are mostly required to control the strength of the contrast enhancement our method brings so that it best fits the intended displaying conditions. Simple by essence, albeit successful in its results, our method is among the first to naturally handle temporal coherence and attach such importance to the perceptual relevance of its results.

Part III finally presents ongoing work that strongly contrasts with the aforementioned research in the sense that it heavily relies on the user's input. It resolves around the perennial issue in expressive rendering of the automatic depiction of 3d geometry as line drawings. Our technique assumes the form of a drawing assistant that considers lines drawn by users and attempts to anticipate their artworks via machine learning. For that aim, we cast our problem of line extraction from surfaces as a binary classification problem whose training data set incorporates users' sketches and corrections.

Naturally, these different work involve many persons beside myself, and beside motivation and inspiration, these different collaborations made possible their publication:

- *Analyse et synthse de textures composes de motifs rptitifs*,
  Pierre-Edouard Landes & Cyril Soler,
  REFIG'09 (*Revue Electronique Francophone d'Informatique Graphique*), later published as a more detailed technical report in English;
- Appearance-guided Synthesis of Element Arrangements by Example,
  Thomas Hurtut, Pierre-Edouard Landes, Jolle Thollot, Yann Gousseau, Rmy Drouilhet, Jean-Franois Coeurjolly,
  NPAR'09;
- Apparent Greyscale: A Simple and Fast Conversion to Perceptually Accurate Images and Video,
  Kaleigh Smith, Pierre-Edouard Landes, Jolle Thollot, Karol Myszkowski,
  EUROGRAPHICS'08;
- finally, the last ongoing and very recently submitted project has been conducted in pair with Amit Shesh.

Part I

Texture Synthesis by Example

This first part brings our reflection on the sources of information at the disposal of any data-driven method to the realm of texture synthesis, and exposes two novel techniques for the handling of a very specific category of textures for which most of earlier techniques usually break. This "specific category of textures" encompasses textures consisting of the spatial arrangement of individually discernible shapes. Both our proposed methods are geared to best handle such inputs either in raster, or vector form. From now on, we refer to such shapes as *patterns* if of raster nature, or as *elements* if vector. Depending on the computational representation of the user-provided examples, the hurdles the analysis stage of our algorithm must face greatly differ. While the main challenges of the processing of a raster input arise from the localization of its patterns and its robustness to their potential mutual overlap, the difficulty of the analysis of a vector arrangement arises from the capture of the spatial layout of its elements in its entirety.

Since the task of manually extending such textures represent tedious and cumbersome work, we believe such methods should be as automatic as possible. Our raster method therefore aches for complete automation and strives to do so without the resort to any prior knowledge about the appearance or disposition of the sought patterns. This objective comes still at the price of assuming that these patterns are seen front a frontal view devoid of perspective skew. The ultimate goal of our method is to detect these patterns from the soup of the input's pixels. We claim that it is only by directly manipulating its relevant constitutive parts instead of its individual pixels that the re-synthesis of such textures can be successful. Our technique for the extension of vector arrangements shares this will of automation, but resorts to statistical models in order to globally study the disposition of the already-identifiable elements.

Context and Motivations

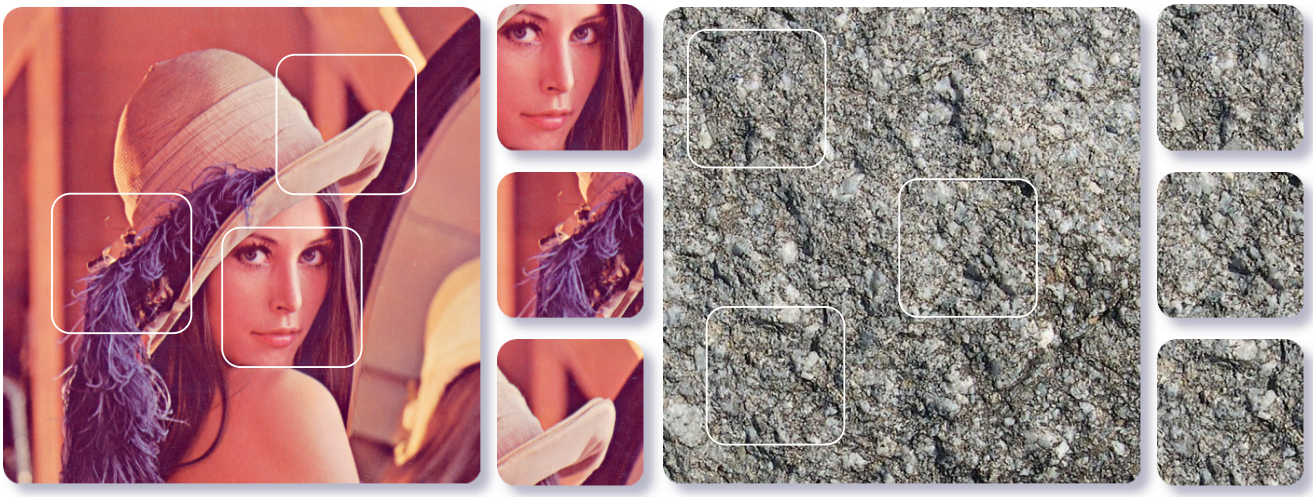
Nowadays, computer-generated art forms propose a stunning visual quality. More and more often to the point that it seems that a ultimate peak had been reached and that no further improvement is possible. Naturally, the amazement caused by subsequent artwork, movies typically, clearly contradict this impression. Along with impressive advances in the modeling and animation departments, the emergence of more and more intricate rendering techniques greatly contributed to the believability of such productions, for the same extend as finely-modeled characters and scenes or convincing body language. The research presented therein is fueled by this observation as well as the prominent role textures have gained for the creation of this compelling virtual reality.

Defining Textures

Giving a precise and final definition to the umbrella term of *texture* is actually no easy task. Indeed, depending on its context, a texture can refer to strikingly different concepts, but all its different meanings still share the reference to some vague notion of our sensory impression of roughness or smoothness. From a loosely mathematical standpoint, *textures* correspond to an infinite pattern arising as the realization of a stationary, local stochastic process (*cf.* Figure 1.9).

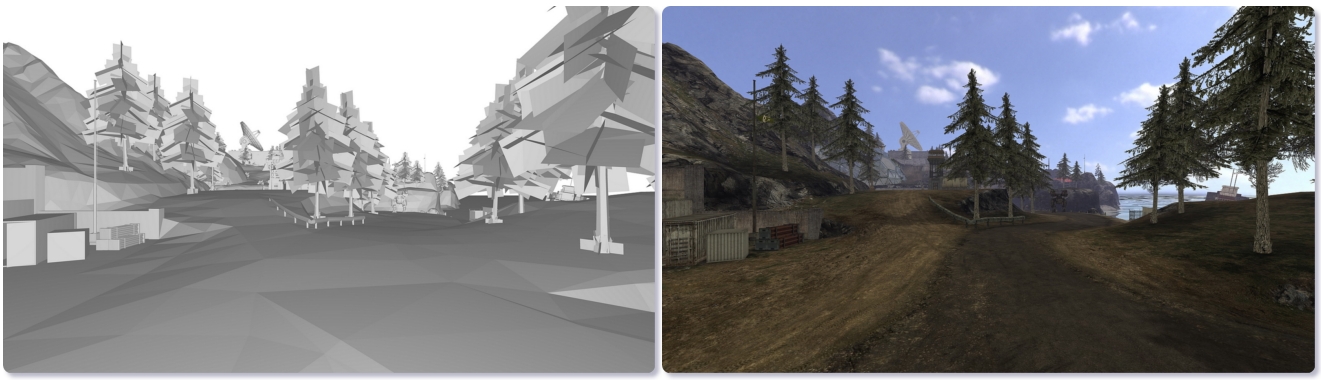
But by a persisting misuse of language, a texture in Computer Graphics mostly evoke their computational representation, namely an ordered one-, two- or three-dimensional set of values used for encoding the spatial variation of the visual characteristics of an object surface. First introduced as raster images – therefore as 2d arrays of pixel colors – by Edwin Catmull in 1974 for decorating the surfaces of displayed 3d objects (*cf.* Figure 1.10), the resort to textures for representing the spatially-varying albedo of surfaces indicated a drastic change of direction in terms of visual quality attainable in Computer Graphics. Indeed, prior to their apparition, only metallic objects could be convincingly portrayed by techniques such as ray casting which thanks their precise handling of the light behavior inspired by geometrical optics could handle the addition of glossiness, reflection and transparency effects. Still, the appearance of these virtually-created objects lacked the impression of "imperfection" real-world objects exhibits because of wear and weathering.

Catmull's contribution has been groundbreaking to the point that no other way better allying efficiency and ease-of-use has been found and texture became an irreversibly indispensable tool in the creation process.



1.9 Difference Between a Photograph and a Texture. Heavily inspired from the visual comparison found in a recent survey [WLKT09], this couple of images empirically shows the differences between general images and the specific sub-case of textures. As previously stated, textures are supposed to result from a stationary, local underlying stochastic process. Here, *stationarity* appears as we move the a sliding window across the rock texture: no matter its actual position, the details it reveals seem roughly similar. Alternatively said, the pixel color statistics remain approximately unchanged. This is definitely not the case with the Lena image whose cut-ins are clearly distinguishable from one to another. *Locality* refers to the fact the colors of individual pixels only seem to related to their immediate neighbors.

1.10 Catmull's Texture Mapping [Cat75]. Catmull proposed back in 1974 the first texture mapped objects even seen on a screen, as the outputs of his patch-based subdivision algorithm for curved surface display.



1.11 Use of Textures in Video Games. The sole addition of texture confers an incredibly amount of details to the even simplest geometry. Note that the bottom image is not even lit. Source: Enemy Territory: Quake Wars, Ray-traced project © *Splash Damage Ltd, Nerve Software*

Simply put, textures now pervade the entirety and all types of computer-generated media art from movies of course, to video games where their efficient display still constitutes a formidable challenge for most gaming systems (*cf.* Figure 1.11), to the more unconventional uses such as for decorating 2d hand-drawn animations (*cf.* Figure 1.12).

Mahiro Maeda,
Gankutsuou: The Count of Monte Cristo, 2004.

Kazuto Nakazawa,
Advertisement for *Asience*, 2007.

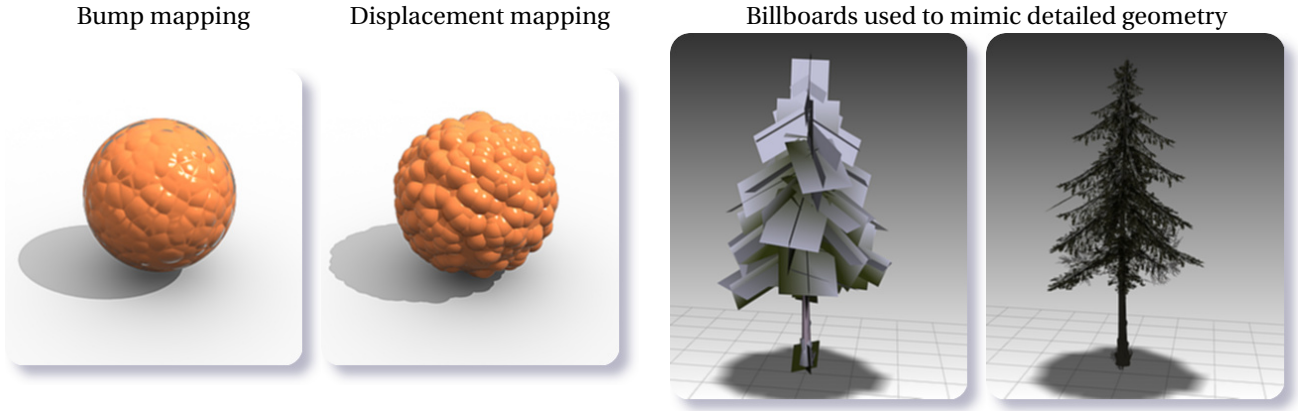
1.12 Recent Unconventional Use of Textures. Recent Japanese animations boast textures for decorating their drawings and conferring a unique look to their creations. In his adaptation of the famous Dumas' novel, Maeha achieves stunning visual by having textures covering the whole screen from the protagonists' clothing to their hairstyle. This abundance of visual details that gave the visual impression of watching an animated marquetry became the trademark of the series, both to its benefit and detriment as such lavishness comes at the price of an hindered animation.

Using Textures

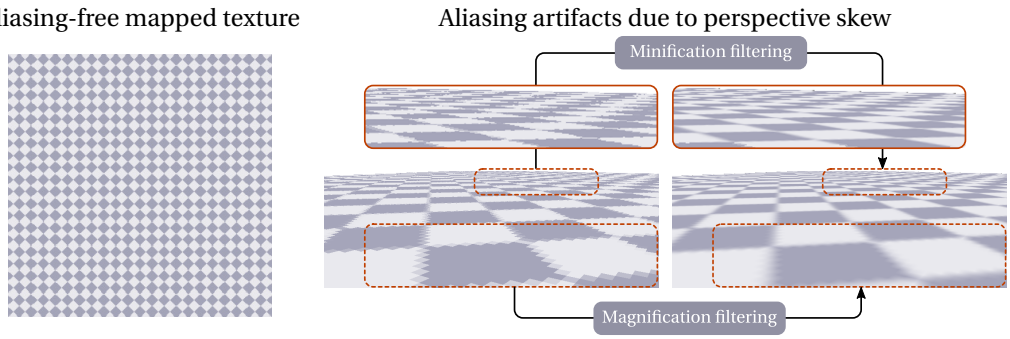
Textures as Appearance Descriptors While it may appear as geared towards the easy painting of a geometrical surface, Catmull's texture mapping technique revealed itself general and efficient enough to allow its use in a much broader sense. Indeed, Catmull's approach lends itself quite easily to the application of any sampled appearance measurements over objects' surfaces. First employed for adapting color depending on the 3d position over the surface [Cat74], it soon enabled the precise spatial control of any other factors used for the computational representation of an object's reflectance property. Examples of data processed as textures include specular reflection [Bli78a] and glossiness coefficients [Bli78b], or even local transparency [Gar85]. Complex appearance models exhibiting a position-dependent behavior such as the Spatially-Varying Bidirectional Reflectance Distribution Functions [NRH+92] and Bidirectional Texture Functions [DNGK97] can also be considered as textures to some extent.

Textures as Geometry Containers Textures can also go beyond their initial task of detailing a material's appearance, and partly encode more or less directly its very geometry. Indeed, from an artist's point of view, drawing surface details over a 2d planar texture plane is incredibly more practical than explicitly incorporate these same details over a 3d model. Consequently many successful attempts strove at transferring this cumbersome 3d modeling task into an easier and eventually re-adaptable 2d art work. Among these, *bump mapping* [Bli78a] and subsequent *displacement mapping* [Coo84] use textures mapped over a 3d mesh in order to modulate its normals' perturbations, or the strength of the displacement of the underlying geometry either by directly moving its vertices or adaptively further tessellating it. But one of the most complex use of textures as surrogate geometry is indisputably Oliviera's *relief texturing* which embeds the surface's finer geometry into textures storing the orthogonal displacement of their underlying surface [OBM00]. These techniques, long confined to high-quality renderers, are commonly encountered in nowadays' video games as attested by the inclusion of *parallax mapping* in *Crytek*'s rendering engine.

Surpassing Geometry By trading its initial local and stationary texture for a general image supporting transparency and mapping it to a simple 3d plane in any scene, Catmull's texturing technique can mimic the finest of geometry at an unbeatably low cost in terms of creation time and rendering performances. This technique referred to as *billboarding* is intensively used, notably by video games, and more than a trick, represents now an unavoidable method for ensuring good visual quality at acceptable performances.



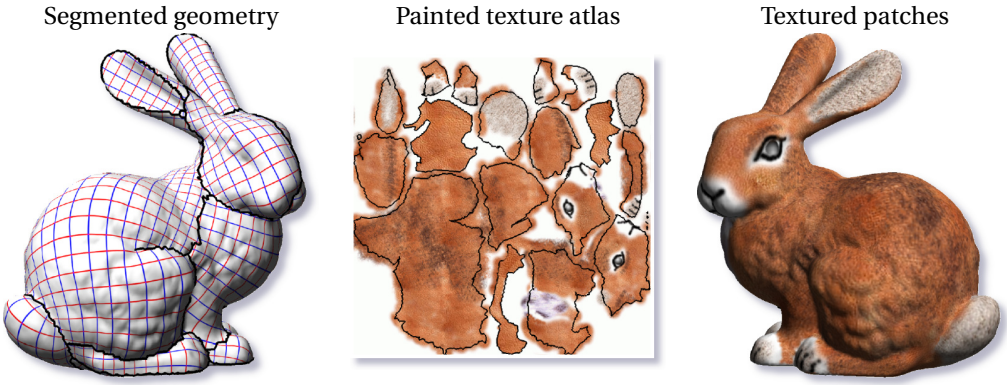
1.13 About the Various Uses of Textures. Even those simple examples clearly demonstrate how ubiquitous and indispensable textures have become for high-quality renderings, especially for the real-time constraints video games face.



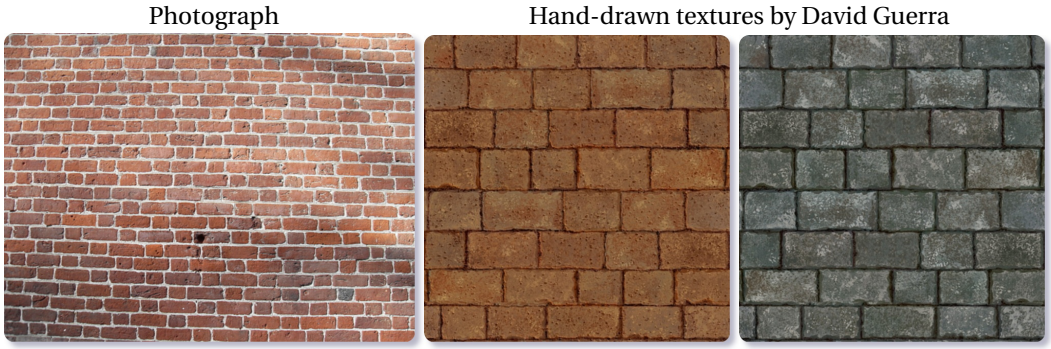
1.14 Texture Rendering Artifacts and Their Correction. Aliasing issues are possibly one of the most (in)famous rendering artifacts and stem from the local maladjustment between the sampling performed by the screen pixels and the resolution of the texture once mapped onto the projected geometry. For limiting the blocky appearance of an oversampled texture or the flickering of a undersampled texture, interpolation or averaging can be performed.

Rendering Issues For rendered textures to look good, both their resolution and the sampling scheme imposed by the projection onto the surface of the pixel lattice must work in concordance. Texture rendering artifacts come with two flavors: If too many adjacent pixels sample a same texel, the texture appears with a blocky appearance, a phenomenon called a *magnification artifact*. Conversely, if the sampling of the pixels is too loose with respect to the texture resolution, some of its features may not appear and over the course of an animation pop in and out constantly. This is called a *minification artifact*. Solutions to these problems are texel interpolation and filtering for handling magnification and minification issues respectively (*cf.* Figure 1.14). These additional treatments are more and more well-handled by rendering architectures from their pre-computed approximations by MIP mapping to the XBOX360's Multiple Sample Anti Aliasing.

Parametrization Issues Another difficulty of employing textures comes from how to map them to an arbitrary geometry. Theoretically speaking, texture mapping boils down to find a functional called a *parametrization* that unambiguously assigns positions from the 2d texture space to the 3d object space of the scene. While closed-form solutions exist for simple geometrical shapes, more commonly used meshes' topologies are not homeomorphic to a disc, which means that they just cannot be "unfolded" over the 2d plane without cuts. Solutions exist for overcoming this hurdle, most of them relying on *texture atlases* which divide the texture into parts to circumvent this lack of topological equivalence between the two spaces (*cf.* Figure 1.15). But special care must therefore be given for hiding possible seams between adjacently mapped patches.



1.15 Atlases for Texturing Arbitrary Geometries. Finding a consistent and visually pleasant mapping from all 3d surface positions to each texel is no easy task for most geometries. Textures atlases are a convenient way to circumvent the lack of global parametrization of such surfaces by segmenting them into patches topologically equivalent to a disc. Automatic methods such as Lévy's exist for finding appropriate cuts that also minimize distortions [LPRM02].

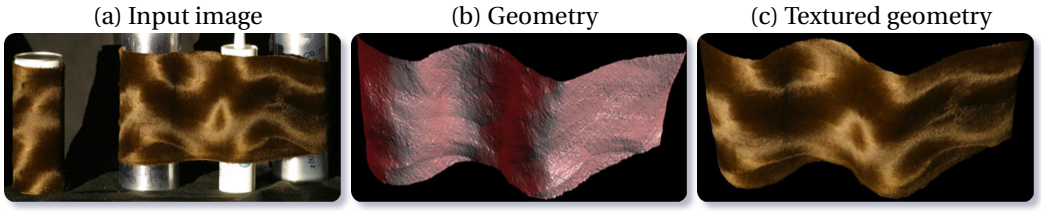


1.16 Scanned Imagery vs. Hand-Drawn Textures. While straightforward to obtain, scanned textures (*left*) often require additional post-processing to eliminate visual artifacts due to non-uniform illumination conditions (cast shadows onto the wall surface) or unwanted perspective distortions. Moreover, such images naturally lack tileability and directly mapping them onto some surface results in visible seams. On the other hand, textures created by an artist (*right*) are free from all those concerns but come at the price of a more cumbersome acquisition process.

Acquiring Textures

But more than the inconveniences a too hasty rendering may cause, the foremost difficulty lies in the acquisition of textures. Several alternatives for obtaining believable textures exist: acquiring them from real-world data, having them manually drawn by a texture artist, or finally synthesizing them.

- Attaining a exploitable level of quality by capturing the visual appearance from real-world objects is trickier than it seems. Usually, directly taking photographs of the texture one wants to exploit requires a lot of care during the shooting and often demands digital post-processing (*cf.* Figure 1.16-*left*). Granted the texture can be found on a planar surface that is possible to take a photograph of, an approximative photography set-up may cause small artifacts jeopardizing the direct use of the photograph as a texture (perspective skew, cast shadows or uneven illumination conditions).
- Hand-drawn textures by artists are of course devoid of such artifacts and certainly represent the most eye-pleasing results one could possibly expect (*cf.* Figure 1.16-*right*). But such levels of quality necessitate a laborious manual work.
- Recent scanning techniques make possible the automatic acquisition of both geometry and material properties of real-life objects and hint the possibility of acquiring not only visually pleasing but also physically accurate textures automatically [HS03] (*cf.* Figure 1.17). Their appearance model allows the capture of anisotropic bidirectional reflectance distribution functions and hence can be used to represent a wide range of potentially visually intricate materials. However, such techniques are not only expensive but also



1.17 Shape and Material by Example. The approach by [HS03] proposes to recover, for a given viewpoint, objects' geometry *(b)* along with their reflectance properties *(c)* from a set of images *(a)*. This method achieves impressive results, but restricting assumptions hinder it from replacing classical texture synthesis.

not that convenient to use in practice.

- A last alternative is *texture synthesis* which arises from the need of always bigger textures and the creation of arbitrarily large textures to ensure the full coverage of any geometry. A first category of texture creation is called *procedural texturing* which relies on the seemingly random aspect of the textures they aim at reproducing. By way of modulated noise functions, these approaches distinguishing themselves by their exclusively mathematical or programmatic nature. Therefore, such textures are *evaluated* by a series of operations taking as argument the 3d position to color, and do not arise from the mapping of some external *stored* image. Along their fast evaluation auspicious for real-time rendering, they can also be computed at any spatial position and resolution. Hence, they do not suffer from the previously mentioned magnification and minification aliasing artifacts and do not require the specification of any mapping for decorating the surface, objects appearing like being carved out of them. However, they suffer from a severe lack of expressive power which makes them all look like natural materials or distributions of objects. They moreover require solid programmatic skills if one wants to take full advantage of them.

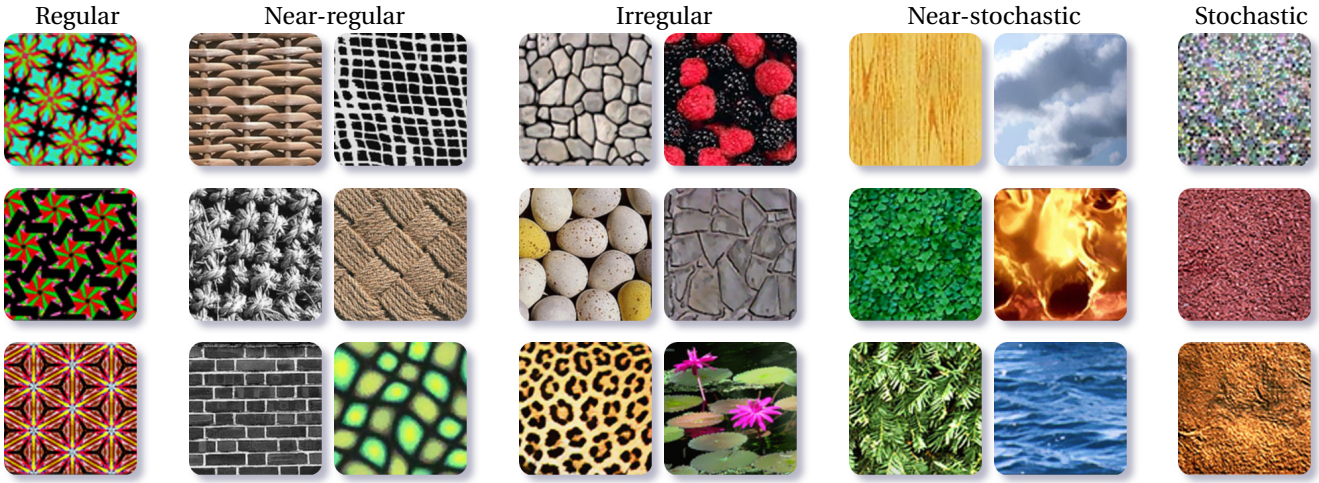
Synthesizing Textures By Example

The second category of texture generation is the synthesis of textures *by example* which constitutes a much more general tool for artistic creation. The goal of example-based synthesis is as follows: given an input, find a way to extend it so that it appears as if arising from the same process as the example. As easy as it may sound, capturing the visual characteristics of this unknown generative process from an unstructured pool of pixels is a thorny issue. Moreover, the extreme sensitivity of our visual system to repetitions forbids the careless tiling of the plane using verbatim copies of the input. Hiding seams between set of grouped pixels, limiting distortions, infusing visual randomness while not sacrificing controllability, all of these concerns represent great challenges for such methods.

Another hurdle comes from the variety of inputs such methods can be confronted to. This fact is the natural consequence of the vague definition of a texture and the subjectivity of our judgments for differentiating a texture from a general image. But a common denominator exists between all textures: the fact that they seem to arise from the repetition of some of their parts. On the other hand, this repetitiveness can exhibit a varying level of predictability, from fixed to purely stochastic. Back in 1986, Gagalowicz wrote that "a texture is generally considered as two levels of hierarchical information. It is a spatial organization (highest level) of primitives (or basic patterns) where each primitive (lowest level) has a random aspect" [GSDM86]. His formalization of texture is in accordance with Hays *et al.*'s subsequent classification of textures [HLEL06] which split up textures into various categories depending on the regularity of their constitutive primitives (*cf.* Figure 1.18).

Our Texture Synthesis By Shape Recovery

By simply looking at Hays' texture spectrum, we can realize that no universal example-based method can possibly capture all the samples it displays; the entirety of the visual cues that condition the believability of the results; and thus ensure the success of the technique. Indeed, a texture's distinguishing features are entangled with our *understanding* of its constitutive components. While the most irregular textures exhibit pixel



1.18 A Glimpse at the Remarkable Variety of Textures. Shown above is the 1d classification of textures proposed by Hays *et al.* [HLEL06].

dependencies contained in reasonably small neighborhoods, highly structured ones sport long-range pixel relationships which are more difficult for the method to identify and track.

From this variety of possible inputs, a myriad of synthesis methods have been proposed. All share the common trait that they consider the observed texture as a realization of a Markov Random Field. This assumption indeed provides the most elegant and tractable framework for studying the texture's global appearance by way of the consideration of the local interaction between its pixel colors.

Yet, despite the impressive advancements in terms of computational elegance and the visual quality of its outputs, example-based texture synthesis still has room for improvement as some textures are still not perfectly handled by any existing method. These textures lie at the center of Hays' spectrum and while they do not exhibit the perfect regularity of the wallpaper patterns found on its far left, they involve larger dependencies than the ones of the stochastic samples found on its right. We claim that previous methods fail to capture their salient features since as they manipulate independent pixels or unstructured pixel patches, they do not evolve at a suitable scale. Conversely, approaches solely driven by the spatial layout of these features enforce highly restrictive priors over their inputs' content and easily break when the placement rules they rely on are not strictly observed. In our work, we propose an attempt at synthesizing such textures made of distinctively discernable shapes which repeat themselves throughout the observed sample. By analyzing the consistent reoccurrences of visually similar local neighborhoods, we can localize and identify these constitutive shapes. Once recovered, they constitute better building blocks for re-synthesis than plain pixels.

Contributions and Outline

The present part of my thesis comprises two of its contributions in the field of example-driven texture synthesis. The first one dwells on the generation of textures represented by *raster* images, whereas the second one focuses on the re-synthesis of arrangements made up of *vector* elements. The former research, detailed in Chapter 3, has been conducted under the supervision of Cyril Soler. The later stems from the direct collaboration with Thomas Hurtut and Joëlle Thollot and is presented in Chapter 4. Since I participated to the success of this second project at a later stage, notably for its publication and presentation, it is simply presented in its published form as I cannot claim ownership over most of its theoretical content. Chapter 2 is dedicated to earlier methods for the synthesis of raster textures by example. As for the existing techniques for the creation of vector textures, they are mentioned directly in Chapter 4 since their number is greatly inferior to the amount of approaches for raster inputs.

CHAPTER 2

State of the Art in Raster Texture Analysis and Synthesis

Textures almost effortlessly enable computer artists to visually enrich their creations with subtle details. Even the coarsest geometry can be transformed by the addition of textures mimicking complex materials or additional geometry. However, this expressive power comes at the price of devising efficient and intuitive texture creation routines.

Two main and complementary trends exist to handle the question of automatic texture generation. The former families of techniques create new textures using explicit mathematical formulations. Their results are referred to as *procedural* textures. The latter category of approaches, called *example-based* or *texture-from-sample* approaches, strive to produce new textures resembling user-provided inputs. Their goal can be formulated as such: assuming the observed input is the realization of some unknown texture generative process, let us design new textures that "appear" to arise from the very same process. Aside from the fuzzy definition of a "similar appearance", the main hurdle is to find a way to model this texture generative process, either explicitly or not, and sample it somehow in order to obtain new realizations.

The present thesis' contributions fall into the second category. However, procedural techniques do propose many advantages and are worth mentioning *per se*. The results they provide are also widely used in practice, experienced users being notably able to handcraft impressive results out of them.

2.1 A Glimpse at Procedural Texturing

Procedural textures refer to textures described by an analytic mathematical formulation which determines its visual appearance through a set of algorithmic routines. Such textures present several advantages: only a few pre-computations is usually required to enable their real-time evaluation; they can be effortlessly evaluated and sampled at any scale; and since their specification relies on a small set of parameters, their representation is especially compact. Procedural textures are typically used for representing natural materials such as wood, rocks or organic patterns as well as other phenomena like smoke, fire or clouds. Their ability to efficiently evoke the randomness commonly observable in nature is essentially due to their reliance on fractals or turbulence functions.

Earliest examples are Gardner's textures obtained by summing sine functions of various periods, phases and amplitudes [Gar85]. But the most famous and widely used procedural texture is the much-acclaimed Perlin Noise [Per85]. Noise designates a pseudo-random function whose most interesting property is to return the same value for different calls given the same parameter. Perlin interpolates between a set of pseudo-random gradient vectors to obtain a texture that continuously spans the 2d or 3d space, and linearly combines several noise functions whose frequencies are carefully chosen. These functions are called octaves whose contribution is made inversely proportional to their frequency. Modulating object colors in accordance with the obtained values enables the easy creation of many effects as shown in Figure 2.2*(a)*.

Another well-known procedural noise is Worley noise, computed as the distance between the position where the noise is to be evaluated and a set of fixed points distributed in space. Again, various effects are achievable by using different distance metrics and applying non-linear operations on the resulting values (*cf.* Figure 2.2*(b)*).

Textures were initially used to effortlessly mimic materials of surfaces commonly observed in nature. As such, it comes rather naturally to attempt to create these by simulating the chemical processes behind the phe-

2.1 Sparse Gabor Convolution Noise [LLDD09]. Here are four examples of texture noises obtained using Lagae *et al.*'s procedural method. The user directly tunes their power spectrum's shape to control the appearance of the final output.

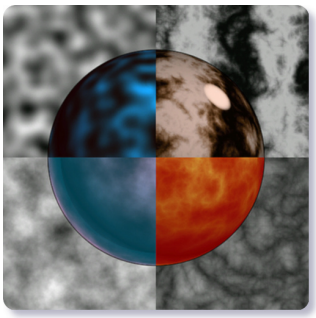
nomena to be imitated. Reaction-diffusion is an illustration of such Biology-inspired imagery. It models the creation of patterns emanating from the interactions of chemical substances whose local concentration evolution evolves according to two processes: reaction – how one substance gets locally transformed into another – and diffusion – how it expands through its neighboring space. From Turing's research on pattern formation in biological morphogenesis, the resulting textures are defined as the equilibrium states of non-linear quadratic differential systems [Tur52]. In order to solve this system at the surface of a mesh, Turk discretizes these differential equations and constructs a cellular automaton over the geometry to be textured [Tur91]. Witkin and Kass extend such approaches by investigating the effects of anisotropic diffusion following competing directions. Patterns achievable by Reaction-Diffusion range from spot-like mammalian patterns, to intricate and curvy stripes as illustrated in Figure 2.2*(c)*.

The usability of procedural textures lies more in their control than in their appearance versatility and it can be made even more efficient for spectrally-defined textures. Good examples are fractals whose self-similarity makes them well-suited to the representation of textures and stochastic surface properties [FFC82]. But a particularly enlightening approach is Lagae *et al.*'s sparse Gabor convolution noise [LLDD09]. Spectral analysis is often used as a *a posteriori* noise quality assessor, but their insight is to use it during the noise creation process. Their procedural noise is defined as the convolution of a Gabor filter with a Poisson-distributed set of impulses. They hence have the closed-form expression of its spectral energy at their disposal and thus can use the Gabor filter's parameters as editing handles that enable users to create new noise functions by intuitively modifying their spectral energy distribution through a graphical interface (*cf.* Figure 2.1). Alongside the creation of a wide range of textures, they also elegantly handle anisotropy and high-quality anti-aliased surface noise.

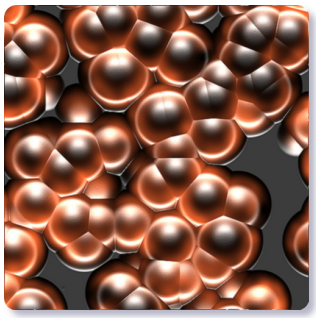
Until now, all aforementioned procedural methods generate results whose overall appearance is more or less random and it would be unfair to imply that procedural texturing is unfit to the creation of more structured textures where shapes can be perceived. Handling such cases boils down to conceive efficient object distributions with controllable and satisfactory spatial properties. They further widen the scope of the set of procedurally obtainable textures, and our gait can actually be seen as the inverse problem: from the final texture, identify and localize the distributed shapes and reverse engineer their distribution rules. An example of such distribution functions is LAGAE's texture basis function based on the Poisson-disk distribution [LD05] whose construction and query can be carried out at real-time performance while guaranteeing the respect of its spectral characteristics. Example are provided in Figure 2.2*(d)*.

This brief overview is obviously not exhaustive and for more details or techniques, our reader is invited to refer to readings exclusively centered on that field such as [EMP+02]. All those aforementioned techniques exhibit many advantages, notably the control they provide and their compact representation which make them especially interesting for memory or bandwidth limited applications. However, their main drawback is the lack of intuitiveness of this very control they grant. We acknowledge that experienced artists can mold impressive scenes with such textures, but the learning curve towards their mastery is steep (*cf.* Figure 2.3). Moreover, in the case of simulation-based textures, computation times prevent interactive feed-back, which makes all creation process by trial-and-error even more difficult. These limitations in terms of ease-of-use fueled most of the research in example-driven synthesis.

(a) Four Perlin noises [Per85]

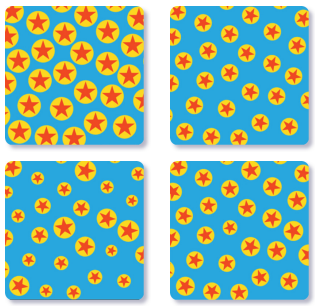


(b) Sinus-modulated Worley noise [Wor96]

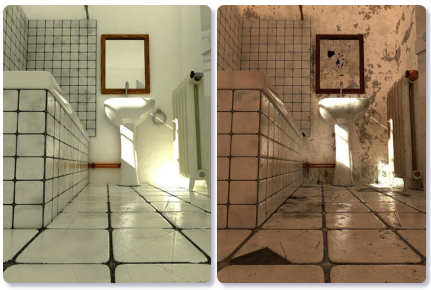


(c) Three reaction-diffusion textures [WK91]

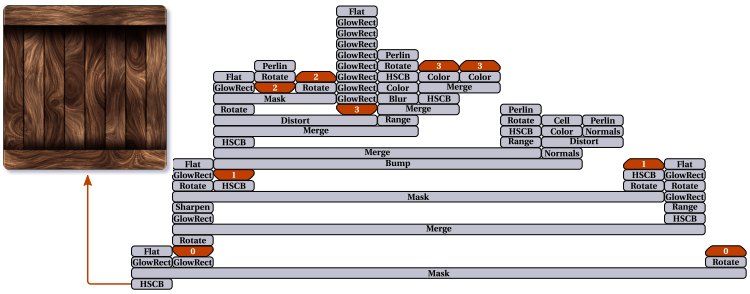
(d) Lagae *et al.*'s procedural object distributions [LD05]



2.2 Famous Illustrations of Procedural Textures.



© *Allegorithmic*



© *.theprodukkt GmbH*

2.3 Promises and Perils of Procedural Texturing. The bathroom scenes on the left attest of the quality attainable with procedural textures. Alas, producing such textures from mathematical functions often turns out as fairly complicated as it requires the clever filtering and blending of many noise functions. This is illustrated by the sequence of compositing operations required to obtain the wood crate texture on the right.

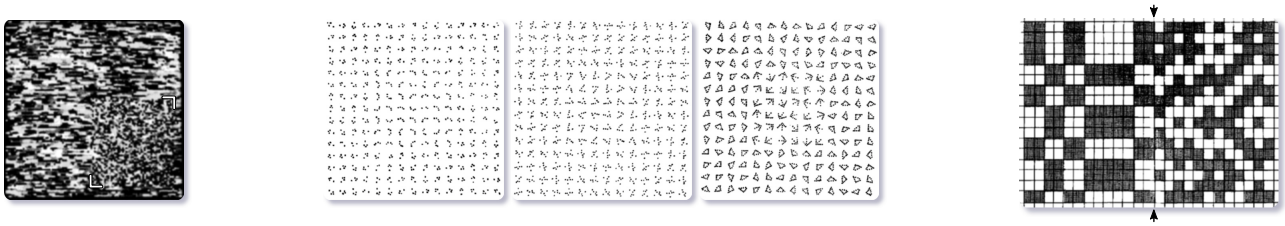
2.2 Example-Based Texture Generation

The principle behind example-based texture synthesis can straightforwardly be summed up as follows: given an input sample of the desired texture, create new textures that elicit the visual impression to emanate from the same generative process. This simplicity of concept and use are obviously the strongest point of this category of approaches when compared to procedural techniques. But this seemingly easiness is quite deceiving as it begs several challenging questions.

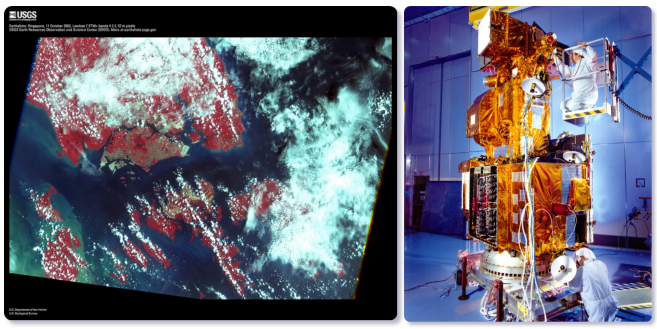
The first issue is to give a proper meaning to having outputs *"appear similar"* to the provided inputs. This trivial-sounding question brings forth concerns from Psycho-cognitive Sciences and requires a broad understanding of the human visual system's image formation processes. Not only useful as a posterior quality measurement, finding ways to encode visual similarity can also be used as a constraint during texture creation. The second question is how to model this texture generative process, train it and yield new simulations from it once fitted to the training example.

2.2.1 Texture Characterization Using Tonal Statistics

The following chapter investigates methods using texture statistics as a means to describe and produce textures. Close to all statistics-driven approach for texture description and modeling take their fundamentals from recently departed Belà Julesz's research in visual discrimination. Discrimination is the spontaneous visual process giving the immediate impression that two visual stimuli differ. Julesz conducted many visual experiments and pioneered the use of computer-generated stimuli in 1962 [Jul62] when research in visual perception relied on impoverished real-world stimuli or visually deficient subjects for controlling the perceived visual information. Computer-generated stimuli provide an increased control over their statistics, topological and heuristic properties. Julesz studied how easily textures of predetermined N^{th}-order joint probability distributions can be discriminated at a pre-attentive visual stage.



2.4 The Rise and Fall of Julesz's conjecture. Early experiments involving textures obtained from 1d Markov chains implied that 2nd-order statistics conditioned discriminability *(left)*. Corners highlight the actual boundaries between the two texture patches [Jul62]. But counter-examples showing that additional cues (quasi-collinearity, corners, closure) must also be accounted for, have been proposed since then *(center)* [CB78a]. Easily discriminable, hand-made textures sharing the same 3rd-order statistics have also been designed by Julesz himself [JGV78]. Arrows mark the limits between the two textures.



2.5 Textures in Satellite Imagery. Shown here is an example of the kind of images requiring automatic terrain classification and which fueled research in the automatic extraction of textural features *(left)*. It was captured by the Landsat 7 satellite launched in April 199 by the NASA *(right)*.

Julesz's first conjecture asserts that pixel values' joint distributions correspond to cues our visual system relies on for effortless discrimination. His early research hence focuses on finding the value of N for which two textures differing in their $(N+1)$-order statistics discriminable. Stimuli obtained through the realizations of 1-dimensional Markov chains tend to show that only the tonal qualities (1st order), and local granularity (2nd order) are sufficient to differentiate stimuli [Jul62]. Julesz also investigates the influence of our natural tendency of bringing pixels of a similar brightness together over early discrimination.

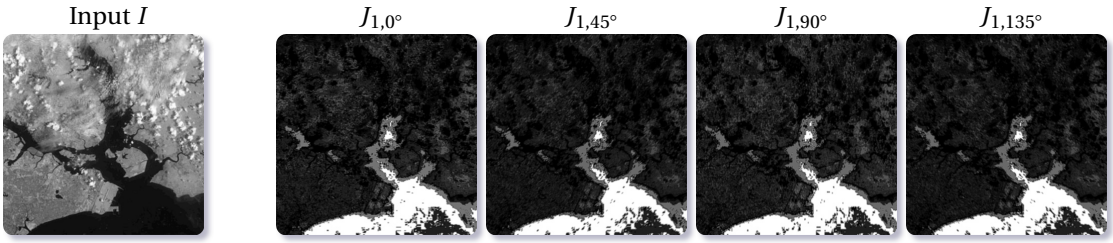
However, as appealing it may be, advances in statistical modeling enabled the creation of stimuli invalidating Julesz's conjecture. He himself exhibited undistinguishable two-dimensional textures differing only in their 3rd-, and higher order statistics [JGSF73, Jul75]. Research conducted with Terry Caelli also stressed the need to incorporate Gestalt theory-related information, such as quasi-collinearity, closure and corners [CB78a, CB78b]. He also contradicted his own intuition that texture granularity is only dictated by 2nd-order joint statistics [JGV78]. Julesz's initial conjecture stating that textural elements are represented at a cellular level through their statistics is thus now considered obsolete. Nevertheless, his idea of constraining textures' statistics for preserving their visual appearance had a tremendous influence on subsequent research in texture analysis and classification.

Early texture analysis research has also been driven by the need of processing and automatically segmenting satellite imagery into regions corresponding to specific types of terrains, e.g. woodlands, fields, urban areas or water (*cf.* Figure 2.5). An efficient way of performing such a task is to extract a descriptive representation of the textures associated with each kind of the landscapes as observed by satellites. For instance, Darling and Joseph identify the type of clouds from the earth's atmosphere by characterizing their texture thanks to the statistics of the tone transition matrices computed between adjacent pixels [DJ68]. Haralick also records co-occurrences of grey levels coupled with the notion of spatial dependence, and extract discriminative features from the input image's Grey Level Co-occurrence Matrices (GLCM) [HSD73]. Given an input texture image I whose pixels are assigned tonal values from the discrete set $\mathscr{G}$, the coefficients of the GLCM associated with the pixel layout defined by the pixel distance d and angle α are computed as follows:

$$\mathrm{GLCM}_{d,\alpha}(g_1,g_2) = \left|\left\{(\mathbf{p},\mathbf{q}) \text{ such that } I(\mathbf{p}) = g_1 \bigcap I(\mathbf{q}) = g_2 \bigcap ||\mathbf{q}-\mathbf{p}|| = d \bigcap \tan^{-1}\left(\frac{\mathbf{q}_y-\mathbf{p}_y}{\mathbf{q}_x-\mathbf{p}_x}\right) = \alpha\right\},\right|$$

$$\begin{array}{ccccc} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 2 & 2 & 1 \\ 0 & 2 & 2 & 3 & 3 \\ 2 & 2 & 3 & 3 & 3 \end{array} \quad \mathrm{GLCM}_{d,\alpha} = \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \end{array} \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \quad \mathrm{GLCM}_{1,0^\circ} = \begin{pmatrix} 6 & 3 & 1 & 0 \\ 3 & 6 & 2 & 0 \\ 1 & 2 & 6 & 2 \\ 0 & 0 & 2 & 6 \end{pmatrix} \quad \mathrm{GLCM}_{1,45^\circ} = \begin{pmatrix} 6 & 1 & 0 & 0 \\ 1 & 6 & 2 & 1 \\ 0 & 2 & 8 & 0 \\ 0 & 1 & 0 & 4 \end{pmatrix} \quad \mathrm{GLCM}_{1,90^\circ} = \begin{pmatrix} 8 & 2 & 1 & 0 \\ 2 & 6 & 3 & 1 \\ 1 & 3 & 4 & 2 \\ 0 & 1 & 2 & 4 \end{pmatrix} \quad \mathrm{GLCM}_{1,135^\circ} = \begin{pmatrix} 2 & 3 & 3 & 0 \\ 3 & 4 & 2 & 0 \\ 3 & 2 & 0 & 4 \\ 0 & 0 & 4 & 2 \end{pmatrix}$$

2.6 Haralick's Grey Level Co-occurrence Matrices. Shown above are four examples of non-normalized GLCM computed from the 5×5, 4-tone input shown on the left. The considered pixel adjacency relations all consider a same pixel distance $d = 1$, but differ in their angular component α. Haralick then gathers statistics over the coefficients of such matrices as distinctive texture features.



2.7 Haralick's Probabilistic Image Transform [Har79]. Built upon his Grey Level Co-occurrence Matrices, Haralick's textural transforms are defined up to a specific pixel adjacency relationship and a mapping f. Here, f coincides with the identity function, making the pixels of the resulting images store 2nd-order tonal statistics.

where $g_1, g_2 \in \mathscr{G}$ represent grey levels, and $\mathbf{p}, \mathbf{q}$ pixel locations. GLCM encode the local spatial distributions of the image's tonal values. Once normalized, they store the joint probabilities of its grey levels for a set of predefined pixel layouts, and the summation of the coefficients on their rows or columns corresponds to the marginal probabilities of a specific tone. Rotation invariance can be achieved by combining the coefficients established for different angles sweeping through 180° as in Figure 2.6. Haralick then extracts fourteen features for describing the textures and improving classification performances, *e.g.* uniformity, entropy, correlation, contrast, correlation, or the probability of a run of length for a specified tone. Difficulties arise from the high dimensionality of the resulting feature vectors and the redundancy between their components once all distances and angles have been considered. Spectral extraction methods have been proposed to handle this issue.

Back in 1979, Haralick already defined textures as a two-layer entity, and emphasized the need of proposing appropriate levels of inspection for their analysis. He presented textures as the spatial organization of tonal primitives, and stated that the identification of such primitives and the discovery of their organization could be seen as different albeit complementary tasks. He proposed a *textural transform* consisting in the construction of a new image J whose pixels indicate how frequent a specific spatial layout between tones occurs in the image at their corresponding location [Har79]:

$$J_{d,\alpha}(\mathbf{p}) = \frac{1}{Z} \sum_{\substack{\mathbf{q} \\ \mathrm{dist}(\mathbf{p},\mathbf{q})=d \\ \mathrm{angle}(\mathbf{p},\mathbf{q})=\alpha}} \sum f\left(\mathrm{GLCM}_{d,\alpha}\left(I(\mathbf{p}), I(\mathbf{q})\right) \right),$$

where Z is a normalized constant. As shown above, a textural transform is defined up to the layout of the GLCM on which its computations are based, and a mapping f. Examples of such transforms are provided in Figure 2.7. Haralick's approach is somehow related to ours as rather than tonal primitives, we strive to localize more complex shapes and capture their spatial organization. However, Haralick's implicit definition of his primitives lacks the structure information necessary to handle our targeted inputs. They are indeed solely determined by the pixel adjacency relation of the GLCM with which the transform has been established.

Many earlier techniques for texture characterization and discrimination are thoroughly presented in Haralick's overview [Har79] and alluded to from the perspective of his two-fold definition of textures. Among these, analysis techniques extracting features from the distribution of spatial frequencies, or using morphological operations with parameterized structural elements over binary inputs are of much interest. First per-pixel

scan-line synthesis experiments using auto-regressive models are also discussed. Their computation of new pixel values from the linear combination of previously-synthesized values and random noise values is highly reminiscent of modern texture synthesis techniques.

2.2.2 Textures as Markov Random Field Samples

Summarizing co-occurrences of tones via a set of moments [Jul62] or matrices [HSD73] can be seen as a non-parametric way for locally encoding the joint probability of the input's pixel values. Research in statistical modeling explicitly handles such quantities and proposes a complete mathematical framework providing closed-form expressions for probabilities estimated over images. Key features of probabilities are the fact they correspond to measurements established over *spatial processes*, and that pixel values are interdependent. Having to resort to high-order statistics to get efficient texture features reveals the need to capture complex tonal interactions between pixels. Research has been conducted for explicitly formulating the joint probabilities of systems of made of spatially-interacting variables. Results such as the ones devised by Besag in 1974 are hence highly relevant for statistical image modeling which aims at capturing the input's salient features by finding an appropriate probability distribution model [Bes74].

From a statistician's point of view, an $m \times n$ image can be seen as a random field defined over a lattice structure $\mathscr{L} = \{i = (x, y);\ 1 \leq x \leq m \bigcap 1 \leq y \leq n\}$. Each pixel is seen as a random variable S_i whose assigned value s_i corresponds to either its grey tone or color. Any image provided as input $\mathbf{g} = \{g_i;\ i \in \mathscr{L}\}$ then represents one of the possible realizations of this 2d Random field. Since its variables are statistically dependent, any image produced by this model is determined by its associated joint probability $\mathrm{P}(\mathbf{g}) = \mathrm{P}(S_i = g_i;\ i \in \mathscr{L})$.

Once fitted to the input image $\mathbf{g}$, the model's joint probability should ideally behave in the space of possible configurations Ω like a Dirac function centered on the training image (close to 0 everywhere except around a subset of images close to the image of interest $\mathbf{g}$). One could thus try to force the field's joint probability to match the image's, or some parametric function approximating it. The problem however if one considers the image in its entirety along with all the dependencies of its pixels, as the exploration of the space of all possible images Ω is of combinatorial complexity.

Besag studied such statistical systems and proved their joint probability distribution is uniquely determined by the *conditional* probabilities of its individual variables S_i. The joint probability ratio of two realizations $\mathbf{g}, \mathbf{f} \in \Omega$ of the random field can indeed be expressed as the product of interleaved conditional[1]:

$$\frac{\mathrm{P}(\mathbf{g})}{\mathrm{P}(\mathbf{f})} = \prod_{i \in \mathscr{L}} \frac{\mathrm{P}\left(g_i \,|\, g_1, \ldots g_{i-1}, f_{i+1}, \ldots f_{|\mathscr{L}|}\right)}{\mathrm{P}\left(f_i \,|\, g_1, \ldots g_{i-1}, f_{i+1}, \ldots f_{|\mathscr{L}|}\right)}.$$

By considering the trivial realization $\mathbf{0}$ for which all values equal zero, he obtains the specification of a single realization $\mathbf{g}$'s joint probability:

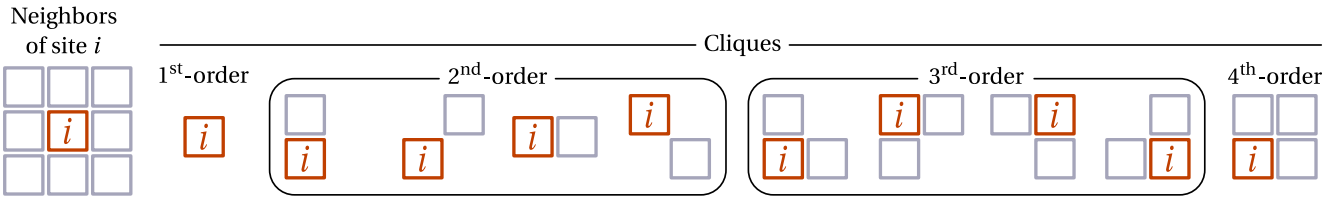
$$\mathrm{Q}(\mathbf{g}) := \log \frac{\mathrm{P}(\mathbf{g})}{\mathrm{P}(\mathbf{0})}.$$

This statement is capital as it allows the specification of the *global* joint probability by a set of *local* conditional probabilities. This is all the more convenient as the pixel grid structure of images provides a natural neighborhood system guiding their dependencies. Hence, in practice, most image probability models only consider dependencies between nearby pixels, and assume the *markovianity* of the image: each pixel is not dependent on the set of *all* the other pixels anymore, but on a limited set of neighbors. This property can be expressed as:

$$\mathrm{P}\left(s_i \,|\, s_j;\ j \in \mathscr{L} - \{i\}\right) = \mathrm{P}\left(s_i \,|\, s_j;\ j \in \mathscr{N}_i\right),$$

where $\mathscr{N}_i$ contains the indices of the ith site's neighbors. Modeling textures as specific realizations of a *Markov random field* therefore lifts the initial intractability of the joint probability specification problem in the general case.

[1] This expression is valid for any random field, the only constraint being the positivity condition $\mathrm{P}(s_i) > 0 \forall i \in \mathscr{L}$ satisfied in most cases.



2.8 Cliques for a 8-Connectivity Neighbor System. Cliques correspond to sets of mutually-neighboring pixels. Texture models usually consider pair-wise interactions for alleviating computational complexity.

An appealing property of a Markov random field is the possible expression of its joint probability distribution as the product of its *clique potentials*. In such cases, the random field's joint probability can be uniquely expressed as:

$$\mathrm{Q}(\mathbf{g}) = \sum_i g_i Q_i(g_i) + \sum_{i<j}\sum g_i g_j Q_{i,j}(g_i, g_j) + \sum_{i<j<k}\sum\sum g_i g_j g_k Q_{i,j,k}(g_i, g_j, g_k) + \ldots + g_1 \ldots g_L Q_{1,\ldots L}(g_1, \ldots g_L),$$

where the Q-functions reflect the neighborhood relationships between the field's variables. Indeed, $Q_{i,j,\ldots k} = 0$ if and only if the variables $s_i, s_j, \ldots s_k$ are mutual neighbors, or alternatively speaking, form a clique in the graph whose nodes correspond to the variable sites, and edges to their dependencies. From it, Besag reaches this last expression:

$$\mathrm{Q}(\mathbf{g}) = \sum_{C \in \text{cliques}(\mathscr{L})} \phi_C(g_i;\ i \in C) \quad \Rightarrow \quad \mathrm{P}(\mathbf{g}) \propto \exp\left(\sum_{C \in \text{cliques}(\mathscr{L})} \phi_C(g_i;\ i \in C)\right) \tag{2.1}$$

where ϕ_C designates the potential function of the clique C. It results that the joint probability of a Markov random field is in fact a Gibbs distribution, an equivalence known as the Hammersley-Clifford theorem. All the previously-established results still assume the compliance of textures to the Markov property which still does not come as a very restrictive prior since no requirement on the order of the cliques are expressed (*cf.* Figure 2.8).

The expressiveness of such statistical texture models lies in the choice of the considered cliques and the specification of the energy $\mathrm{Q}(\mathbf{g})$. In practice, most texture models involve cliques of 1st and 2nd orders only, and their energy is often greatly simplified:

$$\mathrm{Q}(\mathbf{g}) = \sum_i \alpha_i(g_i) g_i + \sum_{(i,j)}\sum \beta_{ij} g_i g_j.$$

Such energy functions characterize to *auto-models* which can differ in their neighborhood system and the expression of their pixel interactions. They are then fitted to the training texture sample by finding the optimal set of parameters (α_i) and (β_{ij}) that maximizes the likelihood of occurrence of the observation by the model. This technique, extremely commonly used in statistical learning, is called Maximum Likelihood Estimation.

Part of the domain knowledge can be embedded into the model itself by choosing different kinds of pixel interactions. Often-encountered pixel conditional probabilities are the binomial distribution (for discrete images) and the normal distribution (for continuously-valued images). They respectively give rise to the auto-binomial and auto-normal texture models whose optimal parameters can easily be obtained by auto-regression. But these models clearly lack expressive power as shown in Figure 2.9. Indeed, the cliques auto-models rely on are too small, the features they capture spanning a couple of pixels at best. The assumed interactions are also too simplistic as several textures just cannot be represented through Gaussian functions, no matter how wide the neighborhoods are. Finally, Julesz's last experiments show that low-level moments between pixel tones are insufficient to properly describe textures and discriminate between them [JGV78].

Obtaining new textures from statistical models in generalis also a thorny issue. If we take a closer look at the right end of Equation 2.1, we see that the exponential expression should undergo normalization in order to

2.9 Textures as Samples of a 50×50 Gaussian-Markov Field. Above are five examples drawn from an anisotropic auto-normal model involving 1st and 2nd order cliques for various sets of parameters. All images are taken from Pérez' technical report on the modeling of images by random fields [Pér98]. They demonstrate the impropriety of strictly statistical models for representing textures exhibiting pixel correlations that span more than a couple of pixels.

Metropolis algorithm

```
initialize g as a random image
repeat
    compute g' by performing random trial move on g
    compute acceptance rate p = min{ P(g')/P(g), 1 }
    if random (0,1] < p then replace g by g'
until equilibrium is reached.
```

Gibbs sampler

```
initialize g as a random image
repeat
    for all pixel i
        for all grey level g
            compute p_g = P(g_i = g | g_j ∈ N_i)
            if random (0,1] < p_g then assign g to i
until equilibrium is reached.
```

2.10 Pseudo-Codes of Principal Monte-Carlo Markov Chain algorithms. Since they enable the creation of Markov chains converging to any distribution probability, MCMC algorithms are widely used for the re-synthesis of new images by sampling a learned distribution function. Note that the Metropolis version involves *ratios* of joint probabilities in order to avoid the issue of untractable normalization.

represent a valid probability density function. However, the explicit normalization would require the complete traversal of the configuration space Ω which grows exponentially with respect to the number of pixels and colors. Since this normalization is untractable, $\mathrm{P}(\mathbf{g})$ is defined up to a constant, and special care is therefore required during the simulation of the model.

Monte-Carlo Markov Chain (MCMC) approaches are commonly used to cope with the issue of getting realizations from statistical models and make possible the sampling from any distribution by the iterative construction of a Markov Chain whose equilibrium distribution matches the specified distribution. Common techniques include the Metropolis algorithm which explores the entirely configuration space by random walk, and the Gibbs sampler that enables the sampling of joint distributions unamenable to direct evaluation by manipulating univariate conditional probabilities. The pseudo-codes for those two methods is indicated Figure 2.10. Robust with respect to the quality of their initialization, they are also slow to converge and no obvious stopping criterion exists. Their main drawbacks are their slowness and the fact that no obvious stopping criteria exist.

Besag's auto-models mostly suffer from their predetermined clique topology and interactions which make them unsuited to most textures and directly sensitive to grey level shifts. Gimel'farb suggests several improvements of such parametric models by first incorporating multiple (short- and long-range) pairwise interactions so that the model could adjust to wider pixel correlations [Gim96a]. His 2th-order cliques are instead defined by a given offset vector (c_x, c_y) and are mathematically expressed as $\{(x_1, y_1), (x_2, y_2) \in \mathscr{L};\ x_1 - x_2 = c_x \cap y_1 - y_2 = c_y\}$. All the cliques of Gimel'farb's model are associated potentials which constitute its parameter set. An interesting side-effect is that Gibel'farm can extract an *interaction map* from the input texture, by thresholding these learned values and recording the clique offsets associated with the strongest interactions (*cf.* Figure 2.11). Even though Gimel'farb admits himself that linking such maps to perceivable sample's structures is not obvious, this attempt at extracting intermediate visual information from the input is anticipates future approaches, including ours. Gimel'farb also addresses the sensitivity of simpler Markov models to grey fluctuations and is confronted to the difficulty that basic grey range transformations usually involve non-local interactions compromising the input's markovianity. Gimel'farb first bypasses this issue by expressing his potentials as functions of grey differences [Gim96a], and later proposes a *non-Markov*

2.11 Markov Texture Model Involving Multiple Pairwise Interactions [Gim96a]. Gibel'farb created a parametric texture model whose joint distribution follows a Gibbs distribution while accounting for wider 2nd-order interactions and limiting the influence of grey level shifts. His interaction maps also reveal some cues about the input's structure.

texture model. Thanks to the fact global interactions remain scarce, this model's joint probability is still a Gibbs distribution [Gim96b]. Gimel'farb's models improve previous results in the sense they incorporate anisotropic, almost-natural textures, but can only support uniform inputs (*cf.* Figure 2.11).

2.2.3 Constrained Texture Models

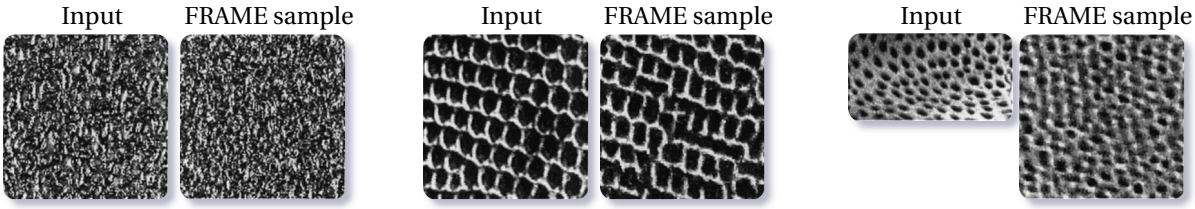
A large number of more recent methods for texture recognition, retrieval and synthesis find theoretical justification in Julesz's, but also in the computational representation of the front-vision processes proposed by the filtering theory. Instead of local joint statistics, they rely on a more elaborate representation based on statistical measures estimated over filtered versions of the input. Early stirrings of filtering theory emerged with Faugeras and Pratt's experiments involving autocorrelation-based marginal statistics [FP80], followed by the application of basic linear filtering operations for acquiring surprisingly efficient discrimination cues [BA88]. This line of research progressively led to the modeling of the human visual system's decomposition of the retinal image into sub-band images as the convolution with a bank of linear filters followed by non-linear processing [Dau85]. The action of the retina's orientation/frequency-selective simple cells is approximated by Gabor filters and center-surround ganglion cells by Laplacian filters, whereas the simulation of more complex cells is carried out by non-linear filters.

Work conflicting with Julesz' conjecture suggest to consider local densities of "micro-pattern features" such as corners, intersections or terminations [CB78a]. Their evaluation still requires the cumbersome extraction of these features. Yielding a comparable discriminative power, filter response statistics hence represented a welcomed alternative. Indeed, filtering comes at a much lower computational cost and also enables the characterization of the input's long-range structures. Similarly to the Julesz's conjecture, filtering theory has nevertheless also revealed its limits, recent experiments conducted by Gluckman exhibiting distinguishable images of equal marginal and joint statistics of the responses to Gaussian, Laplacian and derivative filters [Glu05].

In parametric model-based texture synthesis, analysis comes down to the estimation of the set of parameters that best "fit" the training image. A convenient way of doing so is by assessing that this optimal set corresponds to parameters that maximize the entropy of the model's distribution, while being subject to constraints ensuring the visual closeness with respect to the input image. This assumption is referred to as the Maximum Entropy principle (MaxEnt) and can be formulated as a constrained optimization problem:

$$\mathrm{P}^* = \underset{\mathrm{P}}{\arg\max}\left\{-\sum_i \mathrm{P}_i \log(\mathrm{P}_i)\right\} \quad \text{such that} \quad \mathrm{E}_{\mathrm{P}}\left[f_k\right] = \mu_k, \forall k \in \{1,\ldots K\}$$

where f_k is one of the K features evaluated over images, and $\mathrm{E}_{\mathrm{P}}[.]$ is the expectation operator with respect to the distribution P. P_i designates the ith site's conditional probability, and the constraints (μ_k) correspond to the values of the features observed in the example. The feature functions f_k are to capture the visual "feel" of the training sample **g**, and their enforcement over the output is to grant it faithfulness with respect to the input. Examples of features include the coefficients of the projection of the image onto an orthogonal basis, or its responses to a filter bank. In practice, this optimization scheme can be theoretically solved using Lagrange multipliers $(\lambda_1,\ldots\lambda_k)$, but closed-form solutions are often unavailable as the features gain in complexity. In-



2.12 Textures Obtained by the FRAME Image Model [ZWM96]. Zhu *et al.* use a constrained optimization framework to enable more expressive texture synthesis than regression on auto-models. They use a two-stage iterative procedure to find the minimal set of features whose enforcement is required to achieve satisfactory synthesis.

terestingly, the optimal distribution remains a Gibbs distribution:

$$P^* \propto \exp\left(-\sum_k \lambda_k^* f_k\right).$$

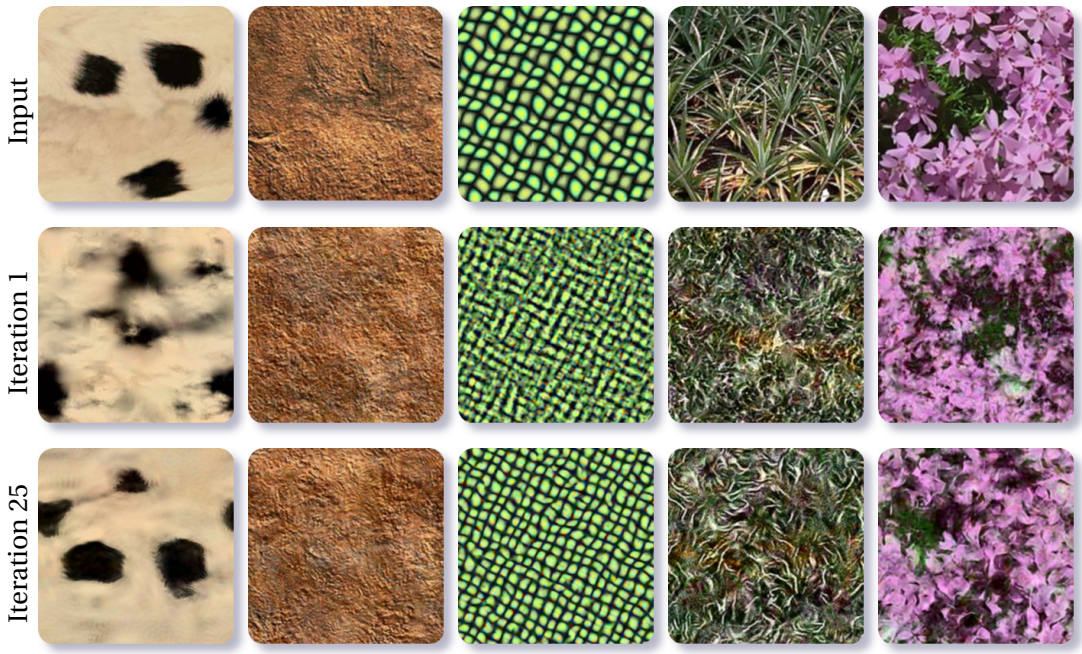
The texture model hence still corresponds to a valid Markov random field whose cliques span the feature functions' domains and can account for wider structures.

The FRAME system (acronym for Filters, Random Fields and Maximum Entropy) is a perfect illustration of such a statistical regression scheme for texture modeling [ZWM96, ZWM97]. Zhu *et al.*'s framework is two-fold and iterative: the first stage – feature extraction – selects a relevant set of filters as features so that the image is well captured, while the second stage – feature fusion – updates the output distribution so that its features' marginal distributions match the training image's. Instead of exploring the space of all possible filter combination, Zhu *et al.* rely on a greedy filter selection that adds one filter at a time, the picked filter corresponding to the feature considered as the most informative for the distribution at the current step. Their bank of filters comprises intensity-based filters, Laplacian-of-Gaussian filters, Gabor filters, and powers of pairs of Gabor filters. Feature fusion then updates the current models' parameters while respecting the set of constraints imposed by the selected features' marginal distributions. Examples of achieved textures are given in Figure 2.12.

To the best of our knowledge, the most recent and successful parametric texture model is the one proposed by Portilla and Simoncelli [PS00], which resorts to a more elaborate set of feature functions based on the joint statistics of the training image's wavelet coefficients. They also generalize Julesz's conjecture to give it a broader scope by assuming the existence of a set of perceptually-relevant measurements (not limited to joint statistics anymore) such that two homogeneous textures are indistinguishable, if and only if they result from random fields equal in term of those measurements. Like for Zhu *et al.* 's FRAME model, they devise an iterative synthesis procedure where each constraint is enforced sequentially. The output is obtained after projection onto the input's Julesz Ensemble defined as $\{\mathbf{g};\ \mathrm{E}\left[f_k(\mathbf{g})\right] = \mu_k\ \forall k\}$, *i.e.* the set of all images sharing the same statistical constraints as the input $\mathbf{g}$. The projection is guided by the direction of the gradient of the constraint functions. Their feature set is imposed at each level of the steerable pyramid from which the measurements are gathered [SFAH92, SF95]. It includes the first three moments and range of the pixel intensities, the local auto-correlation between wavelet coefficients, correlations of the complex magnitude of neighboring (in terms of position, orientation and scale) coefficients, and statistics of the cross-scale phase. This model does yield impressive results considering that only statistical constraints are being manipulated (*cf.* Figure 2.13). Their parametric model is still fairly complicated to implement and use in practice for versatile texture synthesis, but its feature selection provides valuable insights for texture classification and recognition.

2.2.4 Texture Synthesis by Multi-Scale Statistics Transfer

First texture parametric models suffer the limited scope of their achievable outputs and the fact their predetermined probabilistic pixel interactions are rarely adapted to the input's actual salient features [Bes77]. Joint distributions obtained by constrained synthesis [ZWM96, ZWM97, PS00] can account for wider correlations between pixels, but at the price of a much more involving process.

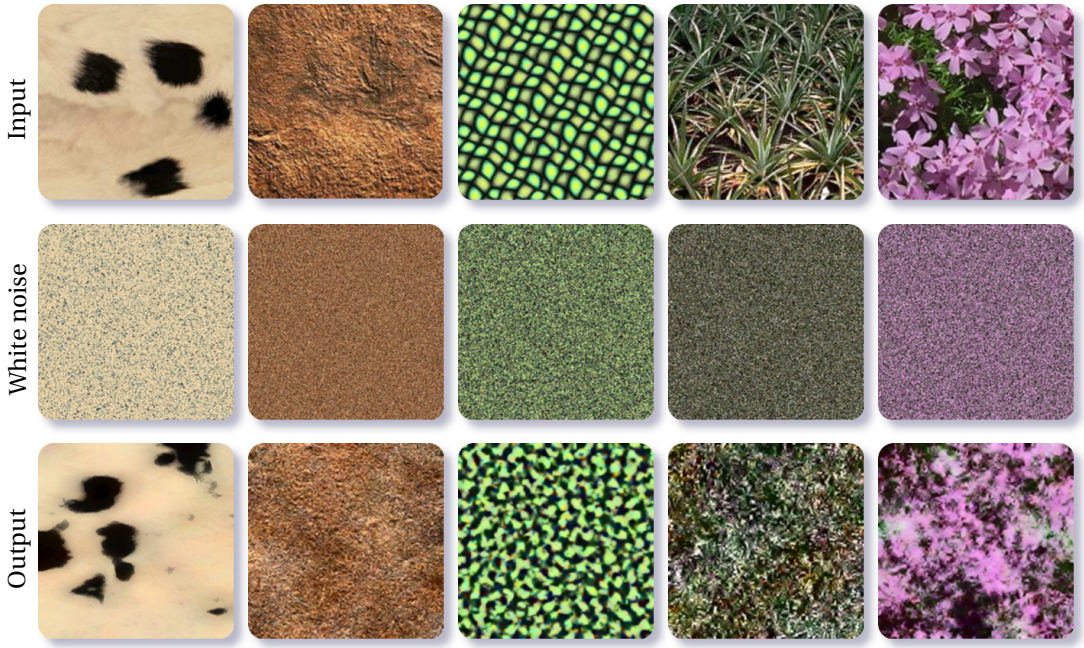


2.13 Re-Synthesis Examples using Portilla and Simoncelli's Model [PS00]. Constraining the joint statistics of the textures' wavelet coefficients, Portilla and Simoncelli's approach is one of the most convincing techniques relying on the sampling of a fitted joint distribution over a lattice.

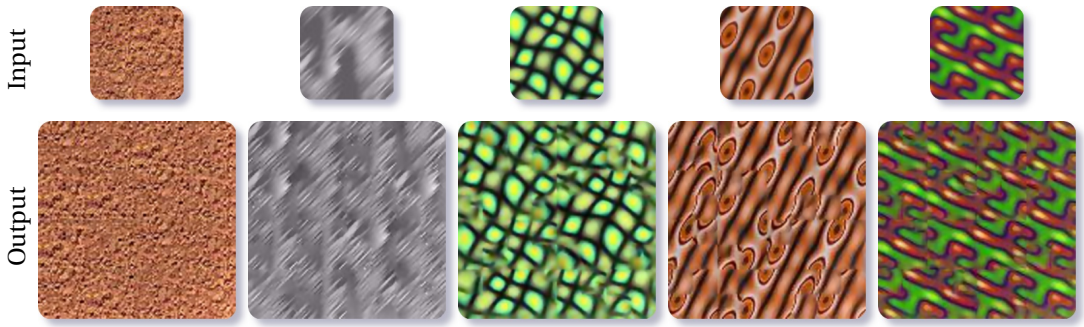
Other joint distribution models do exist, but their specification is carried out in non-parametric way: in place of analytic distribution formulae, are manipulated histograms of filter responses that approximate their distribution functions. Such models represent the first steps towards modern synthesis procedures and tackle the issue of capturing inhomogeneities, as well as long-range structures in the spatial or spectral domain.

A first illustration of those principles is Heeger and Bergen's technique that makes a noise image resemble the input by constraining the 1st-order statistics of the coefficients of its wavelet decomposition [HB95]. The output's Laplacian pyramid is directly initialized with the decomposition of the noise image and altered in ways such that it becomes comparable to the input's by matching at each level the coefficient histograms. After all levels have been transformed, the output pyramid's collapse yields the final texture. Their methods proposes promising results at the light of its simplicity with respect to parametric techniques such as Portilla's (*cf.* Figure 2.14), to the point of straightforwardly lending itself to the synthesis of solid textures. But its direct reliance on 1st-order statistics overlooks spatial correlations between coefficients. Their approach thus only works for homogeneous textures and cannot perform well on near-regular inputs.

For capturing the input's inhomogeneities, De Bonet inspects the behavior of chains of the steerable pyramid's coefficients across the scales [dB97]. He models the generation process of these chains in such a way that the generation of the lower levels (finer scales) is dependent on the higher levels (coarser scales) of the wavelet pyramid. This is formulated in the conditional probability distributions linking the different coefficients of a same chain. These probabilities are again estimated in a non-parametric manner, as the ratios of densities evaluated over Parzen windows. Textures are then modeled as the spatial arrangements of such coefficient chains and are assumed to stem from a spatially-ergodic process (which means that the chains' joint probabilities do not depend on their positions in the image). This representation proves to be notably successful for texture recognition [dBV98] and image retrieval [dBV97]. For texture re-synthesis, De Bonet *et al.*'s capture inhomogeneities via the coarse levels' wavelet coefficients: new textures are generated by re-organizing the low frequency components, the input's visual characteristics being conveyed by the observance of its high frequencies. They infer the output's pyramid in a top-down manner from the shuffled low frequencies thanks to a sampling procedure of the input pyramid that is locally constrained by the coefficient chains from the higher levels. Perceptual dissimilarity is quantified by the sum of the squared distances between coefficient chains. Results are provided in Figure 2.15.



2.14 Pyramid-Based Texture Synthesis [HB95]. Above are illustrations of textures synthesized by enforcing the steerable pyramid coefficient distribution of the provided samples (*top*) onto white noise images (*middle*). The obtained results are satisfactory for "blobby" or purely stochastic textures (*bottom, left*), but this representation fails to reproduce more regular (*bottom, middle*) or even natural, inhomogeneous inputs (*bottom, right*).

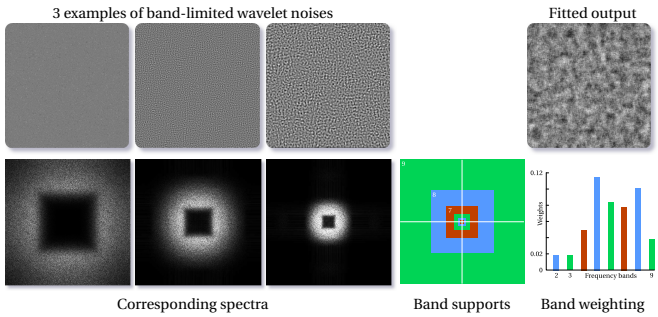


2.15 Texture Synthesis by Multi-Resolution Sampling [dB97]. De Bonet's model of textures takes a special care of accounting for the cross-scale correlations between pixel colors. It captures and restitutes the joint distributions of chains of wavelet coefficients built across the pyramid's levels. While it model performs better than previous parametric models, noticeable discontinuities are apparent.

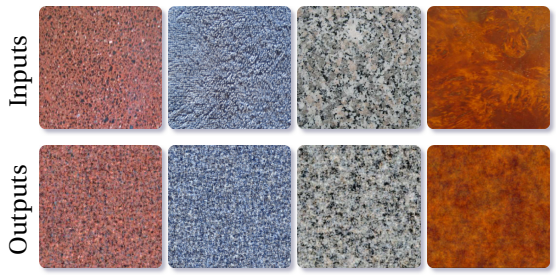
2.2.5 Statistical Transfer for Example-Driven Procedural Texturing

As mentioned in Section 2.1, procedural texturing garners criticism for its steep mastering curve. Either at programming or at designing textures by way of the set of parameters provided by the model, these techniques require much patience and effort to take the full benefit of their expressive power. Since it requires artists to undergo long sessions of parameter tweaking, procedural texturing is often left under-exploited and its actual use confined to filler textures.

Few research is dedicated to the automatic assignment of procedural models' parameters in order to have them best mimic a provided input sample. To the best of our knowledge, the most noteworthy attempts both in terms of generality and success, are the ones conducted by Ghazanfarpour and Dischler [GD95], and Lagae *et al.*'s [LVLD09]. Both techniques base their learning procedure on the spectral analysis of the provided sample, but while Ghazanfarpour builds upon Perlin's texture model (composed of a basis function modulated by a turbulence function), Lagae exploits multi-resolution wavelet noise.



2.17 Isotropic Procedural Textures By-Example [LVLD09]. Lagae *et al.* estimate the best-suited contributions of the different bands for their multi-resolution wavelet noise by transferring statistics of the input's average power spectra.



2.18 Results from Lagae *et al.*'s Fitting Method [LVLD09]. Multi-band transfer of power spectrum statistics performs well for a wide range of material textures. However, their method is intrinsically limited to isotropic textures.

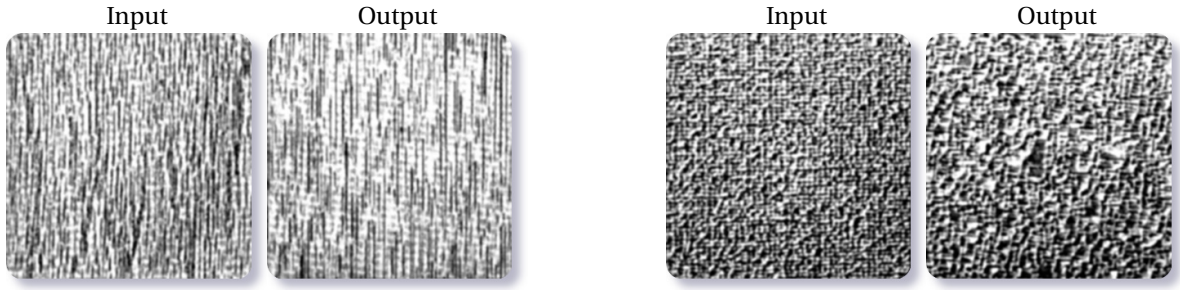
Ghazanfarpour and Dischler determine the basis and turbulence functions that best convey the input's spectral characteristics by the study of its Fourier transform [GD95]. The basis function is obtained by the summation of the cosines associated with the frequencies whose amplitude exceeds a specified threshold. Ghazanfarpour then finds a proper turbulence function in the convolution of a white noise with a piecewise constant transfer function accounting for all significant frequencies gathered after the partitioning of the spectral domain. Even though the resulting 3d volume is constrained by a 2d sample, his method works well for textures whose spectral energy is locally concentrated as shown by the impressive example Figure 2.16.

2.16 Example-Fitted Perlin Noise [GD95].

Lagae *et al.* instead consider multi-resolution wavelet noise defined as the weighted sum of shifted wavelet noise functions evaluated at different octaves. While the maximal number of possible octaves is bound to the resolution of the user-provided texture, the contributions of the different frequency bands can be freely adjusted. Thanks to the closed-form expression of their noise's Fourier transform, Lagae *et al.* use the Parseval theorem to find the weights that best transfer over the output noise the observed contribution of each frequency band. The only approximation of their statistics transfer is the use of the bands' expected average power in place of their actual power because of the inseparability of the input's frequency bands in the general case. The amplitudes of the frequencies from different bands are collected over their theoretical supports (as shown in Figure 2.17) and are stored into a power histogram divided into frequency bands. Once the weights are established, Lagae lastly coerces the color distribution to resemble the input's using histogram matching in de-correlated color space. Their method is one of the most general attempt at example-driven procedural synthesis and is successful for a wide range of isotropic textures (*cf.* Figure 2.18).

2.2.6 Synthesis by Non-Parametric Sampling

Previously-mentioned techniques relying on multi-scale statistical transfer do generate promising results [HB95, dB97], often of a superior quality than the ones of theoretically more accurate approaches [ZWM96, ZWM97, PS00]. And in spite of their rough approximations produce more convincing results. The progressive distance with respect to mathematical exactitude observed in texture synthesis is mainly due to how quickly untractable the enforcement of global statistics gets. The assumption of the textures' Markov property enabling the resort to local statistics as good substitutes for their global counterparts enabled the fulfillment of accurate albeit spatially-limited constraints. The constraining of filter response statistics finally led way to non-parametric specification of conditional statistics via histograms. Surprisingly, even though the problem seems as reduced, the quality of the results attests that these successive simplifications



2.19 Dawn of Non-Parametric Synthesis [GS81]. Garber first proposed a sampling strategy guided by non-parametric constraints. He uses the visual distance between causal neighborhoods as an estimate of the conditional probabilities of the output pixels. His results were stunningly good back in 1981, however daunting computation times (55 days for a 512×512 output constrained by a 55×55 neighborhood) hindered his technique from being widely used.

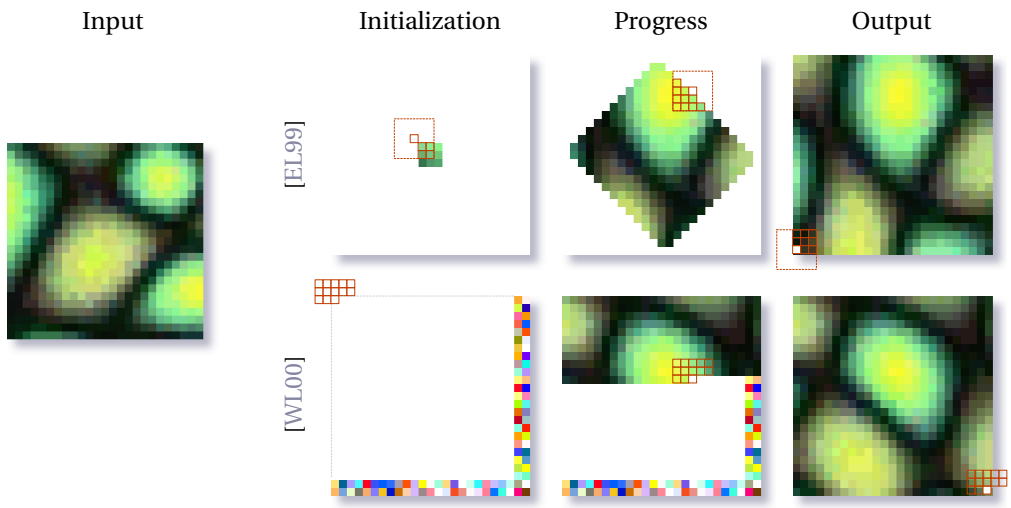
by no way impoverished it. Research in Computer Vision further confirmed this trend by achieving stunning results through even simpler non-parametric methods.

2.2.6.1 Non-Parametric Pixel-Based Synthesis

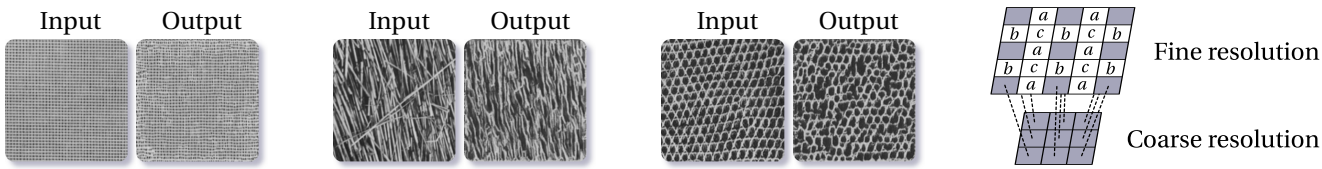
The incredibly simple-sounding approach devised by Efros and Leung in 1999 sure constituted a capital turning point [EL99]: instead of constructing a statistical texture model, parametric or not, they synthesize their outputs by directly copying pixel values from the input. Their sampling scheme is controlled by the pixels' conditional distributions indirectly represented by the concatenation of the neighbors of the pixel of interest. The output texture is created from one single seed in a greedy fashion, one pixel at a time (*cf.* Figure 2.20 *(top)*): when an output pixel needs to be assigned a color, they build a vector out of its already-synthesized neighboring colors and traverse the input image in search for similar vectors. Dissimilarity between such vectors is computed as the sum of the squared differences between the (r, g, b)-triples. The set of candidate colors correspond to vectors whose color difference with the query output vector lies below a specified threshold. They then randomly pick one feature vector among this candidate set and color the output pixel accordingly to the chosen vector's central pixel. By directly manipulating pixels, this method can reproduce higher frequency effects than previous attempts, and by considering local neighborhoods, can preserve spatial correlations often lost amidst filtering. This methods work well for a broad scope of textures, but its quadratic complexity makes it time-consuming. The ever-changing shapes of the query vectors alas prevent the use of common optimization structures. Moreover, the algorithm's greedy nature may endanger the quality of its results as it may fall into a local minimum and get stuck in a "wrong part of the search space". Lastly, the size of the preserved features depends on the size of the neighborhoods used for the construction of the feature vectors and is to be specified by the user.

Among automation, the viability of texture synthesis methods directly depends on their execution time and trading theoretical complexity to algorithmic complexity may prove to be dreadful. Back in 1981, Garber proposed a method similar to Efros' which has been overlooked as it appeared untractable at the time [GS81]. Confronted to the issue of the storage and the reliability of the histogram-based estimation of pixels' conditional probabilities, Garber first proposed to resort to pixel neighborhood matching for synthesizing continuous grey-level textures. For each output causal neighborhood, he computes the image of the visual differences with all the input's neighborhoods and then uses it to simulate a random variable with the desired conditional distribution. The computed value then acts as the output's pixel grey level. Impressive results are displayed in Figure 2.19.

The question of the computation costs is thus paramount here as manipulating non-parametric statistical representations requires more computation and storage capacities. Wei and Levoy addresses these concerns [WL00], partly by considering multi-scale neighborhoods computed across Gaussian pyramids' levels which indirectly encode wider neighborhoods at a lower computational cost. But it is by trading Efros' spiral traversal of the output pixels by a scan line traversal, and thus only manipulating fixed-size neighborhoods (*cf.*



2.20 Importance of the Neighborhood Shape. More than its size, the shapes of the conditioning neighborhoods have a direct impact on the performances of a non-parametric sampling technique. The only constraint over is that they only contain already-generated values. While Efros uses neighborhoods whose shapes vary depending on the output's current state [EL99], Wei resorts to random values at initialization but keeps his neighborhoods' shapes constant [WL00].



2.21 Semi-Parametric Modeling for High-Dimensional Probability Functions [PP93]. Popat fits his probability function on the set of all possible input neighborhood vectors, and generates new textures in a scan line traversal order. Previously-synthesized pixels naturally condition new output pixels' values. All examples are obtained by maximum-likelihood interpolation using the learned conditional probability function. Popat's multi-scale approach distinguishes itself by the classification of pixels depending on their position relatively to coarser resolutions. Each category has its own separate trained model.

Figure 2.20 *(bottom)*), that they manage to greatly accelerate re-synthesis via the dimensionality reduction of the now-bounded search space and its embedding in acceleration structures.

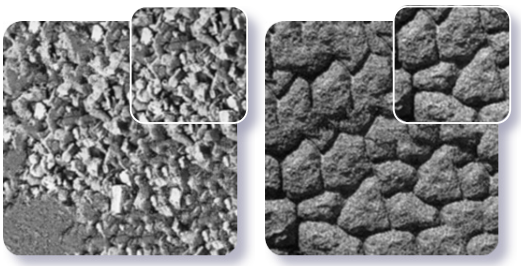
Efros' and Wei's techniques bear a strong resemblance to Popat and Picard's texture synthesis approach [PP93]. Popat and Picard initially considered texture synthesis only as one of the many applications of their statistical modeling of high-dimensional joint probability distributions. They propose a data-driven model built upon the clustering of the observed data in a way reminiscent of the fitting of parametric model mixtures. Their probability model is first trained over the collection of all the causal neighborhood vectors from the input. They then construct output textures in a scan line fashion, by generating new pixels values according to the conditional probability distributions imposed by their respective causal neighborhoods. They also propose a hierarchical variant of their method, characterized by the use of several probabilistic models interleaved across scales (*cf.* Figure 2.21 *(right)*).

Pixel-based synthesis has numerous advantages, the first one being its increased controllability. These allow many variants and extensions, such as constrained synthesis for filling holes in images while ensuring seamless transitions with the constraining boundaries, or even image interpolation [EL99]. Moreover, the highly-optimized pixel-based sampling framework proposed by Wei is perfectly suited to the handling of "*temporal textures*" – spatially and temporally stationary motions such as fire or smoke [WL00]. Furthermore, its overall implementation is simple and elegant enough to transpose it to other contexts, such as the synthesis of textures onto the 3d geometry. By directly synthesizing textures on the surface's tangent planes, issues such as the surface's consistent parametrization and texture placement are naturally eluded. In 2001, Wei and Turk proposed two similar methods for that aim [WL01, Tur01]. Both techniques require the re-tiling/re-meshing

of the surface to ensure a roughly uniform vertex distribution and the construction of a mesh hierarchy mimicking the effect of a Gaussian pyramid of an image. While Wei mostly focuses on generalizing the definition of a neighborhood of vertices by local surface flattening and color re-sampling [WL01], Turk transposes image processing specific routines – interpolation and low-pass filtering – for the geometric case [Tur01]. Only a few user-specified parameters are necessary: first, the vertex density controlling the surface re-meshing directly dictates the size of the output texture's features; and indications of a sparse vector field that orients the local neighborhoods' frames and enables the creation of anisotropic textures. Many other point-based techniques generating appearance attributes directly onto the surface exist [GIS01, YHBZ01, TZL+02, ZG03, ZG04], most of them originating in Efros and Leung's original pixel-based synthesis framework.

2.2.6.2 From Local to Global Optimization

Pixel-based non-parametric sampling methods also have weaknesses, most of which resulting from the greedy nature of their synthesis process[2]. By processing pixel sequentially and independently, Efros' and Wei's techniques fall into this category of algorithms. Their inclination to the accumulation of negligible local errors can result in the exploration of non-representative portions of the search space, or the pursuit of a traversal order creating over-constrained neighborhoods and noticeable verbatim copies of the input's sub-parts (*cf.* Figure 2.22). Resorting to global optimization schemes appears as a natural alternative to cope with such artifacts.



2.22 Drawbacks of Greedy Traversal [EL99].

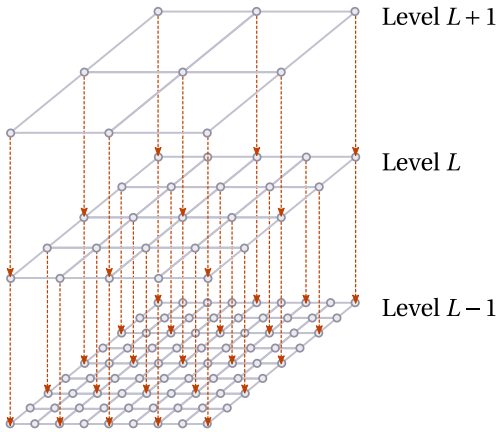
Paget and Longstaff's approach is among the first to cast non-parametric texture synthesis as a non-causal global procedure. They synthesize new textures over a multi-scale Markov Random Field by way of an relaxation scheme that iteratively propagates constraints from coarser levels to the finer ones (*cf.* Figure 2.23). As for simulated annealing, they also assign confidence values to each pixel which, once summed together, describe the uncertainty of the system and control the stopping of the relaxation.

Further investigating the question of synthesis through optimization, Kwatra further investigates the track of texture synthesis by optimization and suggests a technique which voluntarily overlooks the remaining statistical considerations from Popat's method and generates textures as the solutions of an optimization problem solved by Expectation-Maximization [KEBK05]. Although the objective function consists in the summation of *local* matching errors, the E-M algorithm ensures its *global* minimization (*cf.* Figure 2.24). The dual aspect of Kwatra's energy measure, local in its definition, global in its solution, makes for a quite powerful method handling the widest texture spectrum to date. The E-step of the algorithm minimizes the cost function w.r.t. to the output pixels by solving a linear system. The M-step then minimizes the cost function w.r.t. to the input neighborhoods involved in the energy evaluation, and comes down to find the closest input neighborhoods to the generated output pixels. Kwatra's initial method has been later extended to the case of 3d surface texturing and synthesis of volumetric textures, by Han [HZW+06] and Kopf [KFCO+07] respectively. Both improve his method's E-step, by injecting neighborhood coherence to the used linear solver [HZW+06], or by clustering observations for limiting the blurring effects caused by averaging [KFCO+07].

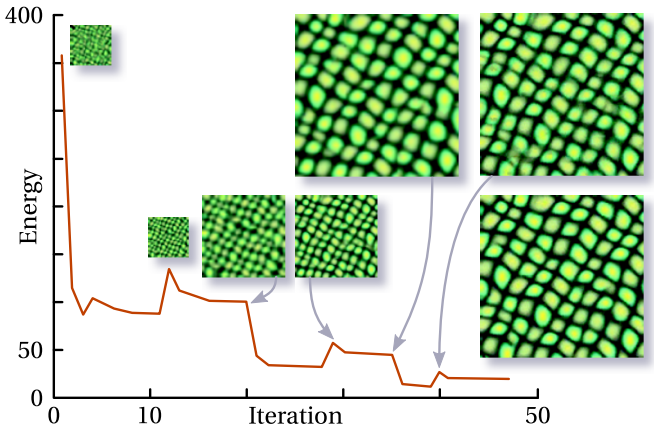
2.2.6.3 Breaking the Output Pixels' Cyclic Dependencies

Efros's and Wei's original greedy synthesis approaches have other, more insidious limitations. Since matched neighborhoods must only contain already-synthesized pixels, they inevitably induces cyclic dependencies between output pixels, the sensitivity to the processing order being another weakness of greedy algorithms. A practical consequence is that such texture synthesis methods are not easily amenable to parallel computing.

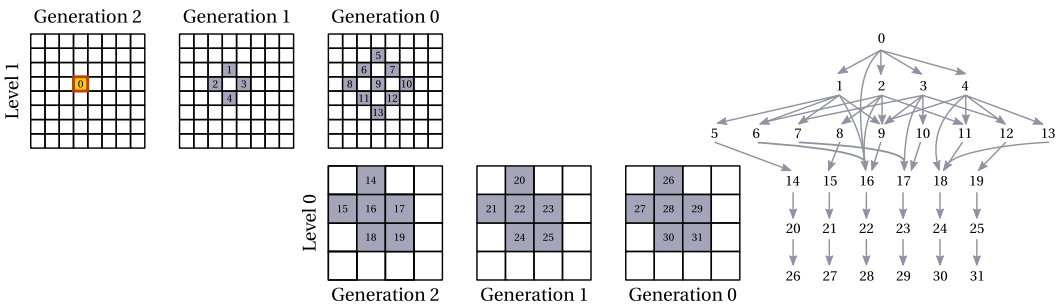
[2] *Greedy* describes optimization algorithms that construct their solutions from successive *locally optimal* choices taken during the course of their execution. They become helpful when it comes down to combinatorial optimization but though efficient, are by no means guaranteed to find the global optimum and end up returning a local optimum instead



2.23 Paget and Longstaff's Multi-Resolution Markov Random Field [PL98]. Paget resorts to a relaxation scheme over the output random field to evaluate its equilibrium state. To accelerate constraint propagation from the local pixel interactions, he proposes a multi-resolution representation whose coarser levels pass their equilibrium values on finer ones.



2.24 Global Optimization-Based Texture Synthesis [KEBK05]. Kwatra *globally* minimizes a cost function defined as the summation of *local* errors using Expectation-Maximization. Energy jumps are caused by the changes in resolution level and matched neighborhood size.

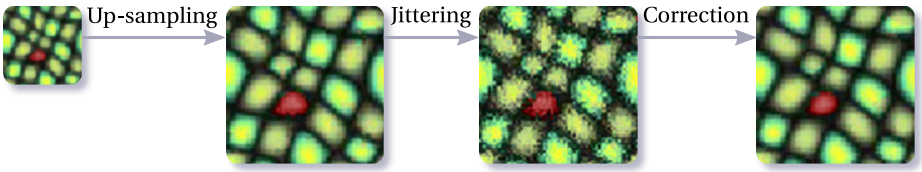


2.25 Order-Independent Texture Synthesis [WL02] . *left:* The shape of the conditioning neighborhood used to assign a value to pixel 0 spans only coarser resolution levels and earlier generations (already-finished texturing passes). Wei's concept corresponds to the unrolling of a multi-pass texture synthesis where the multiply-generated pixel values are stored in separate textures rather than overwritten. *right:* The dependency graph of pixel 0 now does not exhibit any cycle.

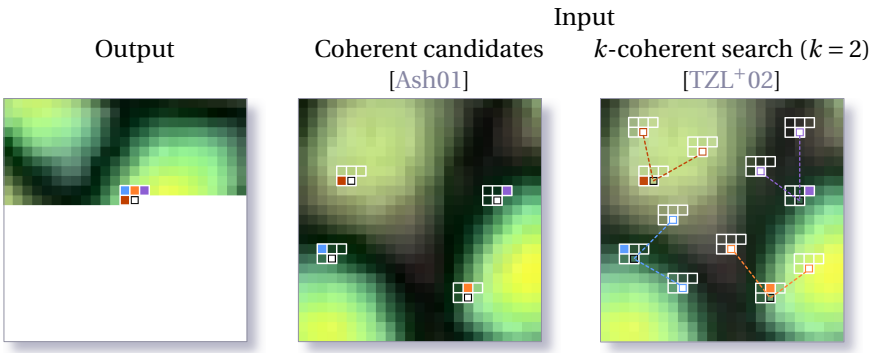
This concern not only limits the possible performance gains, but also compromises their portability to the Graphics Processing Units (GPU).

In unpublished work, Wei and Levoy propose an altered *order-independent* multi-pass technique [WL02]. In order to remove the inter-pixel dependencies caused by the overlaps between nearby neighborhoods, they restrict the conditioning pixel values to correspond to values from previous passes or coarser resolutions as illustrated Figure 2.25. This small modification of the initial pixel traversal scheme allows the independent, and thus potentially multi-threaded, processing of pixels.

Wei also suggests the use of a pyramidal cache structure to take advantage of the spatiotemporal coherence of the texture accesses. While unpublished, Wei's improvements led the way to parallelized pixel-based texture synthesis at the core of nowadays' GPU-based implementations such as the successful techniques by Lefebvre and Hoppe [LH05, LH06]. They formalize GPU-based synthesis as a multi-scale three-stage process. At each resolution level, the current texture coordinates, not colors, are *up-sampled* from the previous level, then *jittered* to introduce randomness, and finally *corrected* to ensure seamless and faithful results (*cf.* Figure 2.26). This last stage directly follows *Wei*'s order-independent framework and its performance gain is so important that they can afford matching pixel neighborhoods in a wider search space represented by their *Gaussian image stack* [LH05].



2.26 Work Flow of Lefebvre and Hoppe's Parallel Synthesis [LH05, LH06]. Each resolution level is processed by three sequential complementary steps: *up-sampling* provides a rough approximate of the finer resolution texture, *jittering* infuses some randomness to the result and *correction* uses Wei's order-independent matching procedure to ensure visual faithfulness to the sample. Note that all involved operations manipulate input coordinates and not directly output colors.

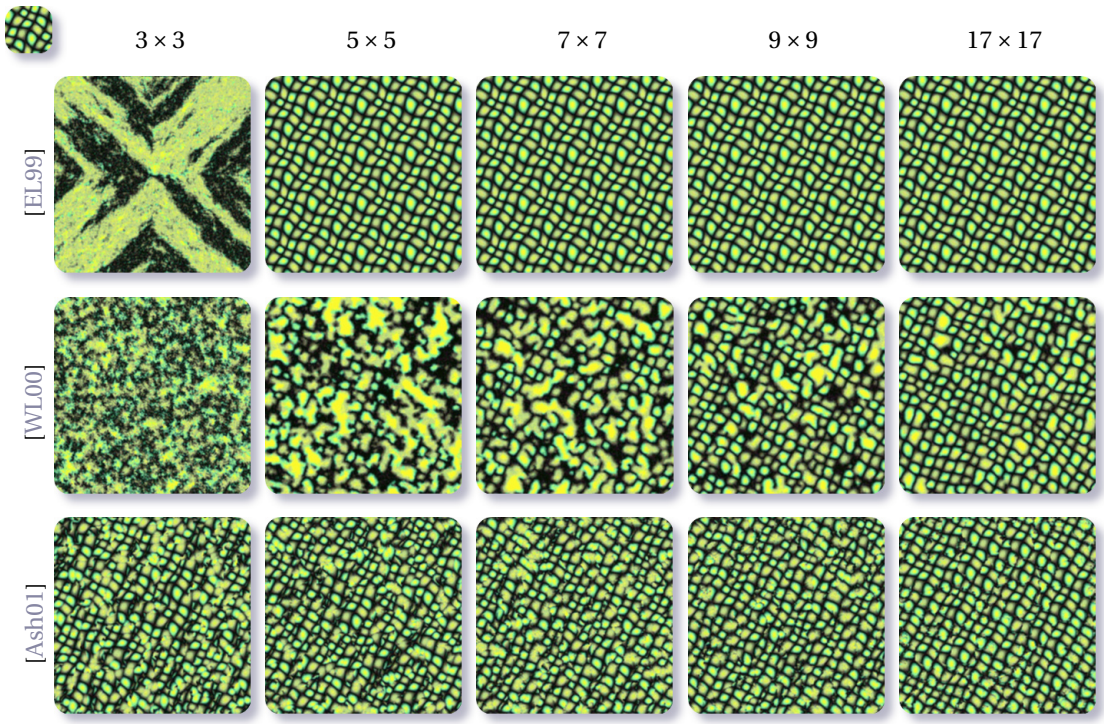


2.27 Coherence-Guided Matching [Ash01]. When assigning an input color to a new output pixel (white pixel on the left picture), coherence directs the choice towards a subset of pixels that correspond to input locations that are close to the ones used to generate its neighbors. Pixels lying at these positions are called the output pixel's *coherent candidates* (displayed in white in the middle picture). More advanced variants include these candidates' nearest neighbors in terms of neighborhood similarity (shown in white on the right) [TZL+02].

2.2.6.4 Coercing Pixel Matching Through Image Coherence

But regardless of their greediness, initial non-parametric pixel sampling methods mostly suffer from their direct manipulation of pixels. Since pixels are independently processed, preserving the input's structures becomes a difficult task and simply increasing the size of the conditioning neighborhoods only constitutes a last resort as it drastically increases synthesis times and results in disturbing repetitions (*cf.* Figure 2.28).

Ashikhmin introduces the concept of *coherence* which accounts for the fact that sets of *nearby* input pixels should also give rise to *nearby* pixels in the output [Ash01]. By doing so, he implicitly embeds the notion of structure to the synthesis process, and introduces a matching bias that favors their re-apparition in the output. Ashikhmin especially strives for improving the quality of the synthesis results of "natural textures" that he defines as "quasi-repeating patterns consisting of small objects of familiar but irregular size" (*i.e.* flower fields, pebbles or tree branches). Indeed, by excessively favoring smooth and seamless transitions, the methods by Efros and Wei often overlook the color discontinuities which are essential visual cues for such shapes, and blend them together. To preserve their boundaries, Ashikhmin tracks the original positions in the input of the matched neighborhoods that were used to generate previous output pixels. He then creates for each new output pixel a set of candidate pixels by considering the positions of the input pixels copied onto its neighbors, and collecting the colors of the input pixels that exhibit with the pasted pixel the same spatial layout as the one formed by the considered output pixel and its redirecting neighbor (*cf.* Figure 2.27 *(middle)*). The creation of such candidate sets is a clever way of exploiting the image's markovianity and the redundancy between overlapping neighborhoods to considerably decrease the size of the search space. Alas, this technique acts more poorly on smooth textures where undue discontinuities appear due to too small candidate sets. Nevertheless, and even though this modified matching departs from the theoretical sampling conditioned by local conditional probabilities, close to all subsequent methods account for Ashikhmin's considerations. In his *Image Analogies*, Hertzmann considers both the nearest neighbors and the coherent candidates during synthesis for combining the strengths of both approaches [HJO+01].



2.28 Influence of the Size of the Matched Neighborhood for Pixel-Based Non-Parametric Approaches.

For their Bi-directional Texture Function synthesis technique, Tong *et al.* improve Ashikhmin's technique by incorporating to the set the coherent candidates' k-nearest neighbors (*cf.* Figure 2.27*(right)*). They also fully exploit the reduction by coherence of their search space to compensate for the impossibility to resort to previous acceleration schemes because of their texton-matching distance [TZL+02]. Finally, Zelinka and Garland embed all coherence-related information in a separate structure created in a pre-computation step [ZG03, ZG04]. Their *jump map* corresponds to a 2d array of the same size as the input, whose entries store each input pixel's possible "jumps", or input locations of its most similar neighborhoods. They perform Poisson sampling for preventing spatial aggregates, and the probabilities of the retained jumps are weighted in accordance to their visual similarity. Linear-time synthesis then only consists in traversing the output pixel grid according to a space-filling curve, looking at the previously-generated pixel's jump list and randomly picking one of them.

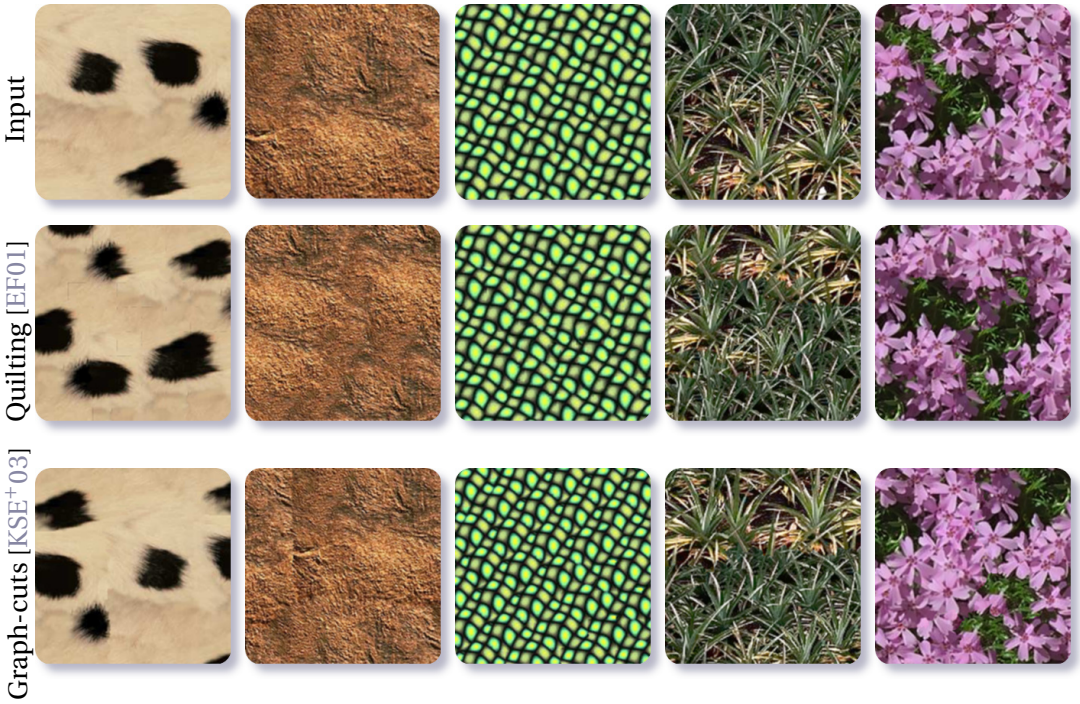
2.2.6.5 Non-Parametric Patch-Based Synthesis

The manipulation of pixel patches is a natural evolution of non-parametric sampling texture synthesis techniques, and is motivated by the need to ever increase the size of the matched neighborhoods, or enforce image coherence to preserve the input's structures (*cf.* Figure 2.28). It also constitutes a straightforward way for factorizing redundant computations due to over-constrained searches at nearby locations. But switching synthesis operands does not come effortlessly: along with the necessary adaptation of the sampling procedure, it begs the question of hiding seams between adjacent patches.

Patches for Texturing Complex 3d Shapes

First work on patch-based texture synthesis revolve more around the question of texture mapping on 3d surfaces than actual texture generation. Since globally-consistent parameterizations[3] are usually not available for

[3] A parametrization designates a continuous, bijective function mapping open sets from the 3d surface onto open sets of the 2d texture space. It is often computed by optimization so that it possesses satisfactory properties, *e.g.* continuity, regularity and minimal distortions.



2.29 Patch-Based Non-Parametric Sampling [EF01, KSE+03]

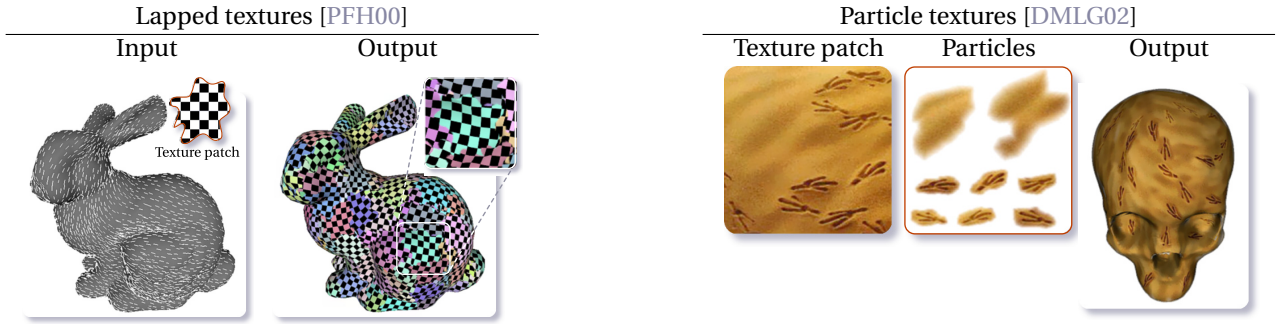
complex 3d shapes, mapping a texture onto their surface requires the resort to local parameterizations instead, and thus the division of the input texture into suitable tiles.

Praun *et al.*'s *Lapped textures* are among the first patch-based texturing technique over a mesh surface [PFH00]. After manually extracting a representative portion of the input texture, the user is to specify a sparse vector field which once interpolated using radial basis functions, constraints the *local* parameterizations of the surface. Patches are sequentially pasted onto the mesh surface until full coverage, by greedily adding faces around a seed point until the patch's topological equivalence to a disk is lost or distortions become too visible. The lapped textures' main limitation comes from the lack of explicit handling of the transition between patches which makes seams especially noticeable for highly-structured textures and inputs with strong low-frequency components (*cf.* Figure 2.30*(left)*). Praun's synthesis framework has been recently extended by Takayama *et al.* to the case of volumetric textures [TOII08], by trading pixel patches for 3d texture volumes and monitoring the distortions caused by their 3d parametrization over the tetrahedralization of the geometry.

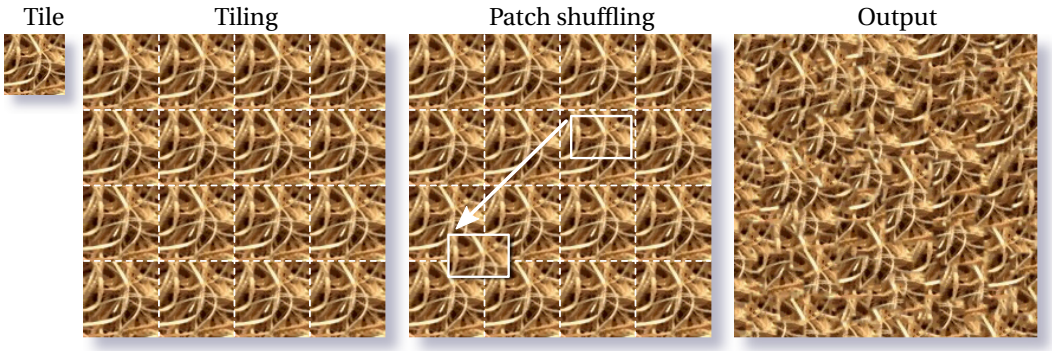
Dischler *et al.*'s *Texture particles* approach is more related to our gait as it extracts and analyzes the spatial arrangement of a set of texture primitives [DMLG02]. But while we strive for the automatic localization of the input's constitutive elements, Dischler has the user manually designate the particles via color quantization and a lasso-like tool. Each particle has up to four neighbors determined by its contact with their successively-dilated counterparts, and their placement is analyzed by the consideration of the relative distances between their bounding boxes. Synthesis is then performed in a greedy fashion, by adding one particle at a time. In their implementation, Dischler *et al.* only account for the case of a single particle class.

Accelerating Texture Synthesis

2d-bound patch-based methods are more concerned about the acceleration of the synthesis process, as well as the improvement of their outputs' visual quality. Xu *et al.*'s *Chaos mosaic* strives for the handling of long-range structures and is specifically designed so that the input's local features are left unaltered while their spatial distribution in the output globally appears as "visually stochastic" [XGS00]. They randomly cut square blocks from the training image, and dispose them on a new output tiling in a seemingly random manner. Instead of shuffling pixel colors directly, they inject randomness at the block level and hence better guarantee the conservation of small shape boundaries. The blocks' positions are determined by Arnold's cat map which



2.30 Patch-Based 3d Surface Texturing. Due to the impossibility of finding a global parametrization, first surface texturing techniques combine local parameterizations instead. And even though they do not aim at explicitly create new textures *per se*, they were the first to handle patches. But neither Praun's lapped textures *(left)*, nor Dischler's texture particles *(right)* tackles the issue of patch seams.



2.31 Xu *et al.*'s Chaos Mosaic [XGS00]. Pioneering the handling of patches in place of isolated pixels for texture generation, Xu *et al.*'s method aches for speed and thus leaves aside elaborated processes for concealing seams. Once the user has selected a representative tile from the sample, the technique produces its outputs by transforming a simple tiling into a more "visually stochastic" result by shuffling random patches taken according to a deterministic chaos mapping.

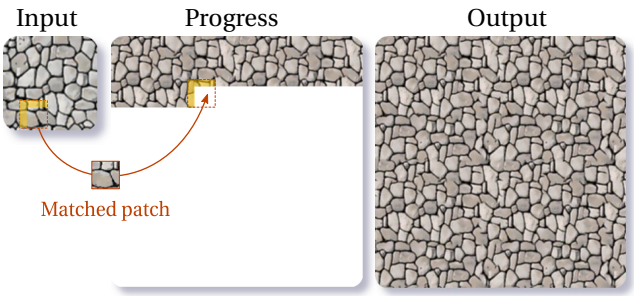
exhibits a convincing visual randomness only after a couple of iterations[4] Xu *et al.*'s use of the cat map is a clever choice as its actual deterministic nature gifts their *virtual texture* representation the efficiency of a procedural model. Focusing on the achievement of real-time texture synthesis, their time performances come at the expense of the end results' visual quality as they favor simple cross-edge filtering instead of constrained pixel-based synthesis for hiding the patches' borders.

Building upon Efros' and Wei's non-parametric representation of the pixels' conditional probability distributions, Xu and Liang subsequently propose the first texture synthesis technique consisting in the patch-based sampling of the input. Their patch-involving matching relies on the squared color distances integrated over the patches' boundary regions. Blocks from the input whose boundary agrees with the constraints imposed by the output patches are added to the result as summed up in Figure 2.32. For rapidity purposes, the treatment of overlapping colors at the conditioning boundary regions is again limited to simple linear blending. Further acceleration is achieved by dimensionality reduction of the search space and the recourse to a quad-tree pyramid. While better handling wider texture features, they mostly substitute the pixel-based approaches' direct sensitivity to the size of the matched neighborhoods for the sensitivity to the width of the conditioning boundary regions. Subsequent work also mention the presence of smeary artifacts that are all the more noticeable as they occur along the straight patch borders.

Exploiting our sensitivity to edges and corners in images, Wu and Yu later alleviate the presence of broken features at patch boundaries by improving the matching error evaluation so that its favors the preservation across adjacent patches of thin curvilinear features such as ridges [WY04]. Following the rationale that defor-

[4] Stemming from the theory of deterministic chaos, Arnold's cat map Γ is a transformation of the torus $\mathcal{T}$, assumed of size $m \times m$, onto itself defined as $\Gamma(x,y) = \big((x+y) \mod m, (x+2y) \mod m\big)$.

2.32 Scan Line Traversal for Patch-Based Sampling [LLX+01, EF01]. Patch-based synthesis follows the steps of successful pixel-centric approaches [WL00, Ash01, HJO+01] and thus first relies on a scan line traversal scheme. It similarly uses a L-shaped causal "neighborhood" at the common boundaries with already-chosen patches *(yellow region)*, the integral of the squared color differences emulating the conditional probability of a new patch.



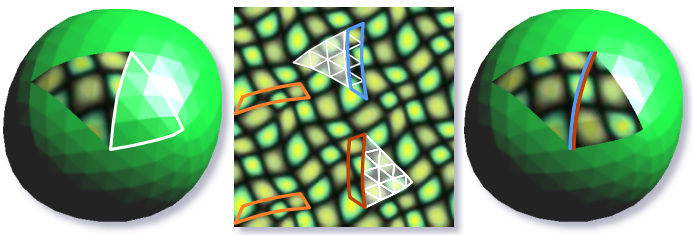
mations are less visible than discontinuities, they also slightly warp the boundaries of any newly-added patch to better ensure the continuation of such features.

Bringing Soler *et al.*'s idea of patch subdivision of error-prone patches back to the 2d case [SCA02], Nealen and Alexa suggest an hybrid approach that takes advantage of the local feature preservation ensured by the patches, coupled with seamless transitions achieved by constrained pixel-based synthesis [NA03]. This seam correction alternative was previously mentioned by Xu three years before, but the idea was dropped due to their performance constraints.

Beyond Square Patches

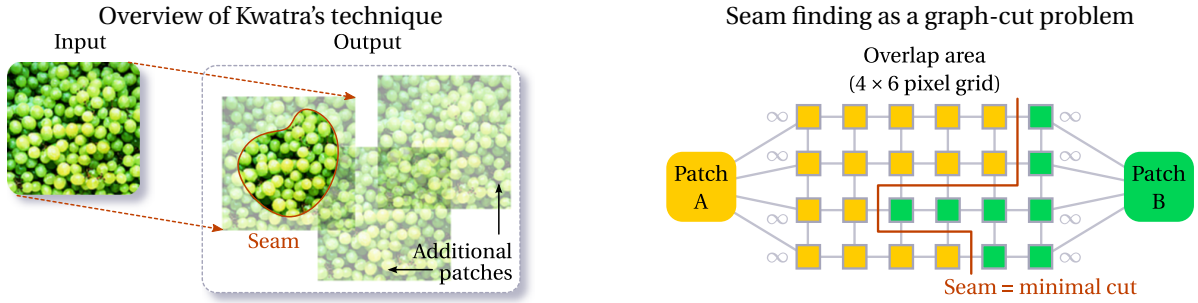
Addressing the shortcomings of their previous technique, Efros and Freeman adapt their non-parametric sampling framework to the case of patches and propose *Image quilting* [EF01] which lifts one of the major weakness of Liang's patch-based technique: the fixed size of its blocks. Once the patch of minimal overlap matching error has been found, its optimal boundary with the output is determined as the minimal-cost path through the overlap region. By making path cost coincide with color differences, they substantially limit color discontinuities while avoiding the blur effect commonly evoked by blending, and prevent their occurrences along straight lines (*cf.* Figure 2.35).

In their hierarchical texturing approach, Soler *et al.* transpose most of the concerns relative to patch-based texture generation to the realm of 3d surface texturing [SCA02]. Their method shows a respectable number of improvements on its own. The surface's coverage by patches is elegantly controlled by its hierarchical partitioning into face clusters that can be recursively subdivided while their associated error exceeds some threshold. Their error measure incorporates the distortions caused by flattening of the clusters, as well as the summation of the color differences at their boundaries (*cf.* Figure 2.33). Contrary to earlier 3d surface texturing approaches, they automate patch extraction and pioneer the manipulation of texture coordinates in place of colors. Lastly, they achieve an important speed-up by formulating their color matching error as the sum of correlation functions whose evaluation can be carried out as simple products in the Fourier domain.

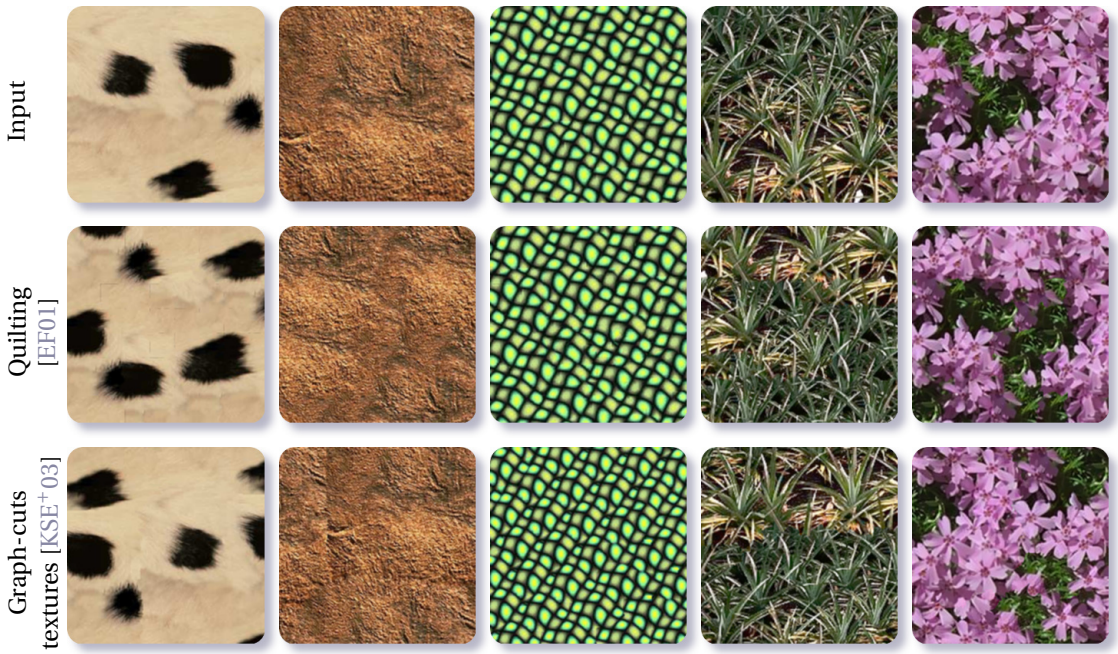


2.33. Soler *et al.*'s Patch Sampling [SCA02]. Soler *et al.*'s technique is the first 3d surface texturing approach to conceal seams by choosing input patches according to their fitting error. *left:* A patch is to be stitched over the white triangular area. *center:* The colors of already-placed patches *(blue strip)* restrict the set of patch candidates *(in orange)*. Only the patch of minimal error *(red)* is retained for pasting onto the surface *right:* This operation is performed by synthesizing appropriate texture coordinates at the newly-textured vertices.

Inspired by Efros's resort to dynamic programming, Kwatra casts the patch boundary finding problem into a graph-cut problem over the graphs of the overlapping regions' pixels [KSE+03]. Ensuring edge-aware segmentation, graph-cuts reshape any-newly added patch in accordance with the already-generated output background. Kwatra overcomes the limitation of graph-cuts to the binary segmentation case by accounting for all the costs cuts through old patches would generate when adding a new patch. His now non-causal synthesis



2.34 Kwatra's Graph-Cut Textures [KSE+03]. Kwatra's method works by pasting input patches onto the output and finding seams that minimize color discontinuities with already-incorporated patches *(left)*. Its notable improvement is the reformulation of seam finding as a graph-cut optimization problem [BVZ01] *(right)*.



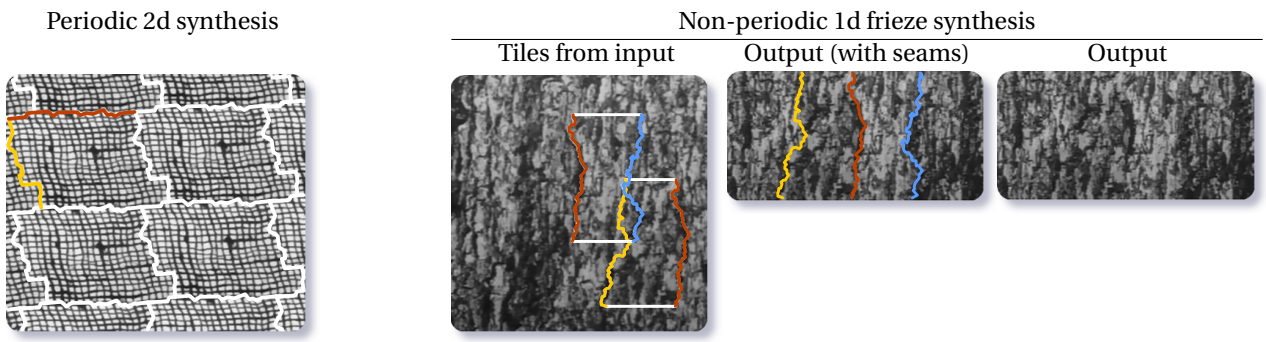
2.35 Textures by Patch-Based Non-Parametric Sampling [EF01, KSE+03]

therefore grants complete freedom to the positioning and ordering of patches that are not to be processed in scan line order anymore. Even today, Kwatra's achievements are among the most convincing patch-based synthesis results to date, both in terms of visual quality and performance as illustrated in Figure 2.35.

2.2.7 From Pixel Patches to Texture Tessellation

From the research presented so far, we see that the analysis required by example-based texture generation started at the scale of the input's smallest components, and progressively considered wider elements to have the synthesis wary of the characteristics of the input's content, from joint distributions of pixel values, to pixel neighborhoods, to free-form patches. The notion of pixel coherence or the edge-aware cropping of patches are successful attempts at considering low-level cues (color co-occurrences and discontinuities) for counter-balancing the lack of semantic knowledge on the input. Another trend of texture synthesis techniques instead take a top-down approach and rather than aiming at synthesizing the broadest range of textures, only targets regular and near-regular textures[5]. For these specific images, pixels are clearly not a satisfactory representation while their replicated tile constitutes an intuitive and efficient synthesis handle. Finding these tiles is at

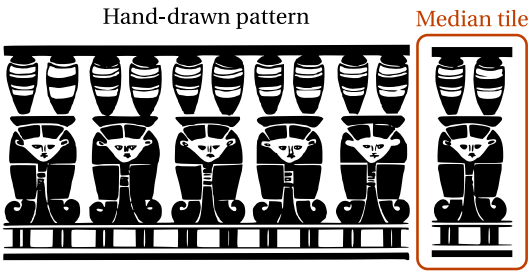
[5] A *regular texture* refers to a congruent, periodic 2d tiling of the plane, and depending on whether its generative transform corresponds to a 1d or 2d translation, corresponds to a frieze or wallpaper pattern. Kaplan and Salesin's *Escherization* counts among the first tiling-based texture generation methods but does not fall into the category of example-based synthesis as the generative tile is provided by the user [KS00].



2.36 Cut-Primed Smart Copying [NZvG03]. Instead of focusing on seamless patch pasting, Neubeck *et al.* find pairs of translated cuts that minimize the resulting stitching discontinuities. Only once cuts are established that patches are defined. Looking for a valid texture tiling, their method works best for periodic textures *(left)*. Hiding repetitions involves the handling of several tiles sharing a similar boundary *(red path on the right)*.

the core of near-regular texture centric techniques. More than mere patches, tiles distinguish themselves as they are to generate a complete *tessellation* of the input texture.

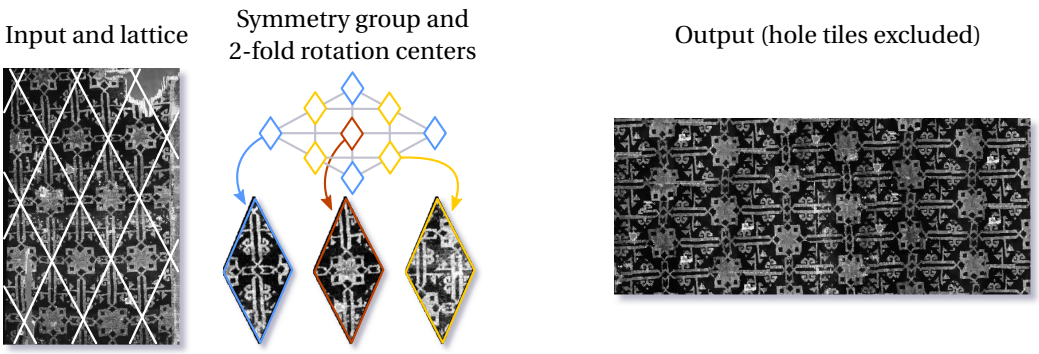
Early attempts at example-based texture tessellation follow the steps of successful patch-sampling techniques [EF01, KSE+03] and do not explicitly target regular samples directly. Neubeck *et al.*'s *cut-primed smart copying* stems from the will to reverse the work flow of previous approaches: instead of tweaking boundaries after a patch has been selected, they first find appropriate cuts through the input texture [NZvG03]. They look for pairs of cuts (the second being a translated version of the first) that minimize their seam visibility cost by shortest-path finding over the graph of input pixels. Two passes are required to find vertical, then horizontal cut pairs. They also provide a multi-scale extension alleviating cut estimation costs. Particularly efficient on near-periodic textures, their technique can handle 1d non-periodic stripes but wider inputs are needed (*cf.* Figure 2.36).



2.37 Automatic 1d Frieze Pattern Analysis [LCT04] .

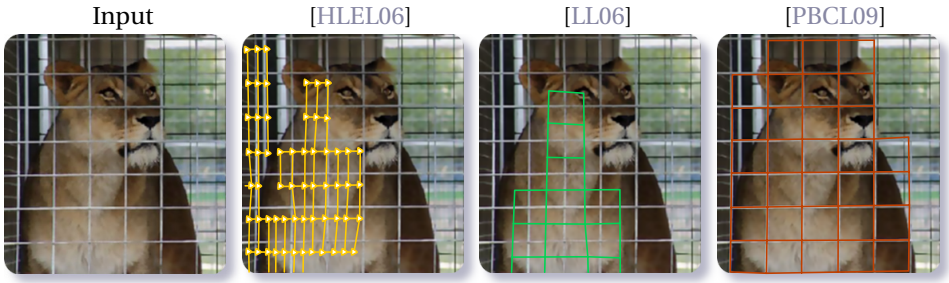
In outstanding research dedicated to near-regular textures, Lui *et al.* make the decision of injecting prior knowledge into their synthesis process and place the geometric nature of their specific inputs at the core of their analysis. They aim at the explicit extraction of the minimal tile whose repetition reconstructs the input, and tackle this issue from the dual perspective of finding the training image's underlying lattice. They first propose a computational model for frieze and wallpaper textures along with a framework for the analysis of the symmetry group structures found in images [LCT04]. Lattice extraction comes down to identify the two shortest, linearly-independent translation vectors along which the tile is duplicated. Such vectors are unveiled by the study of the regular layout of the peaks in the input's auto-correlation surface. Instead of simple global thresholding for localizing these peaks, they suggest a greedy peak selection strategy driven by the size of their region of dominance (the circle centered on the peak devoid of other peaks), and then process the detected peaks via Hough transform to extract the lattice's generative vectors. But the shine of their method is their automatic frieze/wallpaper group classification based on the theory of crystallographic groups. This classification is tractable thanks to the finite nature of the problem: in the 2d plane, only seven frieze groups and seventeen wallpaper groups exist. This enables the modeling of their classification algorithm as a finite decision automaton similar in spirit to Sanderson's algorithm. They extract the median tile from the previously-obtained lattice (*cf.* Figure 2.37), and study its invariance to a finite set of symmetries to determine the input's symmetry group and guide re-synthesis accordingly.

While robust especially to the presence of missing tiles (*cf.* Figure 2.38), Liu *et al.*'s approach assumes textures are observed from a frontal view, user-guidance becoming necessary if their lattice is transformed by an external field [LLH04]. Hays improves robustness to deformations and formulates lattice extraction



2.38 Liu *et al.*'s Automatic Front-Facing Lattice Extraction and Symmetry Group Classification [LCT04].

as an iterative pairwise correspondence problem [HLEL06]. Points-of-interest are detected, and assignment vectors are associated with scores accounting for the visual similarity between the matched points and the overall spatial agreement between assignments. The optimal global point assignment is then obtained by the spectral analysis of the resulting affinity matrix. The lattice is then updated and refined after the removal of outlier points, and the deformation field is interpolated accordingly. Additional points sampled in accordance with the lattice's current state are added and the process repeated. Park *et al.* consider lattice extraction from a tracking perspective [PLC08]. They greatly improve Hays' expansion step by encoding the lattice via a hidden Markov Random Field whose nodes correspond to the tiles' positions and edges account for their spatial consistency and appearance agreement with a common tile template. The final positions are then propagated by optimization over this field.

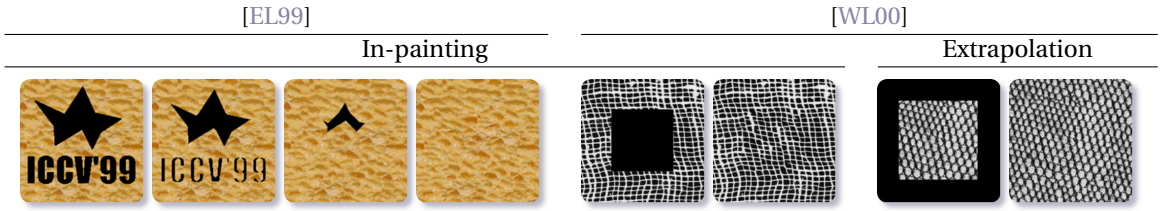


2.39. Improvements in Automatic Lattice Detection. Lattice detection from real-world images yields promising results. First to tackle the issue of robustness w.r.t. deformations, Hays formalizes lattice detection as a correspondence problem subject to higher-order concerns. He proposes an iterative technique that computes the lattice's two translation vectors and estimates its extent [HLEL06]. Building upon Lin's tracking system for handling videos [LL06], Park *et al.* re-cast this problem into a tracking problem and achieve the best results to date thanks to their mean-shift belief propagation [PBCL09].

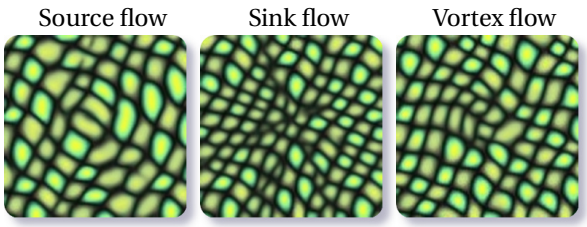
As Lui *et al.* highlight in their open discussion revolving around near-regular textures [LTL05], texturing methods should not impose any prior on the shape, orientation or size of the texture elements the input is composed of, and we acknowledge this opinion. Via their assumption according to which these elements lie on top of a lattice, they successfully separate their disposition from their appearance, and the variations thereof. However, such approaches, despite an ever-improving robustness to the presence of deformations, cannot generalize to textures whose elements do not follow one of the spatial configurations dictated by the symmetry groups. We similarly aim at automatically extracting representative elements from the input with at least prior as possible while lifting their regularity assumption.

2.2.8 Automation *versus* Artistic Control

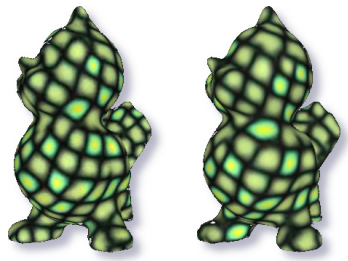
Finding a compromise between automation and control, between efficiency and creativity is often a critical point decisive for the technique's usefulness. This is especially true when it comes to a task such as texture synthesis that highlights the extremes of these two aspects. At first, it may seem like a fairly bland, space-filling chore whose only restrictions is to avoid obvious repetitions. But texture generation can also be approached from a radically different, artistic perspective and while our method is more directed at automation, we believe the mention of this aspect of existing techniques is directly relevant for the topic of this thesis.



2.40 Pixel-based Constrained Synthesis [EL99, WL00]. Non-parametric sampling schemes for texture generation naturally provide a framework for constrained synthesis. The fixed output pixel values condition the neighborhoods used for matching at their vicinity and hence limit visible seams.



2.41 Flow-Guided Texture Synthesis [KEBK05].



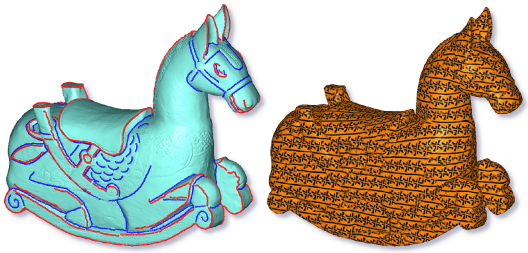
2.42 Frames of an Animated Surface Texture [HZW+06].

From early non-parametric sampling methods, most of the aforementioned techniques provide the user with some control that goes beyond simple parameter tweaking. It is naturally conditioned to the nature of the building blocks manipulated during synthesis: the smaller the synthesis units, the finer the user's control. Such controls fall into four main categories: the imposing of constraints on the output, the guidance of the synthesis by external data attached to the input, the altering of the synthesis process, and the consistent edition.

2.2.8.1 Constrained Synthesis

Constraining some of the output values is the most straightforward way to control the synthesis. Its main task then becomes to find the remaining values so that these constraints appear unnoticed. Earliest pixel-based methods by Efros already propose similar applications for texture synthesis, in-painting being the most representative use case where after removing foreground objects from an image, synthesis is used to fill holes by propagating the background texture from their boundaries [EL99](*cf.* Figure 2.40).

Kwatra *et al.*'s optimization framework appears as even more amenable to the application of local constraints over the output, and enables the distinction between soft and hard constraints. Created textures can follow a flow field [KEBK05] (*cf.* Figure 2.41), or be animated across successive frames [HZW+06] (*cf.* Figure 2.42).



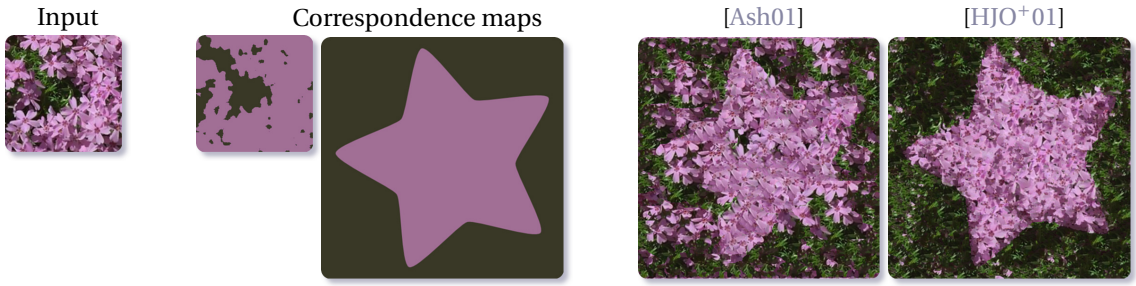
Xu *et al.* propose a feature-aware surface texturing technique by constraining the output according to a vector field extrapolated from the surface's salient curves [XCOJ+09]. Although the alignment of to the surface's features conflicts with the preservation of the texture's structures, their results show great visual improvement with a enhanced perception of shape. Artists can also sketch their own curves.

In the case of 3d textures, hard constraints can be used to impose fixed patterns over slices of the volume which go unnoticed from any other slice [KFCO+07] (*cf.* Figure 2.43).

2.2.8.2 User-Guided Synthesis

User guidance is a more gentle version of constraining. In place of fixing output values, the synthesis is globally guided rather than locally constrained. Ashikhmin proposed the earliest examples of *texture painting* where users roughly paint the color distribution the output texture is to respect [Ash01]. During synthesis, matching

2.43 Constraints over a Slice of a Texture Volume [KFCO+07]. Extending Kwatra's optimization technique to 3d textures, Kopf can easily constraint the colors of an entire 2d slice of his outputs.

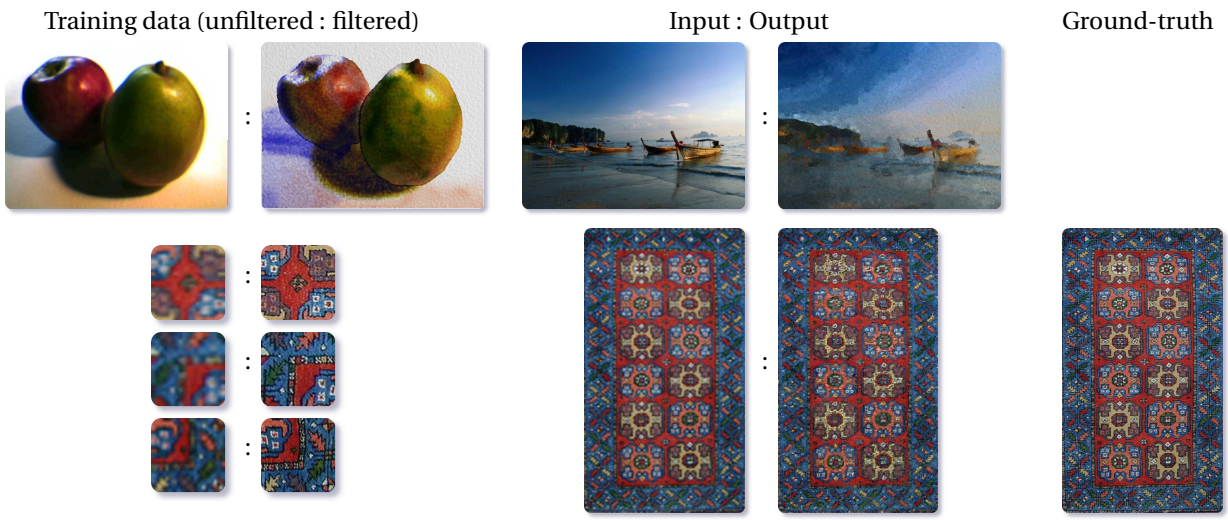


2.44 Pixel-Based Guided Synthesis. By embedding external values that lay additional constrains on the conditioning neighborhoods, non-parametric synthesis methods can perform guided texture generation [Ash01, HJO+01]. First formalizations use *correspondence maps* defined over both input sample's and output's domains (*e.g.* smoothed luminance channel, or user roughs for *texture-by-number*). The juxtaposition of Ashikhmin's and Hertzmann's results underlines the visual impact of pixel coherence which achieves a better preservation of local shape cues at the expense of color faithfulness.

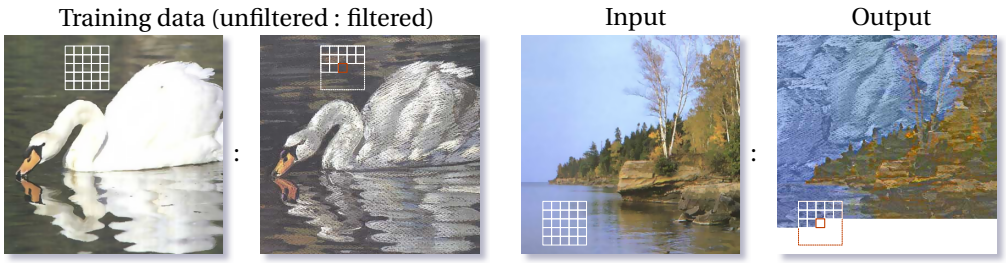
then involves neighborhoods made of already-generated and the user-indicated color values (*cf.* Figure 2.44). Efros *et al.*'s *Image quilting* improves Ashikhmin's concept of guided synthesis and enables *texture transfer* consisting in the rendering of an object with a texture captured from another [EF01] (*cf.* Figure 2.47). They cast this problem as a hidden synthesis process between the two objects' respective correspondence maps. Correspondence maps refer to spatial maps of a common quantity defined over both input and output that are used to estimate the matching error. Since Efros' method directly works with patches, fine local features are well preserved, and transfer of expressive rendering filters yields interesting results. Hertzmann *et al.*'s *Image analogies* formalize the question of by-example image filtering as an augmented texture synthesis problem (*cf.* Figure 2.46), and suggest a very elegant and general framework that is transposed one year later to the case of vectorial curves [HOCS02]. Along with gorgeous results, Hertzmann proposes an impressive variety of applications for his non-parametric analogies: texture transfer, inference of higher-resolution images and texture-by-numbers (*cf.* Figures 2.45 and 2.47).

2.47. Pixel-Based Versus Patch-Based Texture Transfer. Both Efros' and Hertzmann's methods allow texture transfer, *i.e.* the texturing an object with the material properties of another. Transfer is a specific case of non-parametric guided synthesis where additional information is used to bias the matching process. Directly comparing their results highlights the advantage of patch-sampling for local features preservation.

But guidance is by no means confined to sequential synthesis techniques. Dong *et al.* recently proposed a technique for the guided creation of perspective-aware textures by optimization [DZP08]. They compute a map storing the input pixels' scale distortion caused by the perspective projection. It is either inferred from the local density of feature points, or computed by the slant and tilt angles specified by the user, and then acts as an additional channel for guiding the optimization.

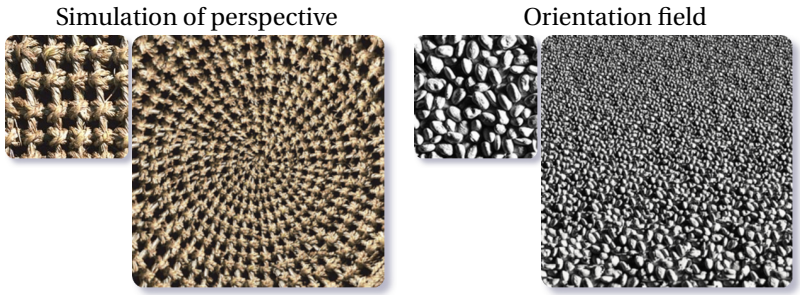


2.45 Hertzmann *et al.*'s Image Analogies [HJO+01]. More of a data-driven tool for pictorial style transfer than a new texture synthesis technique, Hertzmann's *Image analogies* take as inputs pairs of training images *(left)* showing the effects of the "filter" to emulate and then apply it to new inputs *(right)*. In spite of the simplicity of its "capture" concept, this method produces impressive results for an wide variety of filters ranging from artistic ones *(top)* to super-resolution inferrers *(down)*.



2.46 Analogy by Non-Parametric Sampling [HJO+01]. Instead of explicitly manipulating parametric filters, Hertzmann builds his outputs by sequentially copying pixels from compatible neighborhoods. These are similar to the ones used by Wei [WL01], at the only exception they span over different images and resolution levels.

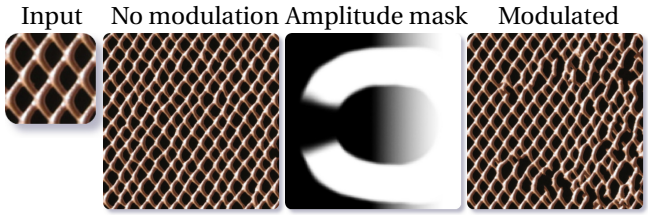
2.48 Tuning the Output Traversal Order [ZG04]. These results illustrate the controllability of pixel-based texture generation techniques. Efficient effects, such as fake perspective foreshortening (*left*) or guided orientation of features (*right*) are straightforwardly obtained by locally altering the output pixel traversal.



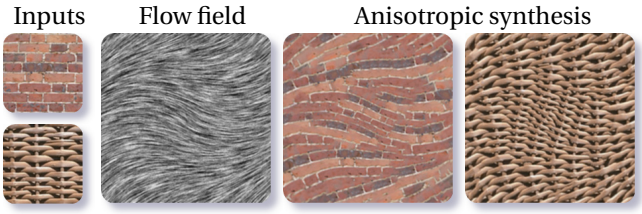
2.2.8.3 Altered Synthesis

A less intuitive, but maybe more powerful editing handle is to alter the synthesis process itself. As it requires the intricate control and deep understanding over the synthesis stage, it is mainly confined to pixel-based greedy techniques. A good example is Zelinka's technique which proposes a 1d traversal of all the output pixels using a Hilbert space-filling curve. By locally changing the step distance and the local frame's "up" direction, anisotropic textures and perspective effects can effortlessly be produced [ZG03, ZG04].

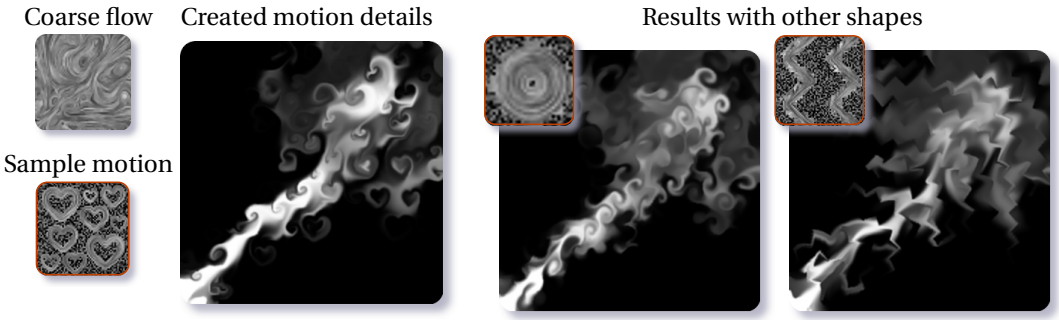
Lefebvre *et al.*'s parallel texturing technique also allows this kind of control. Thanks to their knowledge of the visual effects caused by their method's different stages, they provide users with efficient handles, notably over the jittering step which controls the stochastic appearance of the outputs. By allowing its modulation by a mask, artists can craft spatially-variant textures [LH05] (*cf.* Figure 2.49). Warping the matched neighborhoods



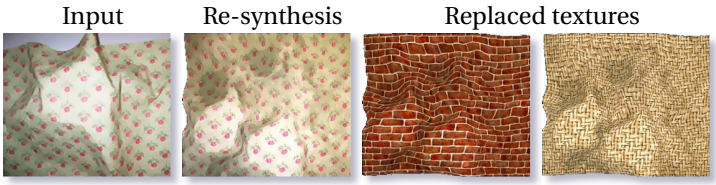
2.49 Spatial Modulation of the Jitter Amplitude in Lefebvre *et al.*'s Parallel Synthesis Technique [LH05].



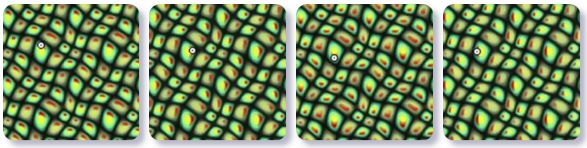
2.50 Flow-Guided Orientation of Local Frames in Lefebvre *et al.*'s Synthesis Technique [LH06].



2.51 Example-Driven Motion Field Synthesis [MWGZ09]. Ma *et al.*extend Kwatra's optimization approach to infuse user-designed detail motions into a coarse, low-resolution motion field and infer a finer version of it. The major differences for handling motions in place of colors stem from the need to re-orient neighborhood frames according to the coarse flow.



2.52 Liu *et al.*'s Near-Regular Texture Replacement [LLH04].



2.53 Brooks *et al.*'s Pixel-Based Texture Edition [BD02].

during the correction step enables anisotropic texture synthesis [LH06] (*cf.* Figure 2.50).

Altering the synthesis process unfolds in an optimization-based framework have also been proposed by Ma *et al.* who "synthesize" artistic motions by adorning coarse motions with manually-designed details (*cf.* Figure 2.51). Motion-specific concerns, *e.g.* fluid incompressibility or boundary conditions, are not directly embedded into the synthesis process though and Ma instead enforces these in a post-process stage.

2.2.8.4 Consistent Texture Editing

The last editing metaphor possible is the consistent editing of the texture elements. Since one of the characteristics of textures is their stationary, they usually exhibit much repetition which can be exploited to propagate the changes made on one element over its replicates. This control is eminently more complex than the previously-mentioned ones as it depends on the notion of elements' *instances*. It is thus mostly suited to the tiling-based approaches. Liu *et al.* notably handle the manipulation of near-regular textures once their lattice structures have been detected and corrected if need be [LCT04, LLH04] (*cf.* Figure 2.52).

Edition propagation is more difficulty handled in pixel-based techniques because of the lack of structural information, the only cue available being the matching error that acts as a dissimilarity measure between local neighborhoods. they can rely on instead is the implied similarity between neighborhoods exhibiting a low matching error as exploited in Brooks and Dodgson's work [BD02] (*cf.* Figure 2.53). But since all operations can only be carried out at the scale of pixels, the range of actions is limited and special care must be provided to the matching error estimation for insuring rotation invariance for instance. With our shape-driven approach, we mainly target this last category of editing approach.

CHAPTER 3

Pattern Recognition for the Automatic Synthesis of Raster Shape Arrangements

3.1 Introduction

3.1.1 Intuition Behind Our Texture Analysis Approach

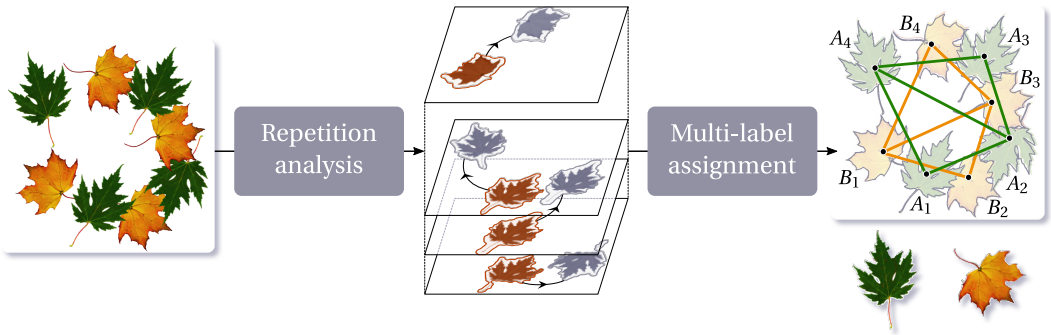
Chapter 2 shows the impressive variety of the research dedicated to raster texture creation, and thus sheds light on the remaining challenges and unexplored tracks we can take to rise to them. Existing example-based texture synthesis techniques are inherently unadapted to textures consisting of a set of randomly disposed, individually discernible shapes. Local methods striving at pixel-based discontinuity reduction hardly preserve input's long-range structures. Alternatively, research built upon the supposed respect by the input's features of given placement rules are too restrictive to be straightly extended to stochastic arrangements. In the following, we present a new method for analyzing and re-synthesizing such arrangements, our ultimate objective being to acquire their constitutive shapes directly in order to enable structure-aware re-generation and use the appropriate building blocks for that aim.

What characterizes such shapes is their repetition at different locations throughout the input. We exploit this trait by recording recurrences of visually similar neighborhoods which are later extended to regions. We bring those together to compute the input's coverage map and extract final repetitive shapes. By directly manipulating shapes, re-synthesis can be enriched with high-level information unavailable in pixel-based approaches. We gather statistics on their placement and appearance variations and use those to produce new images. To achieve this, we draw inspiration and improve techniques for capturing element arrangements, techniques once limited to vector NPR primitives.

The remaining part of this chapter is mostly dedicated to our novel analysis scheme for raster textures which focuses on the automatic detection of relevant shapes through their very repetition, and the handling of information extractable from the provided image sample without imposing any priors. Along with this analysis stage, the directly subsequent synthesis step is presented in a more mitigated way as its intrinsic contribution with respect to related work in vector texture synthesis is not as noteworthy. Our first attempt at re-synthesis draws much of its inspiration from existing work aiming at the automatic creation of vector arrangements by example, more specifically Barla *et al.*'s technique [BBT+06] and Ijiri *et al.*'s procedural extension of it [IMIM08]. Both approaches mostly consist in a witty transposition of notions commonly encountered in non-parametric pixel-based texture synthesis [EL99, WL01], such as the computation of matchable local neighborhoods and the definition of valid metrics on these. The re-synthesis results we propose at the end of the chapter are produced using a similar approach based on the greedy growing of an underlying triangulation. The most contributive aspect of our re-synthesis step mainly stems from the actual nature of the data it manipulates: in place of vectorial shapes, it directly disposes automatically extracted shapes composed of pixels. One could hence consider our shape recovery step as some alternative kind of image vectorization. Follows the rough outline of our analysis approach, also illustrated in Figure 3.1.

3.1.2 Overview of Our Shape Extraction Technique

Given an input image, we express its content in terms of replicated copies of shapes by relying on the multiple occurrences of similar regions only (*cf.* figure 3.1). Recall our method is targeted to the analysis of textures consisting of the arrangements of repetitive planar shapes which are often at odds with most state-of-the-art



(a) Input (b) Region similarity map (c) Patterns and coverage map

3.1 Overview of Our Raster Arrangement Analysis Stage. Given an input raster image *(a)*, we build its region similarity map *(b)* containing all pairs of similar regions across the image, along with the non-linear transformations that tie them together. *(c)* Lastly, we compute the repetitive shapes' instances along with the input's visibility map.

pixel-based techniques. Our ultimate objective is to locate, extract and potentially reconstruct incomplete instances of these duplicated shapes and exploiting their repetition is the lowest-level albeit most meaningful information we can rely on. To get the needed information for the generation of original content from the texture sample at our disposal, we proceed as follows.

First, statistics are performed over the input image in order to detect all pairs of maximal regions which are similar up to a non-rigid transformation. Although regions of a given pair are disjoint, regions of different pairs can overlap or even be completely included in one another. We call this information the *region similarity map* (*cf.* figure 3.1*b*). It contains all the similarity information in the image at the level of entire regions. This step involves the study of the matches between similar pixels, in a way to retrieve connected regions from the input and is detailed in Section 3.2.

In a second step, the region similarity map is exploited in order to turn independent pairs of similar regions into meaningful classes of repetitive objects in the image (*cf.* figure 3.1*c*). For each class, an *original object* is constructed by collecting information from its various instances. For that, all input's pixels must be tagged with the unique object/instance pair it belongs to. We will show in Section 3.3 that this step is an optimization problem we find an approximate solution of using an appropriate heuristic.

Finally, having obtained a discrete representation of the image content, we perform statistics over the placement, orientation and attributes of the recovered patterns. We then use those measurements and adapt existing techniques for copying element arrangements to produce new images, as explained in Section 3.4.

3.2 Constructing the Region Similarity Map

The main insight is to use image regions occurring multiple times throughout the input as handles to recover the structure of the image.

3.2.1 Visual Similarity Detection

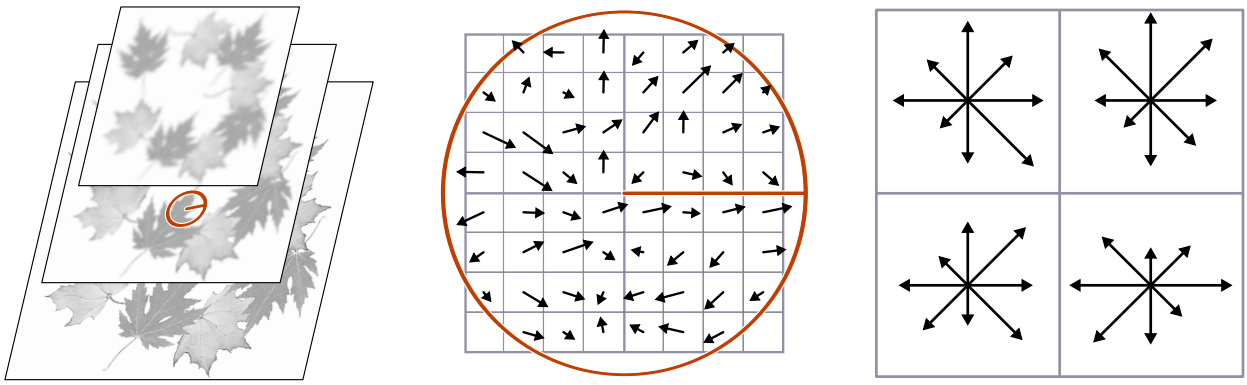
Most successful Computer Vision techniques for image classification and object recognition involve local descriptors. Many variants exist, eventually encoding different visual aspects of images (e.g., luminance, gradients, colors), and they have been widely exploited since they offer improved robustness to noise and partial occlusion. The quality of a local descriptor is evaluated according to its repetitiveness and discriminability. From the literature in that field, studies conclude that Lowe's SIFT descriptor [Low99] yields the best results [MS05]. This descriptor is computed in gradient domain, accounts for both textural and shape local information, and is invariant by 2d similarities.

Optimal Scale Detection To achieve invariance by uniform scaling, we evaluate SIFT descriptors at a specific scale, on an adequately Gaussian-blurred version of the input. We compute all pixels' scale of interest using

Lindeberg's scale selection method in accordance to linear scale space theory [Lin98]: the pixels' retained scale coincides with the scale at which the normalized Laplacian response reaches a local maximum along the scale dimension.

Canonical Orientation Once again, in order to constitute a rotation-invariant description, SIFT descriptors are computed with respect to a local direction. We determine every pixel's associated canonical directions by computing a 36-bin gradient orientation histogram. Neighboring gradients within a 19×19 window are evaluated at the level of the scale space adapted to the pixel's scale. These gradients contribute to the bin associated with their direction proportionally to their magnitude. The pixels' canonical directions then correspond to the bins with locally maximal amplitude.

Local Neighborhood Description Though many variants appeared since its definition, we implemented Lowe's original, 128-dimensional SIFT on the input's lightness channel (*cf.* Figure 3.2). Taking advantage of more elaborate, chromatic descriptors does not change the proposed method and is currently left to future work.



(a) Gaussian scale-space (b) Local gradients (c) 2×2 SIFT descriptor

3.2 **Lowe's SIFT Descriptor.** *(a)* Once we get a scale and direction for a pixel, *(b)* we express its neighboring gradients in its rotated local frame. *(c)* Finally, the SIFT descriptor is the concatenation of several orientation histograms disposed around the pixel. In our implementation, we use 16 8-bin histograms of gradients sampled on a 19×19 window at the selected scale.

Descriptor Matching Our motivation for using SIFT descriptors is their ability to encode visual appearance in a concise, yet meaningful way. Thanks to them, evaluating visual similarity between two neighborhoods up to any rigid transformation becomes straightforward. The smaller the Euclidean distance between their respective descriptors is, the more visually close the pixels' neighborhoods are. This observation allows us to easily find local repetitions by finding for each pixel its nearest neighbors in SIFT feature space. We embed all the computed SIFT descriptors within a Kd-tree and perform for every pixel a fixed-radius search. In our examples, the search window radius ranges between 0.1 and 0.15 in normalized SIFT space.
The highly-dimensional feature space is beforehand reduced by Principal Component Analysis (PCA). It is performed on the whole distribution of computed descriptors and greatly alleviates the pairing computation costs. Neighboring pixels tend to be associated with similar descriptors and blind matching often involves small clusters of redundant pixels. We limit this phenomenon by locally pruning matched pixels in 16×16 windows in image space and keeping only the match with minimal pairing error per window.

3.2.2 Repetitive Shape Extraction

Shape Repetition as Match Aggregations

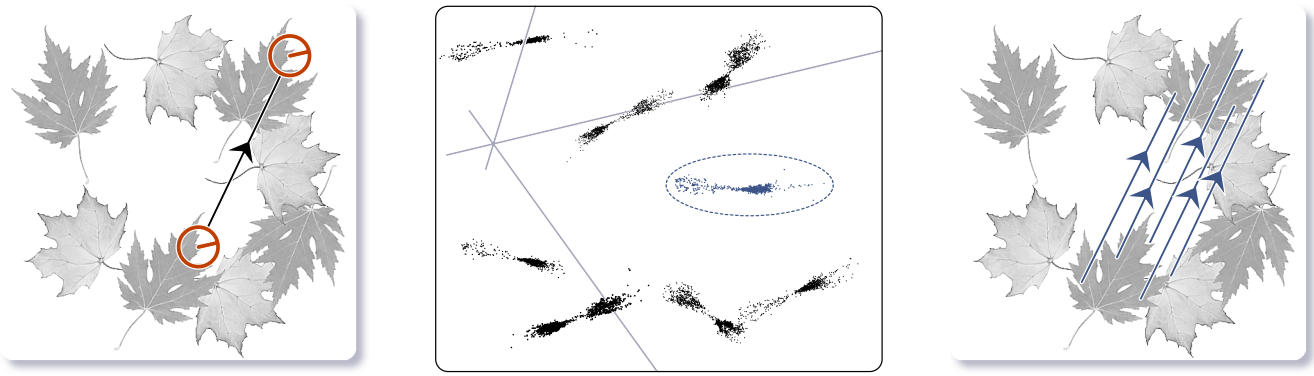
Each neighborhood match defines a unique 2d similarity thanks to the neighborhoods' respective position, and associated scale and rotation. All the established matches then act in transformation space as samples

of some unknown density. Its local maxima correspond to accumulations of transformations. But more importantly, they also attest to the existence of sets of pixels repeating themselves under a roughly common transformation (*cf.* Figure 3.3). We aim at finding those aggregations and use them as hints of presence of repetitive shapes.

We detect such gatherings by *mean shift clustering*, which enables data partitioning with no prior on the final number of classes [CM02]. Clustering is performed on the similarities' 4 degrees of freedom (translation, rotation, and effective scaling). We also enforce spatial locality by taking into account starting positions of the matches during clustering. While all clusters hint the presence of repetitive image regions, their cardinality quantifies their respective relevancy.

Before clustering, we keep the matches corresponding to one half of the most selective pixels only, assuming they yield the most significant information. We also only consider significant clusters, the ones whose cardinality is at least one fourth of the largest one.

Both operations discard "background" pixels, which found either no suitable match (in the case of stochastic backgrounds), either too many of them (near-uniform backgrounds), or did not give rise to consistent transformation clusters.



(a) Descriptor matching (b) Transformation space (c) Local cluster of matches

3.3 Gathering Local Repetition Evidences. Each pair of matched descriptors fully determines a 2d similarity *(a)*, which represents a single sample point in transformation space *(b)*. Finding accumulations of such points is the first step towards repetitive shape recovery *(c)*. For illustration purposes, the 4d transformation space is here projected to 3d space.

Filling up the Blanks

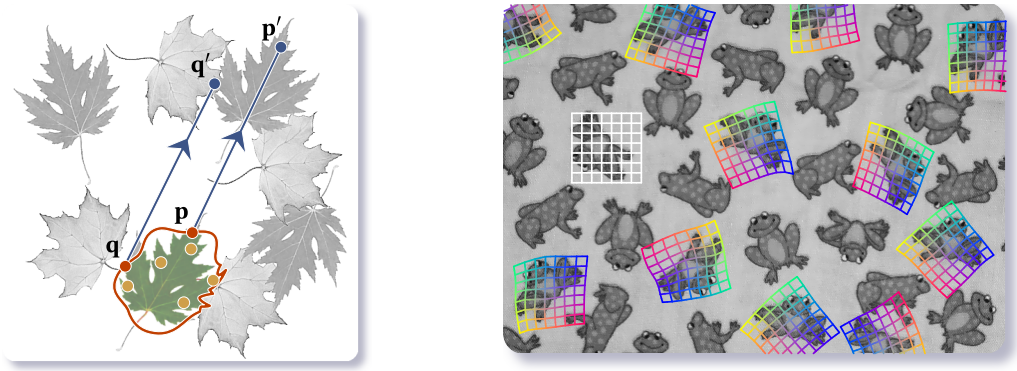
At this point, we have local, but only sparse pairs of pixels following a common transformation. We still need to extract a connected shape, which we obtain by region growing from the starting points of the clustered matches (*cf.* Figure 3.4*a*).

New pixels at the boundary of the ongoing shape are merged to it if the distance between the SIFT descriptors, computed before and after transformation, remains less than a given threshold. By default, it equals 1.5 times the radius used for matching. Special care must be taken at shape boundaries. Indeed, since SIFT descriptors are evaluated over neighborhoods, they are likely to encode part of the background at shapes' actual boundaries. To cope with this, more local versions of SIFT are considered before halting region expansion. But once we resort to those, expansion must be limited.

During this process, self-overlap is forbidden: extracted shapes must not fold onto their transformed counterpart. The order of pixel traversal then becomes important since as the shape grows, some pixel locations get unavailable. We favor regular shapes by prioritizing boundary pixels such as to minimize a constantly updated quality measure. This measure is defined as the ratio of the shape's squared perimeter on its area.

Transformations must also offer more flexibility than plain similarities (*cf.* Figure 3.4*b*). The shapes' mappings are modeled with approximate thin-plate splines, used to describe the behavior across the image of their translation, rotation, and scaling. Those splines are constrained by the matches brought together by the clustering.

After region growing, we obtain the input's *region similarity map*, containing a list of – possibly overlapping –



(a) SIFT-driven region growing (b) Thin-Plate Spline fitting

3.4 Shape Delineation by Greedy Region Growing. *(a)* For shape extraction, we perform region growing from the starting points of the clustered matches (yellow dots). Boundary pixels **p**, **q** are included if their SIFT remains unchanged after projection **p**′, **q**′ respectively. We use a coarser SIFT descriptor when it may encode part of the background (here at **q**′). *(b)* We also fit thin-plate splines in order to better capture subtle deformations. The reference grid is here displayed in white.

regions along with the non-linear mapping toward their duplicates. These regions are still independent and the region similarity map can be thought of as a multi-layer representation of the input. However, not only this representation is over-complete but a lot of information redundancy remains. Intersecting regions give rise to ambiguities which need to be taken care of in order to extract the final image shapes.

3.3 Recovering Pattern Classes

The region similarity map enables us to know all repetitive shapes contained in the input image, as well as the transformations they undergo. All those shapes are still independent though. We must hence gather the ones corresponding to the instances of one same pattern and mold its representative out of them.

3.3.1 Theoretical Analysis

We need to explicitly group the shapes related to the same patterns and compute the input image's *pattern visibility map*. This mapping indicates to which *unique* shape every input pixel belongs. This information is mandatory to finally extract the patterns and eventually deal with occlusions between their instances.
This problem is highly similar to image segmentation, each segment being one of the identified shapes. Piecewise continuous pixel labeling can be achieved via the optimization of a cost function evaluated over a graph. The set of its vertices is composed of the image's pixels –neighboring pixels yielding linked vertices– and as many additional terminal vertices as there are possible labels. First, each pixel node shares a common edge with every label vertex. Then, those edges, except for a single one, must be severed for the pixel to be labeled. All the edges in the graph are weighted, so that any path can be associated with an energy value. This energy can then account for both the contextual information guiding the segmentation and the desired smoothness between neighboring labeled pixels. The segmentation then comes to find a multi-way minimal cut through a flow network.
However, for multi-labeled segmentation, finding this cut is NP-hard and only approximate iterative solutions exist [DJP+92]. Evaluating the cost function is also problematic in our case as it depends on the ongoing segmentation. Indeed, it must favor a label assignment such that a minimal number of maximally-instanced patterns appear on the visibility map. This dependence may cause convergence problems and makes the energy minimization framework unsuited to our problem.

3.3.2 Proposed Heuristic

We decide to use a two-fold approach to finish the analysis of the input. First, we build the *regions' overlap graph* and analyze it to group them together to form classes. Then, given those classes, we compute the pattern visibility map by assigning a unique class and member to each input pixel in an iterative way. Those two separate steps are presented in more details below.

Building and Traversing the Overlap Graph

From the similarity map, we know the locations in the image of the regions before and after their respective transformation. This information and overlaps between these regions allow their partitioning into classes.

We encode those overlaps in a graph: its nodes correspond to the regions and edges are created between them whenever their associated regions, either transformed or not, overlap. Those edges comprise two oriented half-edges –each of them storing the transformation mapping their starting shape to their ending shape– and the normalized strength of the ensued overlap. The overlap strength is equal to the area in pixels of the regions' intersection. Before its assignment to a half-edge, it is normalized by the area of the half-edge's ending shape. Each half-edge can now be interpreted as "to what extend its starting shape, after its associated transformation, contains its ending shape".
This encoding enables us to easily bring together regions sharing *significant* mutual overlap. For an edge to be deemed significant, its half-edges' lowest and highest strengths must exceed specified thresholds. We respectively use 0.25 and 0.75.

At this point, we need to organize the shape nodes and study their transformations, for that step is crucial to get consistent pattern classes. We achieve this grouping via connected component analysis along significant edges, while ensuring the respect of additional constraints. Indeed, each edge describing a transformation, we must guarantee the absence of inconsistent cycles within the ongoing class when adding new nodes to it. Nodes of a same class must not be linked by several paths producing different transformations. Some edges may be associated with the identity transform and nodes adjacent to such edges may naturally overlap. Apart from that case though, we must also prevent overlapping regions separated by other transformation, from ending up in the same class, as it would lead to classes with overlapping members.

Once the graph analysis has been achieved, region nodes sharing the same transformation within the class are merged together in order to form its final members. Now, each class deals with a specific pattern and consists of a set of shapes, repeating themselves throughout the input according to a known set of transformations.

Establishing the Pattern Visibility Map

With the sets of pattern classes in hand, we now study back the input. We assign a class member to each of its pixels in order to get the *pattern visibility map*. This step is mandatory for satisfactory occlusion management between patterns. It also has to robustly identify and lift any ambiguities that would have arisen from spurious or partially-repetitive shapes.
As stated in 3.3.1, using an energy minimization framework would be NP-hard in our case and close to computational intractability. Instead, we propose an iterative method which labels every pixel, first with a unique class, then with one of its members. This two-step approach is necessary to effectively deal with members of a same class sharing a common boundary.

Pixels get ambiguous when they can be assigned more than one class member. Our goal is to pick one unique *visible* candidate. Our iterative approach strives to determine the visibility of the class members in a way that ensures "well-behaved" members. It means a minimal number of classes with a maximal number of visible members.
We start by restricting the candidate members to a single class. First, we flag as *reliable* pixels whose set of candidates are from the same class, and then propagate this initial information. By assuming its membership to different candidates, a pixel can be applied the transformation network of its associated class. Studying the locations where the pixel gets projected by those transformations is paramount to choose among its candidates.

If, while studying the transformations of every candidate, the pixel gets projected onto a position where a reliable class has been determined, without conflict (the reliable class must match the candidate's) nor ambiguity (only one reliable class must be encountered), then we can assign this particular class to the pixel and flag it as reliable. This process is repeated while pixels keep on being labeled. Finally, the class of pixels left unlabeled after this step, is determined by picking the candidate class with maximal score. Those scores quantify "how well" the pixel, if assumed to belong to a class, behaves. Each candidate contributes to its class: we apply its transformations to the pixel and count the number of times it encounters shapes from the same class once projected.
Some pixels may still be ambiguous though, potentially belonging to different members of the same class. This occurs at boundary areas between different instances of the same pattern. We lift this last incertitude by applying a decision scheme similar to the one proposed for the classes: we use the transformations from pixels whose *visible* member is final to iteratively propagate constraints onto other unlabeled pixels. Next, scores are attributed to the candidate members of the pixel's selected class, and the member of maximal score obtains the pixel.

3.3.3 Pattern Extraction

Once the pattern visibility map has been established and all ambiguities lifted, extracting patterns becomes straightforward even if they partially overlap. Indeed, not only we know the positions of the patterns' different instances (corresponding to the classes' visible members), but also the transformations mapping one onto another. Given a pixel on one instance, we can easily compute its corresponding positions on the other instances and then obtain a consistent traversal across them. The final pattern representative is then computed by gathering its instances' pixel colors. To deal with possible occlusions, only pixels appearing at least on two visible instances are taken into account.

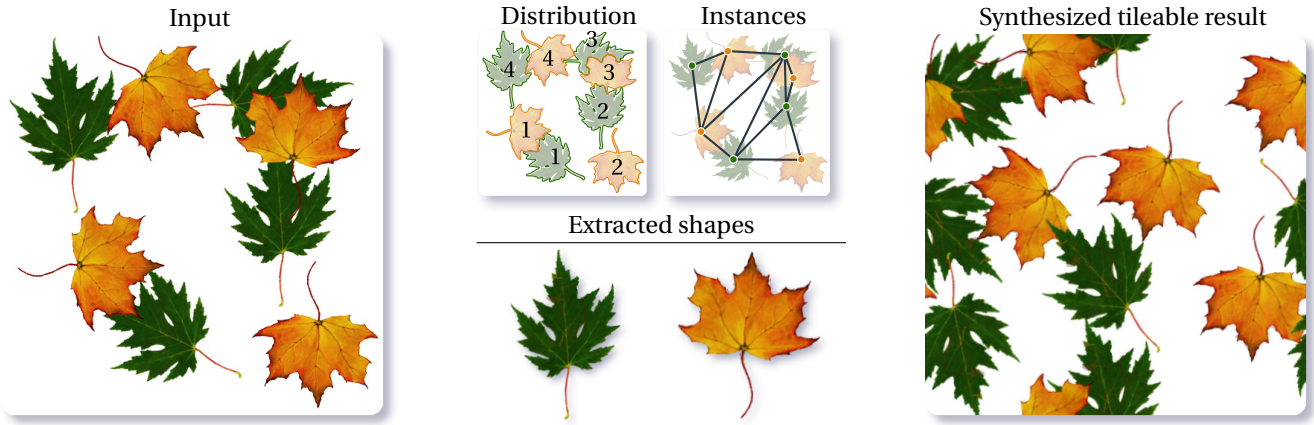
3.4 Controlled Shape Pasting for Raster Arrangement Generation

3.4.1 Re-Synthesis Results

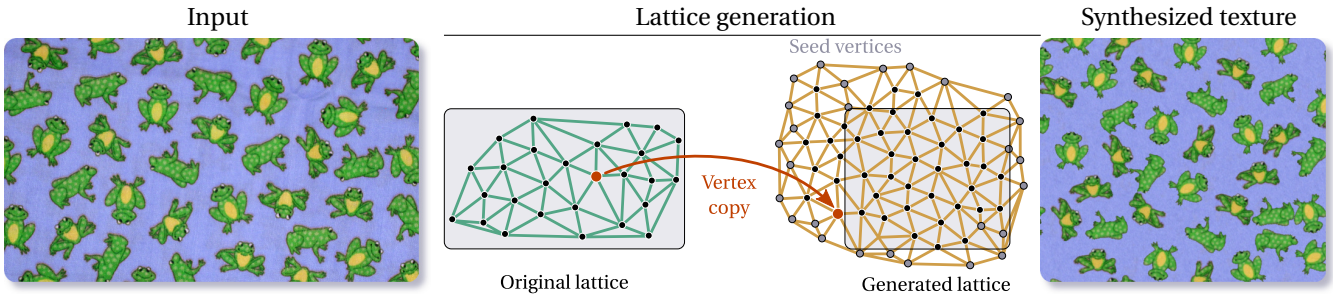
Now having those patterns along with their relative positions at our disposal, we can re-synthesize new distributions of shapes and add high-level randomness to parameters actually defining the input's visual appearance. We now present several application scenarios and show how to apply existing techniques or possibly extend current work to take advantage of the texture representation we propose.

Generating Tileable Textures The simplest application for our method is to convert non-tileable shape arrangement textures into tileable ones. This is made easy once one acquires the image's constitutive shapes. For stochastic placements, we just need to generate some random shape distribution with the only care to respect the output's torus-like topology when rendering shapes crossing the image's borders (*cf.* Figure 3.5). For lattice-based distributions however, achieving tileability is slightly more involved as the transformation group defining the lattice must be compatible with the image's geometry.

Synthesizing Input-Consistent Distributions If we think about the input shape distribution as a parameter of the output, one may want to generate a new distribution remaining close to the input's while being altered by an appropriate degree of randomness. Possibilities range from complete random distributions to faithful replications of the input. To handle the latter case, we build upon recent research in Expressive Rendering which broadened Markov random field-based texture synthesis to element arrangements. To achieve this, they extended the neighboring system, from the image pixel grid to the Delaunay triangulation of their input elements [BBT+06, IMIM08]. We also build a Delaunay triangulation, over the extracted input instances' centers of mass and take a similar approach to generate new distributions out of it. Depending on the desired result,



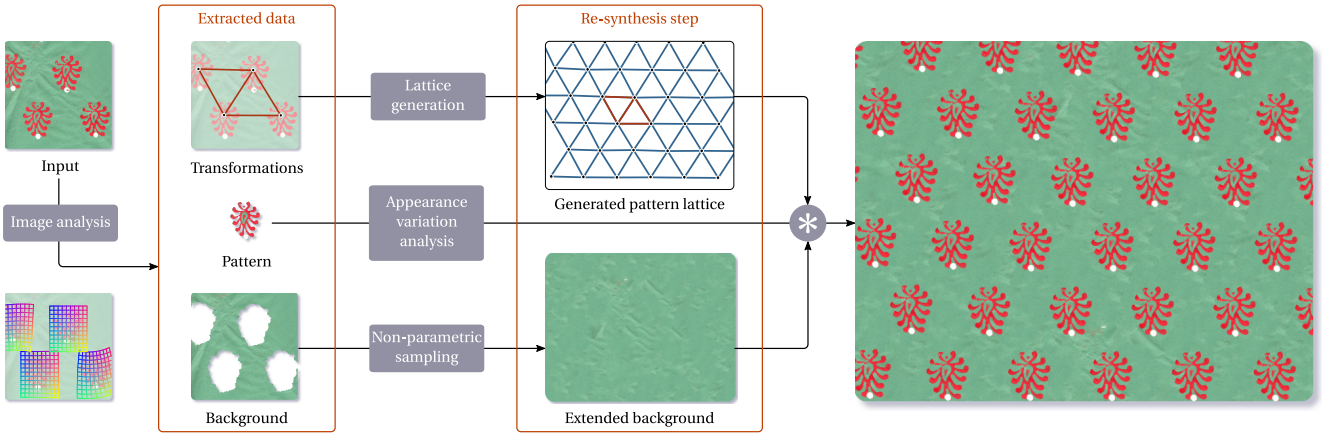
3.5 Example of Tileable Re-Synthesis. This result alone sums up the philosophy behind our raster arrangement method based on the extraction of repetitive shapes. Because of the underlying assumption stating that relevant information is characterized by its reoccurrence throughout the sample, our technique is therefore mostly suited to the analysis of images consisting of shape arrangements such as the one shown on the left. Indeed, repetition of local features is the only means at our disposal to perform our decomposition of the input into basic shapes and obtain a representation of higher-level than the input's original raster representation. Once detected, these shapes can be freely manipulated in order to produce new images and the non-parametric capture of their relative positions in the input is also of great help to ensure visual faithfulness to the provided example. Note that since we infer relevant shapes thanks to their multiple duplicates throughout the image, we do not require them to be entirely visible at once. Partially visible elements can thus also be handled to some extent, as demonstrated for the green leaf of this example whose instances are always partly hidden.



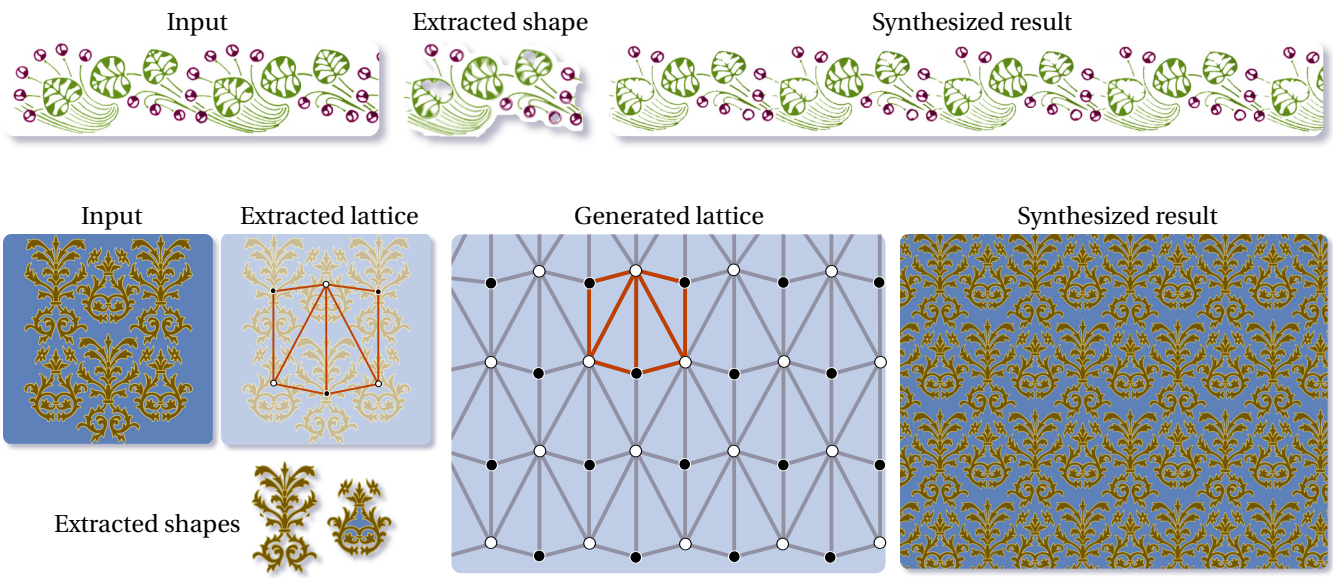
3.6 Non-Parametric Capture of the Shapes' Relative Positions. Extracting the different shapes constituting the input is of course of primary importance, but randomly pasting these to generate a new image is likely to fail to convey the same overall appearance as the input. Shown here is an example where the input image contains different shapes disposed over a somewhat uniform background. Once the different elements have been detected and isolated, we can afford manipulating shapes and not mere pixels or rectangular patches anymore. We can therefore analyze pairwise distances between the shapes' centers scattered throughout the input and notably consider their disposition through the prism of the lattice made of the triangulation of their centers. In order to create a wider, approximatively similar looking lattice for subtending our outputs, we employ a greedy triangulation technique highly reminiscent of NRP research in vector arrangement synthesis [BBT+06, IMIM08]. Independently, the feature-free background is extended using classical non-parametric pixel-based texture generation [WL00].

neighborhood matching is here controlled either by the shapes' positions, orientations or *ids* (*cf.* Figures 3.9, 3.6 and 3.7).

Reintroducing Chromatic Variations Since our analysis is performed on the input's lightness channel only, objects differing only in color are brought together since considered as visually equivalent. We can take advantage of this to study the chromatic appearance variations between instances of a same pattern. As those shapes are related to each other by non-rigid transformations, we place them in a common coordinate system and reduce the dimensionality of their RGB color distribution by PCA [LLH04]. We then generate shapes of slightly varying appearance by modifying the coefficients of the original instance's decomposition onto the obtained eigenvectors (*cf.* Figure 3.9). This treatment is however limited to the case of fully-visible pattern instances.



3.7 Overview of our Lattice-Based Texture Synthesis. After analysis of the input, we dispose of meaningful extracted data *(a)*: its repetitive patterns are expressed as a possibly instantiated object and the set of transformations towards its duplicates in the image. *(b)* From this, we generate a lattice of positions we can paste the original shape onto. The background is extended using non-parametric sampling texture synthesis [WL00]. As a result, we obtain a coherent distribution of patterns which closely mimics the given input.



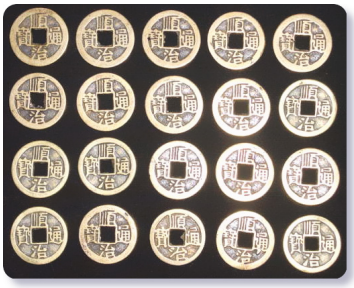
3.8 Results. Two examples of applying our texture analysis technique to redundant, yet not entirely tileable images. The hand-drawn case is particularly challenging since SIFT descriptors easily get very discriminative in presence of high frequencies at the scale of pixels which could impede the extraction of an exploitable, re-synthesis prone shape.

Handling the "Background" Figure-ground separation is an intricate problem which far exceeds the scope of our paper. In our context, the most straightforward definition for background pixels is pixels not instanced after analysis. And, even though counter-example images are easily produced, there are situations where such automatic background extraction becomes satisfactory (*cf.* figure 3.7). This is notably true when background pixels exhibit sufficient isotropy or cannot clearly be assigned a canonical orientation. In this case, we separately extend a wider background texture by using well-adapted non-parametric texture sampling techniques [WL00].

3.4.2 Discussion

3.4.3 Comparison to Related Work

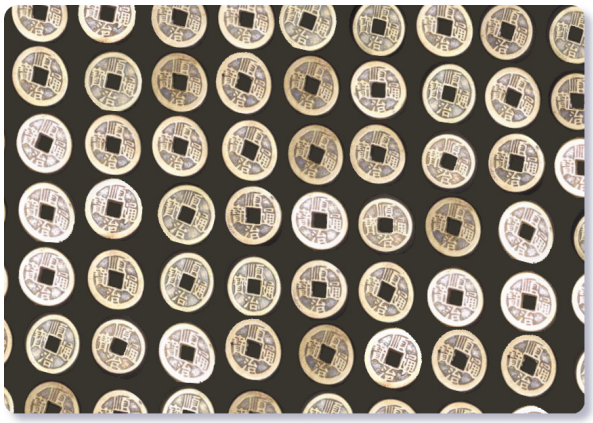
Figure 3.10 shows comparative results between pixel-based existing techniques, namely non-parametric sampling synthesis [WL00] and graph cut textures [KSE+03], and our content-driven method. While they succeed



(a) Input

(b) Average shape

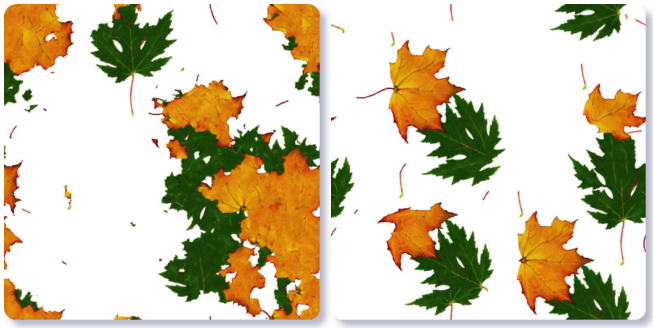
(c) Main covariance eigenvectors



(d) Synthesized output

3.9 Simulating Appearance Variations Using Color Correlations Between Instances. To account for local chromatic variations within the original distribution in the input image *(a)*, a Principal Component Analysis is performed over the different instances' RGB color channels and the set of significant eigenvectors are extracted *(b)*, by using a technique similar to Liu and co-workers' [LCT04]. This vector basis is further used to generate new instances throughout the new texture *(c)*.

in ensuring local continuity, they are bounded by the *a priori* fixed scale of the neighborhood used to evaluate visual similarity. Besides, unable to manipulate primitives other than pixels/patches, they fail to preserve actual structures and shapes end being mixed together.



(a) Non-parametric sampling
$n = 5$ $n = 30$

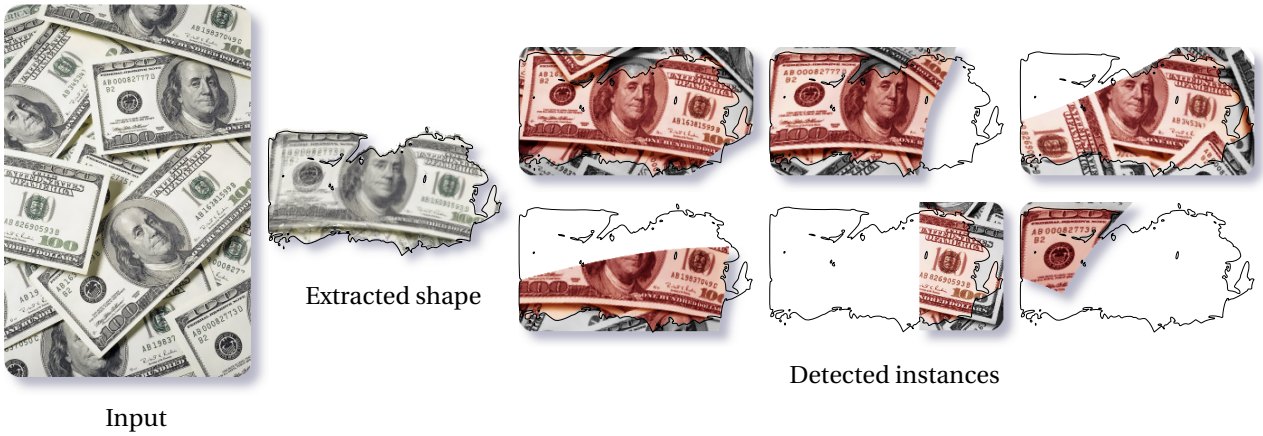
(b) Graph-cut textures

(c) Our method

3.10 Comparison with Main Existing Techniques. By focusing on pixel-based artifacts, both non-parametric sampling *(a)* and graph-cut *(b)* techniques fail to preserve the input's recognizable shapes. Our method on the other hand manages to handle those limitations *(c)*. Increasing the neighborhood size n does help capturing wider structures, but still without handling actual properties of the shape distribution.

3.4.4 Shortcomings

Occlusion Management Though we do handle partially occluded shapes (*cf.* Figures 3.5 and 3.11), some requirements exist. First, in order to be stitched together, occluded parts must share, up to a transformation, a significant overlap to end up in the same pattern class. Second, and more importantly, since our method is devoid of *a priori* knowledge, parts must be detected twice to be deemed of interest. Thus the integrity of the pattern must appear –even separately– throughout the image to be extracted.



3.11 Handling of Partial Occlusions. *(a)* If the input exhibits multiple instances of a single, but always partly occluded shape, *(b)* we can still build its representative pattern thanks to the consistent traversal of its instances allowed by the detected transformations. *(c)* However, its parts must appear at least twice along its recovered instances. This intrinsic limitation of our method is due to its unsupervised nature.

Invariance to 2d Similarities Only Despite the use of thin-plate splines to confer more flexibility during shape extraction, our method relies on the SIFT descriptor and thus only detects shapes up to transformations close to 2d similarities. Reflections for instance are not currently supported, for they would need to add to each pixel a "reflected" version of its descriptor in feature space and weight down computation costs.

Greedy Creation of Pattern Classes Some non-intuitive results find their explanation during the pattern class computation step (*cf.* Section 3.3.2). Once a connected component analysis onto a graph, this step is made dependent on the order of the visited shape nodes by the propagation of transformation constraints along the graph's edges. Several strategies for traversal have been tested but current implementation still needs improvements in that sense.

3.4.5 Computation Costs

The complexity of our algorithm is intrinsically $O(N^4)$ with N being possibly the total number of pixels since we look for clusters of pixel matches. Using discriminative descriptors, accelerated search structures and voxelizing the transformation space is the key of its tractability. It also more directly depends on the input's gradient activity, intuitively speaking "textureness" than its resolution since pixels whose SIFT descriptor shows weak magnitude do not take part in the following computations. Key point matching and shape extraction by region growing are the bottlenecks of our approach. Fortunately, the different parts of our algorithm comply well to parallelism and, in our implementation, all of them, except for the visibility map computation, benefit from multi-threading. The running times indicated below have been obtained on a 64 bit quad-core Intel(R) machine for the different results presented in our paper. References to the figures are given in parenthesis.

	Leaves (3.5)	*Coins* (3.9)	*Frogs* (3.6)	*Wrap* (3.7)	*Dollars* (3.11)	*Frieze 1* (3.8)	*Frieze 2* (3.8)
Description	4 s	5 s	12 s	3 s	3 s	18 s	2 s
Nb features	58,790	77,804	90,251	36,945	56,272	147,764	41,507
Feature dim	22	22	22	21	22	23	22
Matching	**24 s**	**153 s**	**126 s**	**9 s**	**26 s**	**316 s**	**32 s**
Clustering	2 s	3 s	4 s	1 s	2 s	548 s	4 s
Growing	**123 s**	**172 s**	**190 s**	**73 s**	**55 s**	**546 s**	**112 s**
Recovery	5 s	7 s	8 s	1 s	3 s	30 s	8 s

3.5 Conclusions

In this chapter, we have demonstrated a novel technique for raster texture synthesis, especially geared towards textures consisting in the arrangement of discernible shapes. Our analysis stage, contrary to existing techniques', aims at the explicit, unsupervised detection and extraction of the input's constitutive *patterns*. Once they have been recovered, re-synthesis can base itself on higher-level, more meaningful building blocks than individual pixel colors. It can also concentrate on capturing and reproducing relationships between instanced patterns instead of neighboring local pixel variations. We detailed several application scenarios taking advantage of the better insight on the input our content-driven representation can yield. By directly dealing with raster input samples, our gait also enables to use advanced, recent regeneration techniques once confined to vector elements [BBT+06, IMIM08]. Originally designed to tackle non-parametric sampling techniques' weaknesses, our work targets at the handling of cases particularly challenging for traditional example-based texture synthesis. As a counterpart, we can only apply our method to a restricted range of textures.

In the future, we would like to strengthen our technique's robustness to natural images where slight gradient distortions due to perspective or object deformation can endanger the success of our results. To that aim, we would further investigate other local descriptors, either encoding different or complementary visual features of images (intensity- or color-based description) or invariant to more challenging transformations than 2d similarities. A first step toward this latter goal would be to apply Lindeberg's affine adaptation prior the computation of the SIFT descriptors, thus granting affine invariance to Lowe's descriptor. More careful examination of the transformation space could also prove highly beneficial: detecting grid-like placement of match aggregations would lead to the early discovery of transformation groups (such as the ones generating congruent periodic textures) which could be used for optimal shape extraction instead of pixel-based region growing. Replacing our current shape extraction step by a global optimization such as graph-cuts [BVZ01] step would also be worth investigation. The technique detailed in the following chapter also lifts several limitations in terms of the capture of the layout between the texture's constitutive elements, by notably relying on the *global* statistical analysis of their spatial distribution.

CHAPTER 4

Spatial Modeling for the Automatic Synthesis of Vector Element Arrangements

---

The contributions presented in this chapter are the result of a fruitful and much enriching collaboration with Thomas Hurtut[1] and under the direct supervision of professor Joëlle Thollot. Discussions on the topics alluded herein arose after Thomas' arrival in our research unit, and highlighted our common interest in the analysis of expressive renderings by example as an empirical capture of artists' style. At the end of his post-doctoral position during the first quarter of 2009, we worked together for submitting an article proposal to the 7th venue of the *NPAR*[2] conference, and my participation mostly consisted in the writing of the article and its presentation once published. I then helped in part of its implementation after its submission and therefore will not pretend to have an exaggerated scientific ownership over its techniques. In accordance with its first author's will, I still gladly present it here in its published form, the only modifications being the inclusion of additional results, confrontations with existing techniques and more in-depth perspectives for improvement.

This research is directly complementary to the raster texture analysis and synthesis technique presented in the previous chapter. While the latter automatically localized and extracted patterns from raster shape arrangements, the former now examines and captures the spatial relations between vector elements in a statistical modeling approach. Published under the title *Appearance-Guided Synthesis of Element Arrangements by Example*, this work additionally required the valuable help and contribution from Yann Gousseau[3], as well as statistician veterans from the *SAGAG* team[4] in the persons of Jean-François Coeurjolly and Rémy Drouilhet. We are also much thankful to Xavier Descombes Florence Forbes for their inspiring conversations.

This chapter details a technique for the analysis and re-synthesis of 2D arrangements of stroke-based vector elements. The capture of an artist's style by the sole posterior analysis of his/her achieved drawing poses a formidable challenge. Such by-example techniques could become one of the most intuitive tools for users to alleviate creation process efforts. Here, we propose to tackle this issue from a statistical point of view and take specific care of accounting for information usually overlooked in previous research, namely the elements' very appearance. Composed of curve-like strokes, we describe elements by a concise set of perceptually relevant features. After detecting appearance dominant traits, we can generate new arrangements that respect the captured appearance-related spatial statistics using multi-type point processes. Our method faithfully reproduces visually similar arrangements and relies on neither heuristics nor post-processes to ensure statistical correctness.

4.1 Introduction

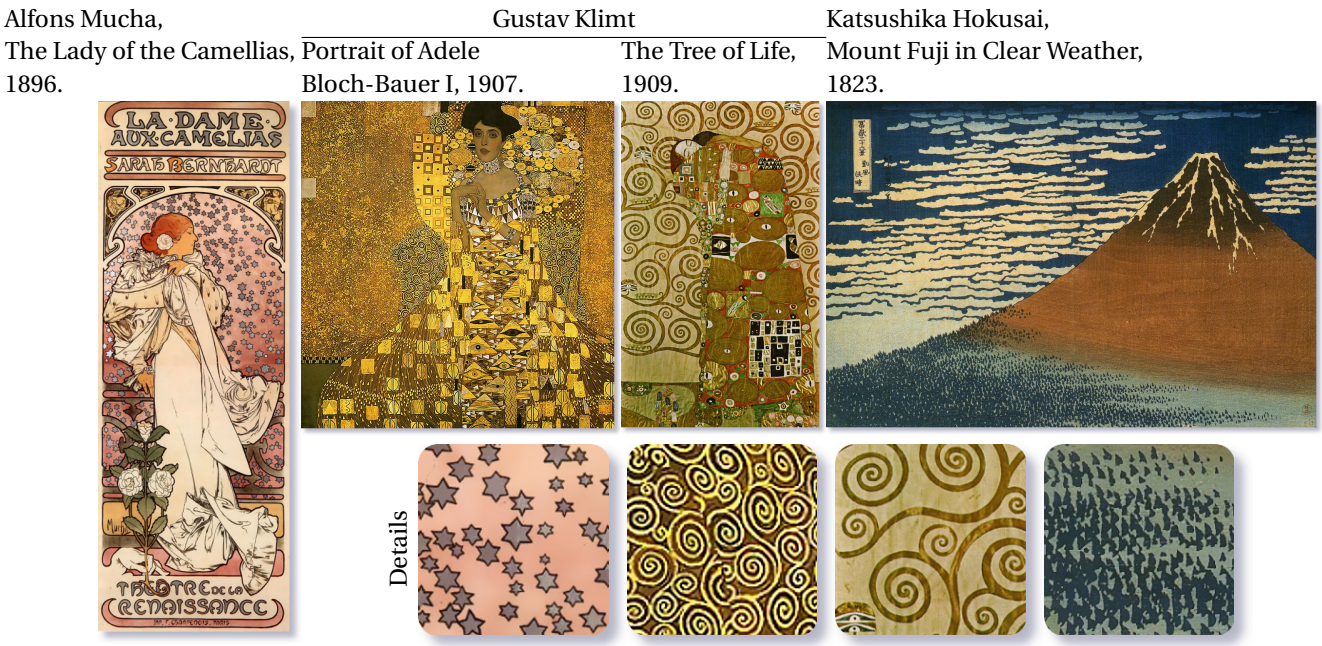
Automated stroke-based rendering systems are common in non photo-realistic rendering (NPR). Successful systems used to generate NPR depictions of 3D scenes or photographs are mainly based on heuristics or hard-coded rendering rules and it is up to the artist to take advantage of them to convey his/her own style. Fewer

[1] Thomas is currently an associate professor in the Paris Descartes University (http://lipade.math-info.univ-paris5.fr/).

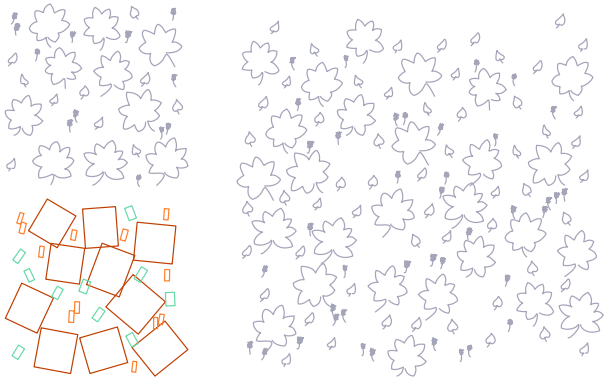
[2] International Symposium on Non Photo-realistic Animation and Rendering

[3] Yann Gousseau is currently an associate professor in the TSI department – *Traitement du Signal et des Images* – at Télécom ParisTech. His professional web site can be found at the following address: http://perso.telecom-paristech.fr/~gousseau/.

[4] http://sagag.upmf-grenoble.fr/



4.1 Examples of Targeted Arrangements in Real Art. Ornamental art, especially *Art Nouveau* iconography, abounds in the repetition of simple, highly-geometrical shapes disposed in a more or less predictable fashion. They mostly appear in artworks whose style is characterized by a strong will to abstraction and stylization. Excerpts from these masterpieces are inspiring illustrations of the kind of arrangements we would like to automatically synthesize with our method. But contrary to the raster texture-centered research presented in Chapter 3, our inputs are now made of *vector* elements. We can therefore focus on the study of their appearance and the restitution of their relative placement.



4.2 Appearance-Guided Synthesis of Vector Arrangements. Given a reference arrangement composed of vector elements *(top left)*, our analysis scheme divides the raw element set into appearance categories *(bottom left)*. Spatial interactions based on appearance can be learned by statistical modeling and exploited to yield visually similar arrangements *(right)*.

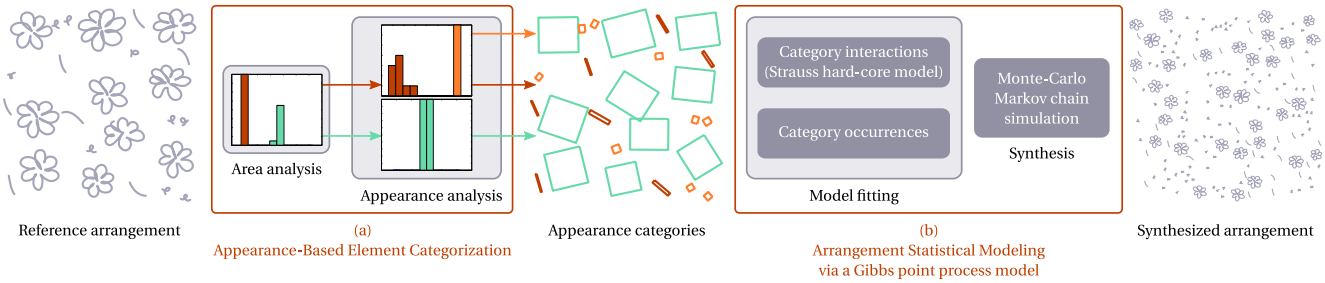
techniques, on the other hand, attempt to automatically learn it instead. In such approaches, the artist provides the system with an example, typically an eventually partially-finished drawing, which has to be analyzed in a way to grasp part of the user's style. The information extracted by this analysis is then used to automatically synthesize new examples visually similar to the original. Such approaches constitute very intuitive tools for artists to handle cumbersome and repetitive tasks, such as creating filling patterns. The main challenge of these techniques is to identify from a limited input what can be assimilated to style and capture it in a way that allows further synthesis. Moreover, to ensure a satisfactory variety of styles, priors that could restrict the scope of supported examples need to be avoided as much as possible.

We focus here on the synthesis of stroke-based elements arrangements. By arrangements, we mean distributions over the 2D plane of visual primitives that do not obey any placement rules or geometric constraints. In such cases, statistics over distances between elements are of primary importance and greatly characterize the input distribution. Our primary goal is then to faithfully reproduce these statistics in order to generate new resembling arrangements.

We claim that, more than sole spatial considerations, the distributed elements' appearance has to be inves-

tigated. More specifically, we believe that reproducing pair-wise occurrences of specific visual cues is mandatory to confer output arrangements the same "feel" as the given example. We therefore propose to model and take into account the elements' appearance in the synthesis. To achieve this, we concentrate on stroke-based elements defined as a set of of path-following strokes. Each stroke is a vector curve allowing us to take advantage of studies in line perception [Jul86] to yield an effective element description.

Note that we assume that already-built elements are provided by the user, not individual strokes. They can either be directly drawn by an artist as a whole (this is the case of the examples shown in the paper), but could also be the result of a stroke clustering pre-process similar to the one proposed by [BBT+06].



4.3 Overview of our method. First, our *appearance-based element categorization (a)* examines the reference elements' shape and divides input elements into appearance categories. We perform this analysis by two successive grouping steps: according to the elements' area, and then to their visual appearance. Once the categories are established, we carry on with the *arrangement statistical modeling* of the user's input *(b)*. Its goal is to capture spatial interactions within, and between categories. For that aim, we infer the parameters of a Gibbs point process model from the categorized example. We can then generate new arrangements by establishing new realizations from the fitted model using Monte-Carlo Markov chain sampling.

4.1.1 Related Work

Our technique exploits the vector elements' appearance to guide the synthesis of new arrangements. Related issues arise in various research fields in Computer Graphics, from raster texture synthesis to NPR stroke-based rendering systems. To provide users with intuitive manipulation handles, we favor example-based approaches over procedural techniques. Since texture synthesis is a rich Computer Science field, our review will focus on example-based methods only, before exploring line appearance encoding.

4.1.1.1 Pixel-based Texture Synthesis

Raster texture synthesis is very inspiring as it focused on example-based approaches early on, see for instance [HB95]. In that case, elements are mere pixels and many successful techniques consider their colors as the realizations of a hidden Markov Random Field (MRF). Their objective is to simulate further sampling to generate new visually-close textures. Most techniques non-perimetrically sample their input and use pixel neighborhood matching as an efficient way to implicitly capture its local behavior [EL99, WL00, Ash01]. However, both the appearance and relative placements of such elements are quite limited. Though extra features can be embedded for improved matching [WY04, LH06], pixels can only be assigned colors and are to follow the lattice structure imposed by the raster grid.

In our case, we aim at producing new arrangements of richer elements distributed over the 2D plane without any prior placement.

4.1.1.2 Patch-based Texture Synthesis

Motivated by the need to capture visual structures lost by the independent process of pixels, texture synthesis involving wider elements, namely pixel patches, were proposed. Few of them however take care of explicitly capturing and handling their relative spatial arrangements.

An interesting example is the case of the texture particle representation [DMLG02]. The input bitmap texture is decomposed into a set of small blob-like elements, coined *particles*. Their relative placement is captured by the distances between neighboring particles' axis-aligned bounding boxes. Neighborhood relationships are determined by successive morphological dilation operations until contact between elements is established. Re-synthesis is achieved via a seeding procedure that uses non-parametric sampling for additions of new patches.

Earlier work by [GZW01] further completes the analysis process as they acquire their elements, introduced as *textons*, by visual learning. They infer the parameters of a texture model defining the input image as the composed realizations of two stochastic *texton processes*. Elements' appearance, density and spatial arrangements are embedded in this unified model whose configuration likelihood is described by a Gibbs distribution. The parameters maximizing it are estimated by gradient ascent, the overall arrangement evolving according to a Markov chain process. Though powerful, their method requires the evaluation of many parameters and the output texton set gets visually relevant only after hundreds of iterations. We still aim at using similar statistical tools since they provide us with an elegant way of enforcing appearance-based statistics over the output. Not only transposing those techniques to vector elements, we also propose faster solutions.

Other techniques, dedicated to near-regular textures, strive to explicitly identify the underlying lattice structure in the input in order to obtain meaningful building elements [LCT04, LLH04]. Regularity between peaks of auto correlation is investigated and *tiles* – minimal set of patches whose periodic repetition defines the texture – are extracted accordingly. Their insight is that the number of possibilities of tiling the 2D Euclidean plane is limited to the finite number of wallpaper groups. However, such approaches are difficult to generalize to non-regular arrangements of vector elements such as those we want to re-synthesize.

4.1.1.3 Extension to Vector Primitives

Many generative NPR systems use *strokes* as their basic rendering building blocks. Such inputs (stipples, curves or brush strokes) can be handled in vector form. Compared to pixel patches, this representation grants a more subtle description of the elements' content. This enables us to find new approaches extending example-based raster synthesis procedures.

First attempts consist of parametric approaches. [JEGPO02] first deal with the synthesis of hatching patterns by modeling 1D hatch sequences with an explicit MRF to reproduce local pair-wise distances between successive elements. The statistical modeling is elegant but difficult to extend to automatic 2D drawing analysis. Similarly, [BBMT06] propose a method to synthesize 2D arrangements of both points and lines and enforce specific statistics on element in a corrective step.

As in texture synthesis, efficient non-parametric sampling techniques were devised, like in Barla and co-authors' subsequent work [BBT+06]. Their main contribution is to yield an intermediate input representation by building elements out of strokes using proximity and continuation constraints. For re-synthesis, they first generate, for a given density, a 1D or 2D set of seed points. Input elements are then pasted to those locations by local neighborhood matching. The employed neighborhood system is the Delaunay triangulation over the elements' barycenters and additional perceptual measures determine the matching. Though used during element building, appearance attributes do not contribute during the synthesis step and supported distributions are uniform due to the Llyod relaxation performed on the seed points. [IMIM08] propose a similar, more synthesis-oriented method. New arrangements are created incrementally and rule-based heuristics ensure the correctness of the ongoing triangulation. Again, elements' visual attributes do not influence the distribution itself and most of Barla's perceptual matching considerations are gone for the sake of interactivity.

Our inputs are similar to Barla's or Ijiri's with subtle differences though. We directly have already-built elements at our disposal contrary to the former, while our elements are not explicitly labeled contrary to the latter. Our approach is also different from their work as we formalize arrangement analysis and re-synthesis as a statistical learning problem.

4.1.1.4 Line Appearance Encoding

One of our contributions is to use dominant element appearance traits as soft constraints influencing the synthesized distribution itself. All previously mentioned techniques only account for spatial considerations to determine their output arrangements. Even methods whose inputs allow relevant appearance analysis overlook this valuable information.

Element's appearance encoding is thus of primary importance. NPR research in style transfer accounts for that concern, a proper representation of the strokes' visual attributes largely contributing to the transfer success. Freeman's work on line drawing stylization uses an implicit representation of line appearance by using training data sets of lines and finding nearest neighbors in the target style set [FTP03]. The user's line drawing style is captured by the WYSIWYG NPR rendering system by encoding over-sketch as offsets relative to the line base path [KMM+02]. Style is then encoded as an explicit MRF which allows further 1D synthesis. [HOCS02] extend their analogy framework to polygonal chain stylization by example and match neighborhoods of the curve's points by comparing point positions and tangent magnitudes. Finally, [BSS07] capture line style as the details functions yielded by a wavelet-like decomposition of the lines.

In this paper, we dispose of compound elements composed of several path-following strokes. This representation of our input allows us to propose more elaborated measurements inspired by line perception studies and use those as relevant features for appearance categorization.

4.1.2 Contributions

Proposing new approaches for both arrangement analysis and synthesis, the contributions of our method are two-fold:

1. We propose a new algorithm to categorize the elements of a given arrangement using perceptually motivated measures.
2. Based on these measures, we use a multi-type point process model to perform synthesis. We chose a model adapted to the capture and restitution of appearance statistics evaluated between, and within element categories.

The main advantages of our method are that it does not require any assumption concerning the input arrangement's distribution and that it performs accurate handling of the elements' appearance. We provide a detailed overview of our arrangement synthesis method's work-flow in Figure 4.3.

4.2 Appearance-based Element Categorization

The first step of our method aims at categorizing the example's elements according to their appearance. If some elements exhibit a similar appearance thorough the input, we want to recognize them as belonging to a same category. Elements considered as unique in the example will be grouped in an outlier category. Note that this step corresponds to an automated solution for the manual labeling of [IMIM08]. Our result could, therefore, be used as an input for their algorithm.

The reasons behind our appearance-driven element categorization are the following. According to the Gestalt law of similarity grouping, the Human Visual System tends to mentally perform perceptual categorization and build groups from isolated elements. Once those ensembles are established, strong visual interactions can arise. Not only elements can be perceived as interacting with the other members of its group, but interactions can also occur at the group level. This phenomenon is illustrated in Figure 4.2 where elements are visually split into three main appearance-based categories, namely the elongated, the cross-like, and the smaller strokes. Since all those elements do not overlap and are mixed quasi-regularly, inter-category interactions are considered repulsive here. On the other hand, intra-category interactions are different and could be described as follows. If considered only with respect to the other members of their own category, the elongated and small strokes seem regularly placed. For the cross-like elements, however, this placement rule does

not apply. This distinction between inter- and intra-category visual interactions is mandatory to devise a good capture of the arrangement's visual attributes. We propose a method that can account for it.

4.2.1 Stroke-based Element Description

[Jul86] studied human perceptual discrimination of textures composed of stroke-based elements, which he called *textons*. In his theory, discriminative features include the element's principal orientation, as well as its number of crossings and extremities. We use these features as our elements' appearance descriptors. Besides, Julesz's textons all shared the same size. To account for that, we add two features to each element's description: its area and elongation. In practice, elements' orientation, elongation and area are estimated on their fitted bounding box. Crossings and extremities are measured directly on the strokes constituting the element.

Next section tackles the issue of grouping together elements that meaningfully share similar characteristics in this feature space. This brings us to directly compare features that capture drastically different visual characteristics. This question is common to all clustering problems in heterogeneous spaces. Before comparison, features are normalized on $[0,1]$. Care must be taken that the $[0,1]$ interval still covers enough visual variation for each characteristic. Orientation is normalized by 2π. Elongation is defined as the ratio of the element's major axis over its minor axis and is normalized by 3. Elements whose elongation before normalization exceeds 3 are tagged as *very elongated* and their associated normalized elongation value is limited to 1. Similarly, elements whose area is larger than 5% of the reference arrangement's area are considered as *large* and attached a normalized surface value of 1. Since we have vector elements at our disposal, we can accurately estimate the curvature of their constitutive strokes. We embed this valuable shape information into our description by counting the number of points of strong curvature along the elements' curves. This feature intuitively corresponds to the number of perceived extremities and is normalized by 10. Lastly, we account for the number of crossings within each element and, as for the extremities, normalize it by 10.

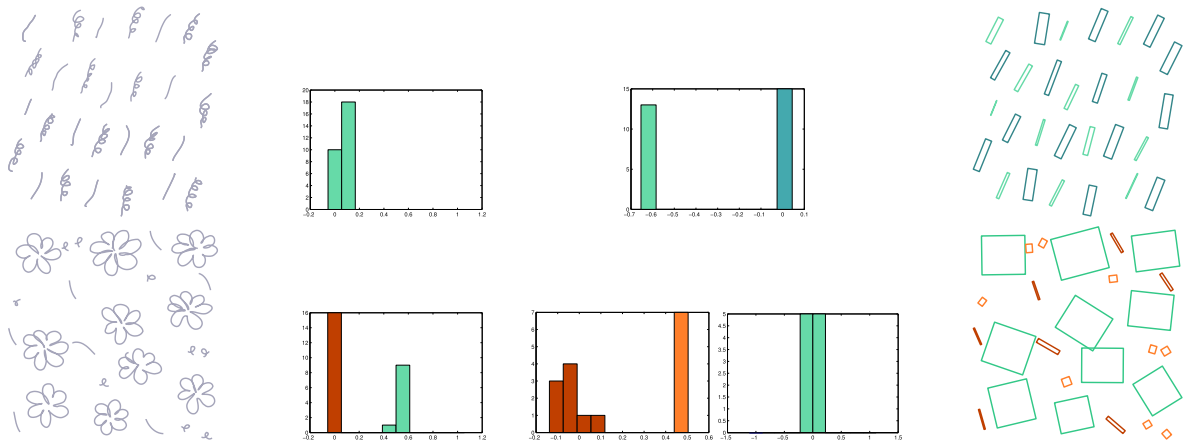
In summary, this gives the following feature set:

Element features	Normalization constant
Area	5% w.r.t. reference arrangement surface
Principal orientation	2π
Elongation	3
Number of extremities	10
Number of crossings	10

Our description is highly discriminative and focuses on the lines' geometrical shape. Yet notice that the proposed line representation is by no means final, and incorporating other features could be possible. One needs to carefully choose those as the more features are added, the more observations in the input must be provided in order to devise meaningful statistics over a more highly-dimensional feature space. Correlation between descriptive components should also be as low as possible to reduce redundancy. For instance, embedding elements' colors in our descriptor set would thus require special care, such as palette extraction, to avoid the classical "curse of dimensionality" issue.

4.2.2 Detection of Meaningful Feature Modes

Our goal here is to categorize elements sharing common visual characteristics once represented in the previously introduced feature space. Our approach is based on two important perception considerations. First, visual perception argues that size is the first information to be perceived for visual recognition tasks. Consequently, we need to bring together elements whose area is roughly the same. Second, as Julesz observed in his studies, it often happens that, depending on the observed elements, not all the descriptive features participate to the perceptual categorization process. Not only useless for categorization, the remaining non-discriminative features also add noise in our sparse feature space which suffers from the usually low number of elements provided by the user. Identifying noise-inducing features is then crucial for ensuring a robust appearance-driven analysis.



(a) Reference arrangement (b) Area histogram (c) Appearance histogram (d) Appearance categories

4.4 Categorization process. Given a reference arrangement *(a)*, the modes of the area histogram are first detected *(b)*. In this example, one mode is found for the top row, while two are identified for the bottom row corresponding to the small and large elements. For each resulting area category, modes of the appearance histogram are then computed *(c)*. Here, two modes are detected for the first row, the most discriminative feature being the number of crossings. The bottom row displays the appearance histogram of small elements where two MM-modes appear *(blue and green modes)*. Only one MM-mode has been detected on the appearance histogram *(red mode)* for the larger element set. The resulting categories are finally shown with corresponding colors *(d)*.

Our categorization scheme thus falls into two stages. First, we categorize elements according to their area. Second, for each of the obtained groups considered individually, we perform another categorization step according to the elements' dominant appearance, computed via dimensionality reduction on the remaining four appearance features. Figure 4.4 illustrates this two-step scheme.

Element grouping according to element area and appearance is established by detecting relevant modes of those two features' density that we approximate by histograms evaluated over the reference arrangement. Because of the lack of prior concerning the number of expected categories, we rely on the *a contrario* method proposed in [DMM03] as our mode-seeking procedure. We recall this method in the following section to make the paper self-contained.

A contrario Methods

A contrario approaches have been successfully applied to many Computer Vision problems among which the analysis of histogram modes. The main insight is to rely on a general perception law called the Helmholtz principle which states that an event is perceptually meaningful if it is unexpected. More formally, if the expectation of its occurrences is low under a random assumption.

In the case of histogram analysis, the random assumption is that the descriptor values are i.i.d. uniformly in the L histogram bins $\{b_1, \ldots, b_L\}$. Let us consider an interval noted $S_{i,j} = \{b_i, \ldots, b_j\}$ with $i \leq j$. The prior probability $p_{i,j}$ that an element has its feature descriptor in $S_{i,j}$ is then $p_{i,j} = (b_j - b_i + 1)/(b_L - b_1 + 1)$. Following [DMM03], we define $S_{i,j}$ as an *ε-meaningful interval* if

$$N\mathscr{B}(p_{i,j}, N_E, k_{i,j}) < \varepsilon \tag{4.1}$$

where $N = L(L+1)/2$ is the number of possible connected sets of bins; N_E is the number of input elements; $k_{i,j}$ denotes the number of elements in $S_{i,j}$, and $\mathscr{B}$ is the tail of the binomial distribution:

$$\mathscr{B}(p, n, k) = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

The quantity $N\mathscr{B}(p_{i,j}, N_E, k_{i,j})$ can be interpreted as the expectation of the bins from $S_{i,j}$ to occur by pure chance. If this estimate is very low, such bins constitute a meaningful interval. The ε parameter has been

shown to cause a logarithmic ε-dependency on meaningfulness, making such approaches robust with respect to their unique parameter [DMM00]. When set to 1, this leads to the following intuitive interpretation: bins appearing at least once in a random situation are considered as meaningful events.

In order to separate modes inside meaningful intervals, we can similarly define *ε-meaningful gaps* within the distribution histogram as the intervals containing fewer points than the expected average. We say that $S_{i,j}$ is an ε-meaningful gap if

$$N\mathscr{B}(1 - p_{i,j}, N_E, N_E - k_{i,j}) < \varepsilon$$

A *meaningful mode* is defined as a meaningful interval that does not include of meaningful gaps. Lastly, in order to forbid the case of non-disjoint meaningful modes, a meaningful mode is said to be *maximal* if it does not contain, and is not contained in another mode showing greater meaningfulness. Maximal meaningful modes are mentioned as MM-modes in the rest of the paper.

Categorization Algorithm

Using *a contrario* histogram mode detection, we then obtain the following categorization algorithm. First, we compute the MM-modes of the element area histogram estimated over the complete arrangement. This provides us with a preliminary set of categories. Any connected interval of bins which does not belong to an MM-mode is considered as an outlier area category.

Second, for each of these categories (including the possible outlier categories), we perform dimensionality reduction on the four remaining appearance features by Robust Principal Component Analysis [HRV02]. We then identify the MM-modes of these features after their projection onto their first principal component. Each found MM-mode defines an appearance category. Similarly to step one, for each appearance histogram, any connected interval of bins which does not belong to an MM-mode is considered as an outlier appearance category.

In the context of example-based methods, we consider arrangements that have a rather low number of elements, typically below a hundred, and thus a low number of distinct appearance categories. This restricts the precision of the histograms we can analyze. In all our experiments, the distribution of features is estimated on 10 bins, but the discretization scheme can be made more accurate as more input elements are provided by the artist. Likewise, if an area category contains less than 10 elements, we do not split it any further.

4.3 Statistical Arrangement Modeling

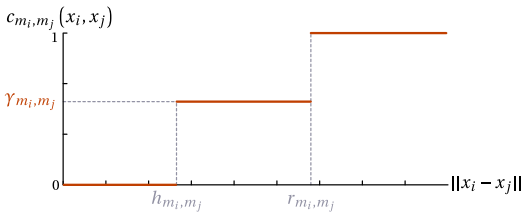
This section presents the statistical process that models the spatial arrangement of categorized elements. Once the parameters of this model are learned on the reference arrangement, the synthesis step consists in running realizations of this model at the desired scale, shape, or density needed by the user.

4.3.1 Multi-type Point Process Model

With the input's appearance categories at hand, we now investigate the elements' relative positions from the perspective of their visual aspect. For that aim, we propose to capture their spatial arrangement via a multi-type point process, a statistical model dedicated to the analysis of interactions between a finite set of typed categories. By considering pair-wise element distances as interactions between our established categories, we implicitly grasp the underlying correlation between the elements' appearance and their spatial organization. This model accounts for the interactions gathered over the whole input and supports a wide range of distributions, from stochastic to near-regular.

In our specific case, we assimilate the point data resulting from a realization of this model to the set of the N_E input elements $\mathbf{x} = \{x_1, \ldots, x_{N_E}\}$. Given an element $x_i \in \mathbf{x}$, we associate its corresponding appearance category label m_i to it, m_i being taken from the N_C possible categories labels stored in the label set $\mathscr{M}$. It should be noted that, since $N_C < N_E$, the labels m_i, m_j, may refer to the very same appearance category even though they are related to two distinct elements x_i and x_j.

$$c_{m_i,m_j}(x_i,x_j) = \begin{cases} 0 & \text{if } ||x_i - x_j|| < h_{m_i,m_j}, \\ \gamma_{m_i,m_j} & \text{if } h_{m_i,m_j} \leq ||x_i - x_j|| < r_{m_i,m_j}, \\ 1 & \text{otherwise.} \end{cases}$$



4.5 Interaction Probability Function Profile.

A way to construct a point process model is to write down its probability density function (PDF) with respect to a Completely Random Situation. Such point processes are called *Gibbs point processes* and offer many advantages. Manipulating their PDF to make them account for intricate interactions is easy and further simulation is ensured by well-known Monte-Carlo Markov Chain algorithms. Since we focus here on pair-wise interactions between element categories, we can define our model's PDF, noted $f(\mathbf{x})$, as follows (see [Rip81] for further details):

$$f(\mathbf{x}) \propto \left[\prod_{x_i} d_{m_i}(x_i)\right]\left[\prod_{x_i \neq x_j} c_{m_i,m_j}(x_i,x_j)\right] \tag{4.2}$$

where $d_m(\cdot)$ is the occurrence probability function of elements from the m category and $c_{m,m'}(\cdot,\cdot)$ is the interaction probability function between the m and m' categories.

A good rule of thumb for statistical modeling is to exploit models whose number of parameters does not exceed the number of observed data. Here, we thus use the simple *Strauss hard-core interaction* which directly relates interaction probability between appearance categories to the Euclidean distance between their elements:

The explicit definition of the interaction probability function of a category pair $m, m' \in \mathcal{M}$ then requires the estimation of three constant parameters noted $h_{m,m'}$, $r_{m,m'}$, and $\gamma_{m,m'}$. The first two are distance thresholds, called *hard-core distance* and *trend threshold* respectively. The last one is a scalar in range $[0,1]$ defining the *interaction strength* and its tuning enables us to model a variety of arrangements from regular to random. Since our proposed interaction functions are symmetric, we just need to evaluate $3N_c(N_c+1)/2$ interaction parameters to completely define our statistical arrangement model, with N_C being the total number of appearance categories.

4.3.2 Estimation of the Model Parameters

The multi-type Strauss hard-core model is a generic descriptive model that can reproduce various spatial arrangements. This diversity is embedded in the parameters that need to be estimated from the input arrangement by likelihood maximization.

Given the limited set of provided elements, we need to make an important simplifying assumption to ensure a tractable statistical fitting. We suppose the reference arrangement is *stationary* which intuitively comes down to presuming that the artist draws homogeneously over the reference surface. Our re-synthesis still allows the creation of inhomogeneous element distributions. This simplification allows us to treat the categories' occurrence probability functions $d_m(\cdot)$ as constants during the estimation of the parameters. We denote this set of constants Δ.

Moreover, the statistical approach we adopt to estimate the parameters of the multi-type point process is hazardous for extremely small categories. In practice, we assume that the user did not draw groups of similar elements containing less than three elements.

4.3.2.1 Hard-core Distances $h_{m,m'}$ Between Category Pairs

Given a pair of appearance categories $m, m' \in \mathcal{M}$, the hard-core distance obtained by likelihood maximization estimation $h^*_{m,m'}$ corresponds to the minimum distance between pairs of elements picked from the specified

categories:

$$h^*_{m,m'} = \min_{\substack{x_i,x_j \\ m_i=m \\ m_j=m'}} ||x_i - x_j||$$

4.3.2.2 Trend Distances $r_{m,m'}$ Between Category Pairs

Maximizing the trend distances' likelihood estimator is more involving and intuitively corresponds to finding the circular window radius from which the reference arrangement is seen to be the most regular.

To compute that radius value, we use Ripley's $\mathscr{L}$ function which quantifies the deviation of the arrangement, when investigated at a specified scale, relative to a Completely Random Situation [Rip81]. Here follows its formulation :

$$\mathscr{L}_{m,m'}(r) = \sqrt{\frac{\mathscr{K}_{m,m'}(r)}{\pi}} - r$$

$\mathscr{K}_{m,m'}(r)$ is the expected number of elements from the m category lying at a distance r of a randomly picked element of the m' category. As such, it gives an estimate of the element density evaluated at a given scale of a category with respect to another and is normalized in a way that a purely random distribution yields a constant value $\mathscr{L}_{m,m'}(r) = 0$ for all r. Distributions which exhibit more regularity present a negative $\mathscr{L}_{m,m'}$ profile. We thus look for r^*, the first value for which $\mathscr{L}_{m,m'}$ reaches a local minimum. This attests that regularity occurs with maximal amplitude at that scale.

4.3.2.3 Interaction Strengths $\gamma_{m,m'}$ and Category Occurrence Probabilities d_m

The estimation of the remaining models parameters involves the maximization of the PDF of our model evaluated over the reference arrangement $f(\mathbf{x})$. Finding the optimal parameter sets $\Gamma^* = (\gamma^*_{m,m'})$ and $\Delta^* = (d^*_m)$ comes down to find the best "explanation" by our statistical model of the observed input. However, as Equation (4.2) suggests, $f(\mathbf{x})$ is defined up to a normalization constant whose explicit evaluation is intractable. To circumvent this problem, we instead maximize the following log pseudo-likelihood involving ratios of f:

$$\sum_{x_i} \log\left(\frac{f(\mathbf{x})}{f(\mathbf{x}/\mathbf{x_i})}\right) - \frac{1}{N_c}\sum_{m=1}^{N_c}\int_{W_{\mathscr{R}}} \frac{f(\mathbf{x}\cup u_m)}{f(\mathbf{x})}du \tag{4.3}$$

where $W_{\mathscr{R}}$ corresponds to the input drawing window and u_m to an element from the mth appearance category. The involved PDF ratios can be understood as such: given the fixed element distribution $\mathbf{x}$, they quantify the conditional probability of observing an element at a specified location u. The first term of Equation (4.3) favors locations where observed elements actually lie, while the second term penalizes all the other locations within the drawing window W. The integral is usually estimated using a grid on $W_{\mathscr{R}}$ where locations u are the centers of each grid cell weighed by its surface. In our experiments, we use a regular grid.

This formula was first proposed by [Bes77] and later extended by [JM91]. Its suitability to a wide range of Gibbs point processes has been recently proved by [BCD08]. It admits a unique extremum in the (Γ, Δ) parameter space which we find using a Newton-Raphson approach.

4.3.3 Synthesis by Markov chain Monte-Carlo

As stated in Section 4.3.1, one noticeable strength of Gibbs point process models is their easy simulation using Markov chain Monte Carlo methods. This interesting property provides us with a convenient means to generate new arrangements that apparently obey the same stochastic process as the provided input. Since we cannot directly sample from $f(\mathbf{x})$, we construct a Markov chain whose set of vertices coincides with the set of elements from the reference arrangement $\mathbf{x}$ and whose equilibrium distribution is to converge to our fitted model's PDF $f(\mathbf{x})$.

We can now compute new realizations of our statistical model over a synthesis window $W_{\mathscr{S}}$, namely new element arrangements, by using a variant of the Metropolis-Hastings algorithm adapted to point processes [GM94].

randomly initialize output arrangement $\mathbf{x}_0 = \mathbf{x}$ s.t. $f(\mathbf{x}) > 0$
for time-steps t from 1 to T do:
  alter current arrangement $\mathbf{x}_t$ using one of the two following equiprobable perturbations:
  Element birth:
    add an element u at random location in $W_{\mathscr{S}}$
    assign random category label m_u to u
    candidate arrangement $\mathbf{x}' = \mathbf{x}_t \cup \{u\}$
    compute acceptance rate $R_b = \frac{f(\mathbf{x}')}{f(\mathbf{x}_t)} \frac{n_t}{A_t}$
  Element death:
    pick random element u from $\mathbf{x}_t$
    candidate arrangement $\mathbf{x}' = \mathbf{x}_t / \{u\}$
    compute acceptance rate $R_d = \frac{f(\mathbf{x}_t)}{f(\mathbf{x}')} \frac{A_t}{n_t}$
  if $R_{b/d} > 1$, then accept perturbation ($\mathbf{x}_{t+1} \leftarrow \mathbf{x}'$)
  else accept perturbation ($\mathbf{x}_{t+1} \leftarrow \mathbf{x}'$) with a probability $R_{b/d}$
  otherwise keep current arrangement unchanged ($\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t$)

4.6 Arrangement Synthesis by Metropolis-Hastings Sampling.

In the pseudo-code provided Figure 4.6, we denote $\mathbf{x}_t$ the state of our Markov chain at time-step t. For a specified number T of iterations ($T = 10^5$ in our experiments), we slightly perturb $\mathbf{x}_t$ by introducing or removing one element and obtain a new candidate state for the chain $\mathbf{x}'$. These elementary perturbation events, respectively coined the *birth* or *death*, are effectively taken into account if they satisfy an acceptance rate criterion. Acceptance rates for births and deaths, called R_b and R_d, depend of the ratio of the model's PDF evaluated over $\mathbf{x}_t$ and $\mathbf{x}'$, as well as the current arrangement's area A_t and element number n_t.

The simulation output is a spatial distribution of category labeled elements. We finalize our synthesized arrangement by directly pasting onto each output element's location a reference element randomly picked from the correct appearance category.

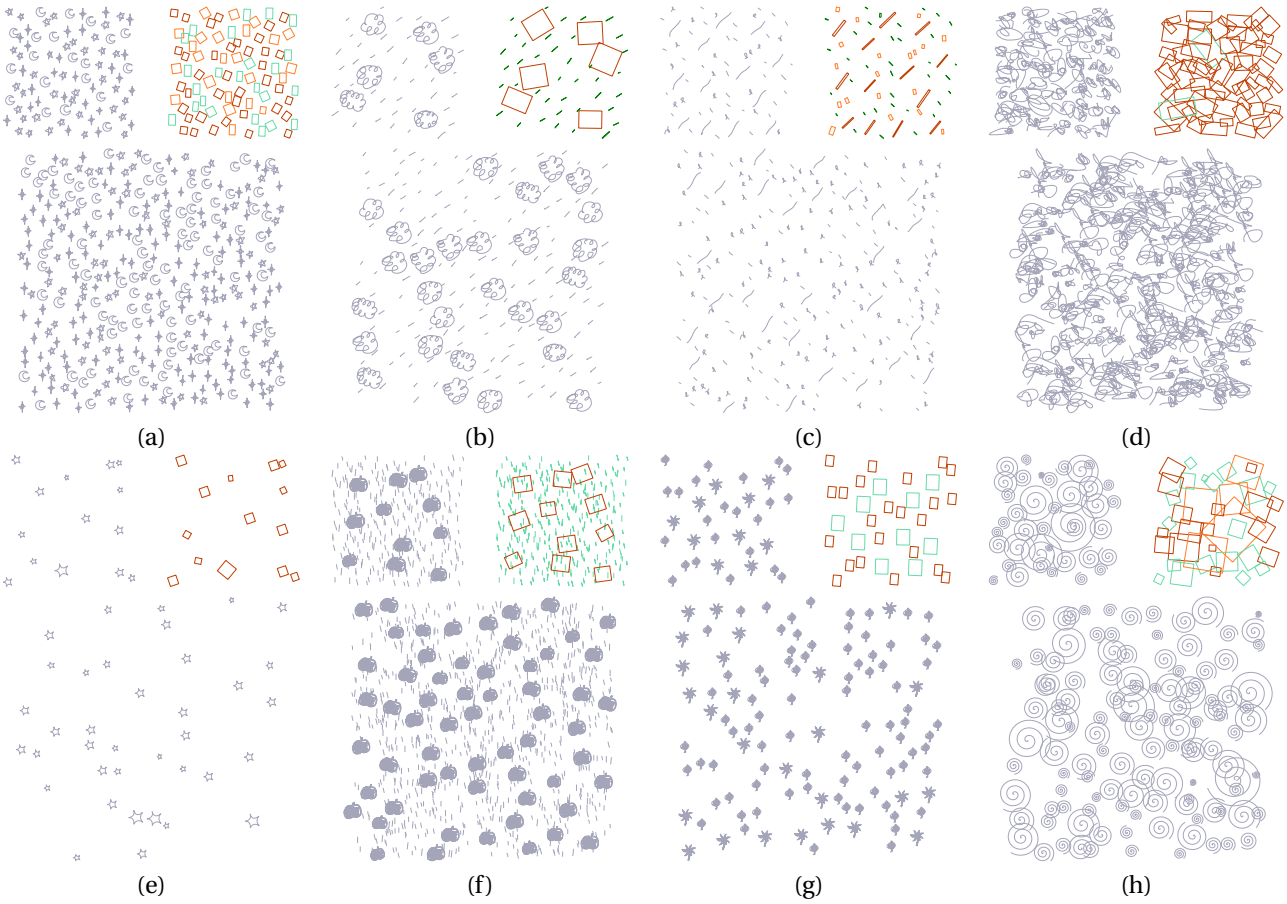
4.4 Results and Discussion

We now present some results and put our technique into perspective with previous methods before discussing its current shortcomings.

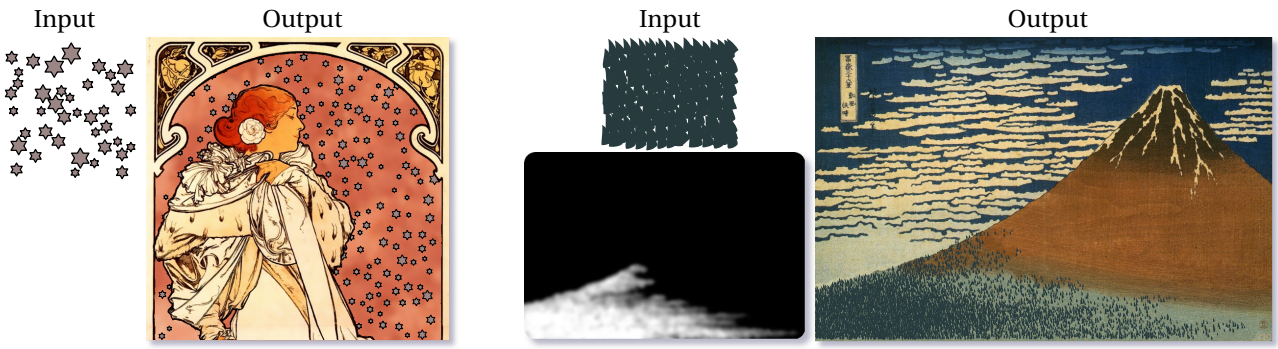
4.4.1 Experimental Results

Examples of categorization and synthesis are shown Figure 4.7. Those examples attest that the Strauss hard-core process can reproduce various kinds of element distributions, from fairly regular to completely random (e.g., Figures 4.7-*(a)* and 4.7-*(e)* respectively). Thanks to the global multi-type optimization procedure, distances between elements are adjusted according to the interactions occurring within and between categories. This important property is especially visible in Figure 4.7-*(b)* where the large elements push back the smaller ones beyond the hard-core distance estimated between the two involved categories. Techniques relying on regular element distributions cannot respect such placement constraints, and in this case, element overlap would then occur.

[5] *Adobe Systems* being the right holder over Ijiri's original implementation, we had to resort to ours to provide these results. And in spite of the care provided to ensure faithfulness to their published algorithm, we cannot guarantee the exactness of our program with respect to theirs.



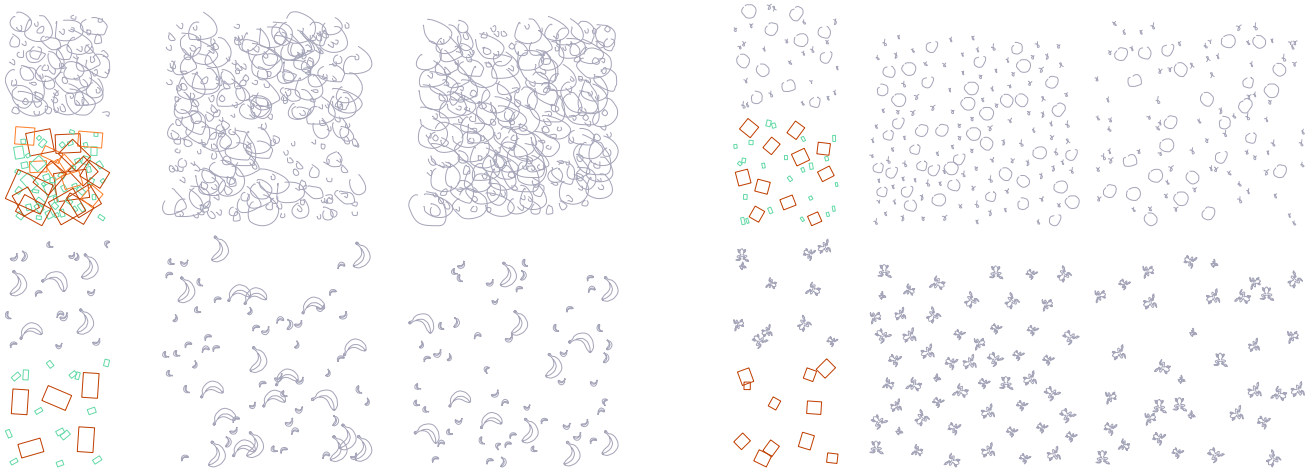
4.7 Synthesis Results. These generated arrangements show the variety of spatial distributions that our model can handle. The multi-type Strauss hard-core model captures distributions ranging from fairly regular to random, *(a)* and *(e)* respectively. This diversity can be observed inside each category of elements. For instance, apples in *(f)* are regularly distributed, whereas the background is randomly arranged. Similarly, interactions between categories can also vary from repulsive to random, *(g)* and *(h)* respectively.



4.8 "Emulating the Masters."

Once our model's parameters have been estimated over the input, new arrangements can easily been synthesized onto various shapes, with possibly different element densities. A gray-level brush-like tool can then be used to intuitively draw arrangements of user-specified densities as illustrated in Figure 4.11.

In terms of performances, both appearance categorization and statistical fitting are interactive, the bottleneck of the method being the Metropolis-Hastings sampling procedure used for re-synthesis. All examples provided in the paper take about 5 seconds to be categorized and generated on a standard PC. Improvements can be investigated to reduce this computational load. More sophisticated perturbations in the MCMC procedure, such as translation and rotation of elements, could be investigated [Gre95]. We could also use a spatially discretized grid during sampling, since our application does not meet accuracy requirements as high as the



4.9 Comparisons. *(left)* These examples motivate our appearance-driven categorization. All arrangements are generated by a Strauss interaction model, the ones in the center via a mono-type point process overlooking interactions between elements of different appearance. Note the apparition of holes or overlaps which are absent from the input. *(right)* The upper reference is the combination of a more and less regularly distributed sets of elements (the circles and crosses respectively). The approach taken by Barla *et al.* enforces regularity over the output while our approach captures the element interactions and preserves them. The lower example consists of a non-uniform arrangement which cannot be represented by the distribution obtained after Lloyd relaxation. A similar phenomenon would arise if using Ijiri *et al.*'s seeding procedure as close seeds are merged below some distance threshold and additional seeds are created in empty regions.

classical uses of such statistical tools.

4.4.2 Comparisons With Related Work

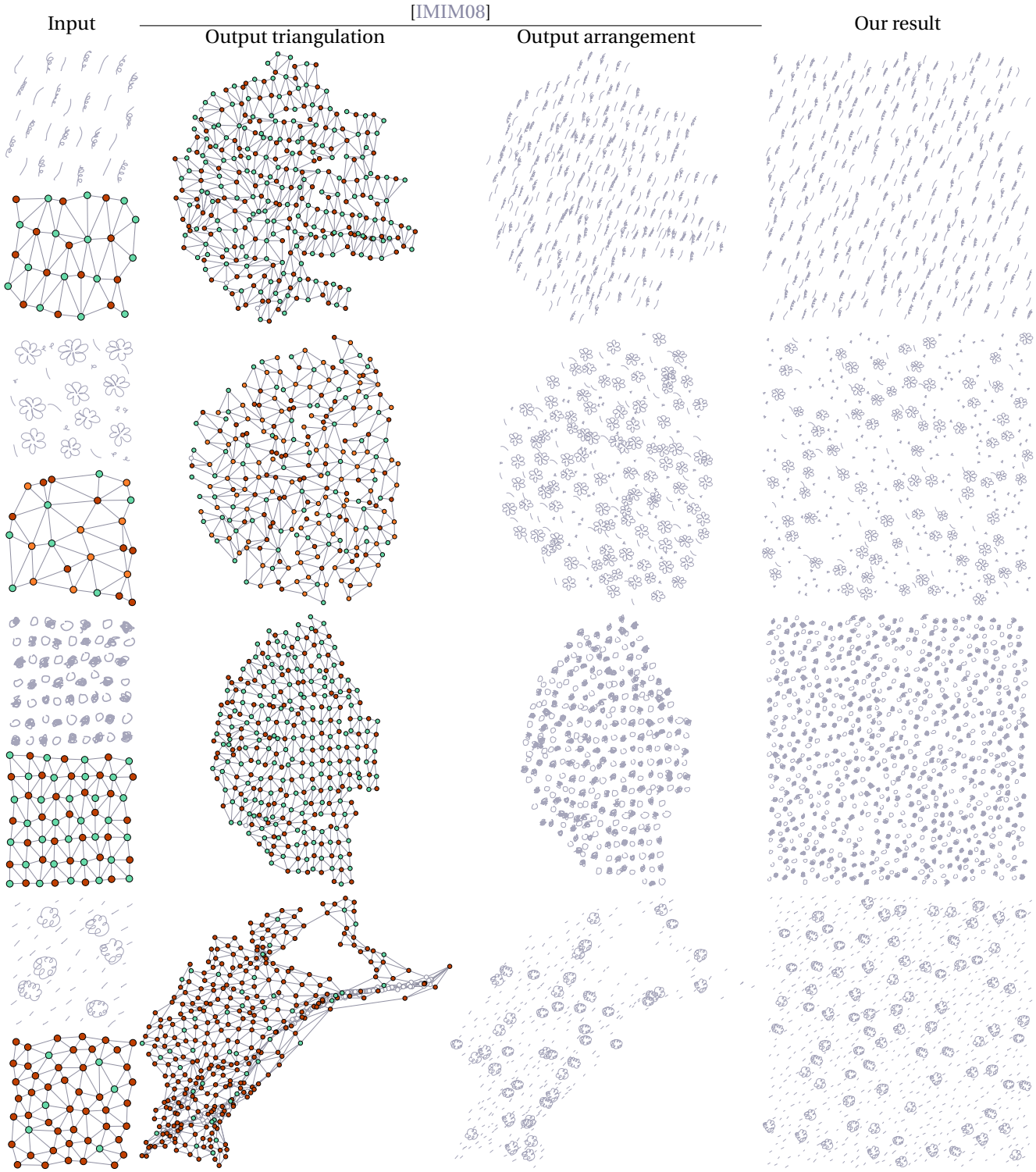
Improved Handling of Appearance

The results on the left side of Figure 4.9 demonstrate the importance of our appearance analysis step. Arrangements generated by a *mono-type* Strauss hard-core model, which considers all elements as visually equivalent, are not fully satisfactory. Even though the overall spatial distribution of the elements' locations is captured, undue holes or overlaps occur and compromise the resemblance of the results with the provided reference.

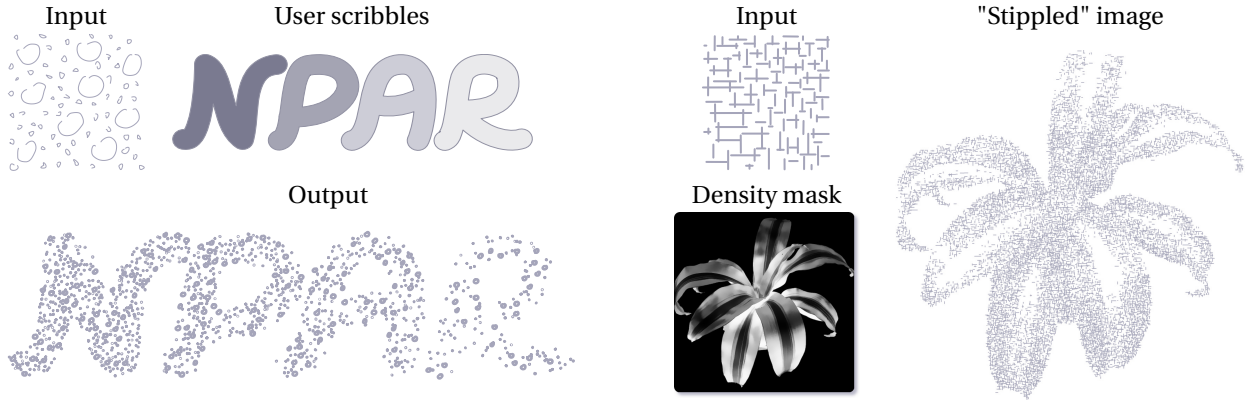
[BBT+06] do account for visual similarity to some extend, by notably defining a perceptual distance used to compare element neighborhoods defined over the Delaunay graph and pick the input element to stitch to a given output location. Their measure, however, only relies on the elements' bounding boxes, whereas our approach integrates more perceptual features enabling the distinction of elements of comparable bounding box. The handling of the elements' appearance by [IMIM08] serves the very same purpose of guiding neighborhood matching. They do not provide any automatic analysis method though, as they require the user to manually label the different elements. Finally, our technique directly correlates statistics between appearance and spatial locations of the elements over the whole input and not just local neighborhoods.

Supported Element Distributions

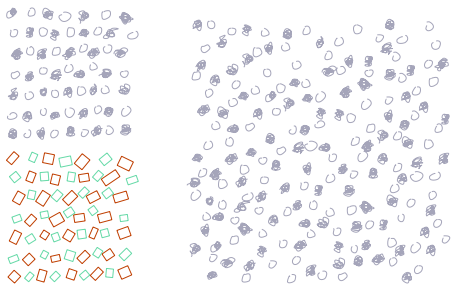
As already presented Figure 4.7, our method can faithfully reproduce a wide spectrum of element distributions. We notably capture *non-uniform* distributions, which is an improvement over the approaches proposed by Barla *et al.* and Ijiri *et al.*. Re-synthesis results displayed on the right side of Figure 4.9 attest that fact. Those consist of sets of irregularly distributed elements whose spatial organization cannot be captured by the point distribution resulting from a Llyod relaxation used in [BBT+06]. As such, Barla's output arrangements always seem to follow an underlying hexagonal lattice structure. The procedural technique suggested by Ijiri *et al.* also leads to similar results as they devise several growing rules – such as the seed merging and empty space-filling



4.10 Comparison. Shown above are comparisons between our method and Ijiri *et al.*'s procedural arrangement synthesis [IMIM08] [5]. Only the placement of the output elements is in question here as Ijiri's technique does not provide any automatic means for appearance categorization. Elements of distinct appearances are displayed as colored dots over the triangulations upon which Ijiri's method performs its computation (white dots being seeds). Although their technique works for most uniform cases, it exhibits a high sensitivity to the input triangulation's topology which often requires user correction to properly handle strongly-regular arrangements *(3rd row)*. More problematic is their greedy element pasting procedure which, only driven driven by 1-ring neighborhood matching, may create overlaps unseen in the input *(2nd to 4th rows)*. Lastly, along the course of its execution, more and more partially-matchable neighborhoods accumulate and impede global relaxation as comparable input neighborhoods cannot be found. Their corrective step is therefore particularly useful at the beginning of the synthesis, but progressively loses its beneficial influence *(last row)*.



4.11 User Control over Synthesis. *(left)* Example of user-drawn brush strokes whose intensity grants effective control over the synthesized densities of elements. *(right)* By way of an image whose intensity values control our model's 1st-order statistics, we can also guide synthesis so that the output reveals the image through the local density of its elements in similar ways to stippling.



4.12 Handling of Spatial Regularity. This example illustrates the main limitation of the Strauss model. Since it only accounts to second order interactions, it cannot reproduce well strongly regular arrangements such as this reference whose elements are pasted onto on a rectangular lattice. More sophisticated models that uses higher order interactions could be investigated to push back this limitation.

rules – that force the output Delaunay triangulation to be unskewed.

The special case of *strongly regular* distributions is different, as it is not properly supported by our current Strauss model. While this shortcoming also exists in [BBT+06], specific care has been provided by [IMIM08] to handle such arrangements, but to the price of user intervention, such as the manual correction of the extracted Delaunay triangulation. In our case, this issue is related to the fact we only consider pair-wise interactions. We believe that considering statistics of interactions involving more than two elements could help us lift that limitation.

Automation vs. Versatility

By locally modifying its element density parameters, our model inherently proposes some intuitive handles for the user to design the synthesized arrangements. For instance, it is straightforward to make our output distributions follow a specified path, typically drawn by the user via a intensity-varying brush tool. This allows the same kind of expressiveness as the *spray* and *boundary tools* in [IMIM08]. However, our local element density control is novel.

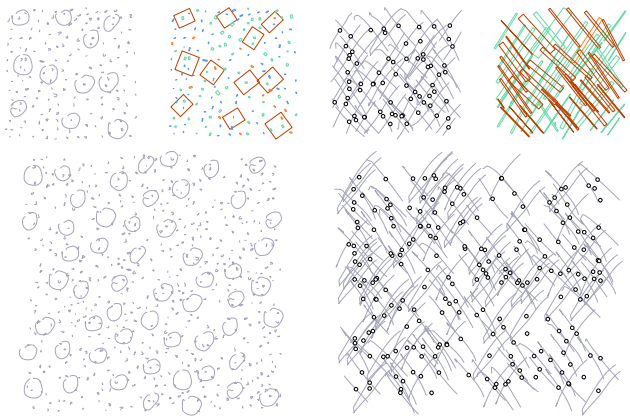
Further control on the elements' orientation, such as randomization or flow-guided harmonization, could also be added. Those would take the form of post-processing steps, however, since incorporating too many parameters would endanger the tractability of our statistical fitting. Considering other appearance features, like elements' size or color, could also lead to interesting results and is currently left to future work. Indeed, the focus of the present article is to increase the amount of automatically extractable information from a provided example and capture the interplay between the appearance and the spatial organization of a set of observed elements.

4.4.3 Limitations and Future Work

A group of perceptually similar elements can sometimes be over-categorized. This effect has yet no consequence on the synthesis step since it leads our model fitting to infer interactions between similar objects

stored in different appearance groups. These interactions are reproduced in the synthesized arrangement, but remain unnoticed –as in the reference arrangement– because of the perceptual similarity of the elements (Figure 4.13-*left*). Actually, over-categorization on that specific example involves very small elements and a closer inspection on those tiny shapes does indicate differences between them. Such dissimilarities are hardly visible without explicit zooming though.

Another limitation of our approach is the spatial representation of the elements by their respective centroid. When elements are strongly elongated, this representation is not adapted and is misleading for the model. This is visible in Figure 4.13-*right* showing a hatching arrangement. The centroid distribution is well reproduced. Yet, interactions between centroids is not representative of interactions between the actual elements. This drawback could be circumvented by using the Hausdorff distance between bounding boxes instead of the Euclidean distance as the parameter for the interaction functions. Besides, the synthesis of such hatching patterns implies to answer another ill-posed problem. Would the artist expect the system to cover the output window with strokes picked from the example, or by directly adapt their length?



4.13.Limitations. *(left)* Visually similar elements may sometimes end up in more categories than necessary, such as the small elements displayed here which fall into three distinct categories instead of one *(upper right)*. We call that phenomenon *over-categorization*. It has close to no impact on the visual quality of our synthesis results, as the system then strives to reproduce unnoticeable interactions between categories containing resembling elements. *(right)* Since elongated elements are not well represented by their sole centroids *(red dots)*, our interaction model based on point-wise distances does not accurately account for the actual interactions between elements. It should noted that the distribution of the centroids is yet preserved.

4.5 Conclusion

We have presented an example-based method to synthesize arrangements of vector elements that combines the appearance-guided categorization of the elements and the statistical modeling of the spatial interactions occurring within and between appearance categories. The categorization step is based on several perceptual principles and it could be profitably exploited in other methods such as the procedural approach of [IMIM08]. To the best of our knowledge, our statistical modeling for 2D element arrangements is novel. We believe that multi-type point processes –and marked point processes in general– constitute interesting and flexible theoretical tools that could be further investigated.

Our method currently yields satisfactory results (*cf.* Figure 4.8), both in terms of analysis and synthesis. Julesz' line descriptive features prove efficient for our element appearance categorization and our automatic clustering constitutes a great improvement with respect to previous work. Similarly, the multi-type point process we rely on for the statistical description and generation of the elements' spatial behavior also commands a marginal advantage thanks to its global nature alone, which contrasts with existing techniques [BBT+06, BBMT06, IMIM08] based on the concept of neighborhood obtained through triangulation. Seemingly simplistic (*cf.* Figure 4.5), our model's interactions nevertheless successfully account for previously-overlooked spatial relations such as overlap, extraction or repulsion. While not as user-centric as Ijiri *et al.*'s approach, our method also proposes original synthesis control, *e.g.* "paint-by-number" synthesis (*cf.* Figure 4.11*(left)*) and "element stippling" (*cf.* Figure4.11*(right)*). A careful examination of the output's elements shows that, although the first-order statistics are voluntarily ignored, second-order interactions keep on being observed. Still, our current – and highly experimental – implementation hinders us from providing users with an interactive paint interface comparable to Ijiri's. But more than a mere technical issue, theoretical work would also be needed to make our MCMC sampling synthesis scheme fully compatible with a real-time application.

4.6 Future Work and Perspectives

Notwithstanding its merits, our method naturally has its share of shortcomings, much room being left for improvement as indicated in the remaining of this Section.

4.6.1 Improving the Arrangement Analysis

Improving the robustness of our arrangement statistical analysis represents a tricky issue because of the intrinsic conflict between our end application purposes and the mathematical tools it relies upon. Ideally, we want users to start drawing the example arrangement, and our method automatically bring together visually-close elements. But for being of actual use for artists, our analysis has to handle limited inputs. Alas, this practical constraint makes it fall prey to a common hurdle in statistical modeling: the *curse of dimensionality*.

The curse of dimensionality directly binds a statistical model's expressive power to the number of its observed realizations. The fitting of a statistical model fitting eventually boils down to finding a hidden functional that wraps model's parameter space according to the training observations. As the number of parameters increases, this space grows at an exponential rate, and therefore requires an as increasing number of observations scattered across it to yield a properly-constrained mapping. The curse of dimensionality plagues all steps of our approach, from the approximation of the appearance features' probability density functions by histograms to the choice of our point process model, and forces us to always find a careful compromise between robustness and usability.

For our classification into appearance categories, we could envision to relax our automation objective by allowing users to correct misclassified elements through a point-and-click graphical interface without endangering the overall ease-of-use of our method. User intervention should then be confined to the attachment of additional importance to ill-grouped elements, or maybe for explicit correction.

Lifting the restrictive influence of the curse of dimensionality over our spatial model is more involved though. As alluded in Section 4.3.1, a spatial pattern is mathematically defined through the derivative of its probability density function with respect to a entirely random Poisson distribution, called its Radon-Nikodym derivative. The formula of our model's such derivative (*cf.* Equation 4.2) shows that its number of parameters is directly linked to the number of appearance categories, the complexity of the interaction functional, and the interaction order. In order to avoid the curse of dimensionality and guarantee the solvability of its fitting, our model could only embed pair-wise statistics. The trade of our current two-step interaction function (*cf.* Figure 4.5) for a linear function would considerably complicate the fitting process, and considering statistics involving more than two elements would require much larger input arrangements. This last restriction is especially aggravating as it prevents us from properly capturing and reproducing regular arrangements. Finding ways to incorporate higher-order relations in our model may enable us to do so without imposing priors on the element layout – such as Guo's or Dischler's assumption that an element always has four neighbors disposed in each quadrant [GZW01, DMLG02] – or enforcing regularity via post-optimization – such as Guo's simulated annealing or Ijiri's relaxation [GZW01, IMIM08].

A final shortcoming of our method stems from our model's weakness for expressing arrangements whose elements are disposed in a pronouncedly clustered fashion. While it can capture non-uniformity to some extent, re-synthesis performs poorly for such cases. Our maximum likelihood estimation of its trend distances $(r_{m,m'})$ also loses meaningfulness as the Ripley's function would start as monotonically increasing while we look for its first local minimum. This begs the question of the multi-scale analysis of our inputs. Provided the fact the input remains stationary, building a hierarchy of nested element clusters in a preliminary spatial clustering step could allow us to handle such cases by then applying our technique in a top-down manner. Alas, the higher we would go up in the hierarchy, the fewer elements we would have at our disposal for analysis. Considering back greedy, local methods could then be the only alternative left, leaving our global statistical approach confined to the finest levels of the hierarchy.

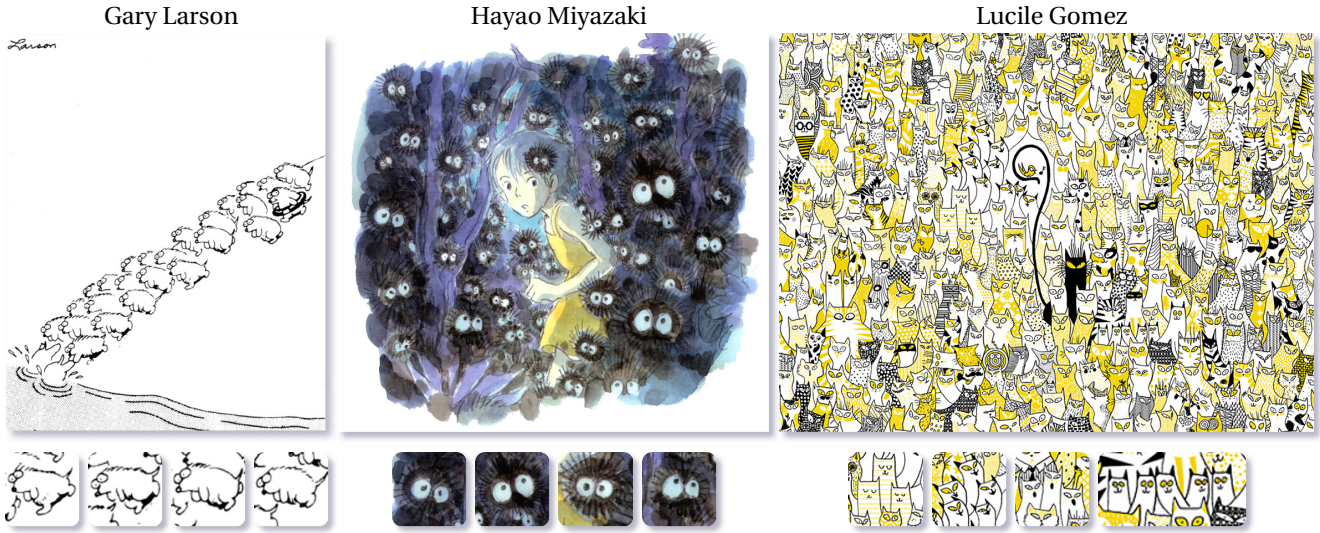
4.14 Natural Variations in a Hand-Made Drawing. When manually drawing a shape or a character several times, no matter its visual simplicity, some small differences always manage to slip between its different instances as shown by the close-ups from this Japanese advertisement. Emphasized by the artist's uncluttered style, such natural variations confer the hand-drawn "feel" to this image. This example is especially inspiring as its creator's style, privileging simple and apparent lines, makes the repeated elements comparable to the ones handled by our method. Source: Courtesy of William Baxter.

4.6.2 Improving Synthesis

While devising more elaborate stopping criteria involving the evolution of the model's energy during its simulation, it seems that the visual quality of the synthesized arrangement actually lies more in the element duplication procedure than in the synthesis stage itself. Our direct copy-paste approach indeed creates instantly noticeable repetitions throughout the output. In hand-made examples, such a verbatim duplication never occurs even for the simplest of drawing primitives. Naturally, the more complex the elements get, the looser the exactness of the replicates becomes, which confers a handcrafted and artistic flair to the drawing (*cf.* Figures 4.14 and 4.15). The introduction of natural variations between our pasted elements thus appears as necessary for hiding the computational origin of our output vector textures and improving their visual quality.

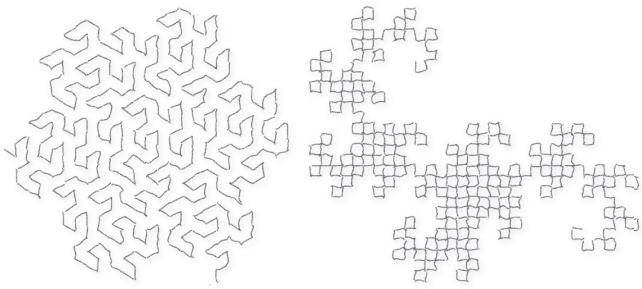
Similar concerns already arose in the texture re-synthesis stage presented in the previous chapter where small variations in the pasted patterns' appearance were simulated by sampling the principal axis of their color distribution in the input. Now, in place of sets of color pixels, we handle vector elements consisting of path-following splines.

Altering the Lines' Appearance By their definition, strong connections can be found between our vector elements and Strothotte's concept of *higher-order curves* [SS02], and distinction between the *path* and *style* of a rendered stroke. While its path describes the curve's geometric position, its final appearance is dictated by its style, represented by Strothotte as a functional creating small geometric deviations to the path for mimicking the natural shaking of the hand, and sequences of width and brightness values describing the evolution of the pen's pressure and saturation. By sampling such stylistic parameters, we could make each instance of our output arrangements appear pre-attentively unique. While research for successfully transferring line's style by example exist in the Expressive Rendering literature [KMM+02, HOCS02, JEGPO02, FTP03], a simpler solution, better suited to cases where stylistic analysis is limited by small inputs, would be to follow the recent steps of AlMeraj *et al.* [AWI+09]. Having built a collection of line styles from real-life scanned drawings, their "out-of-the-box" approach lends itself more easily to our application case. Indeed, contrary to aforementioned techniques that require training data to capture the artists' styles through statistics or transfer, their system's only mandatory inputs are the control points describing the curve's path (*cf.* Figure 4.16).

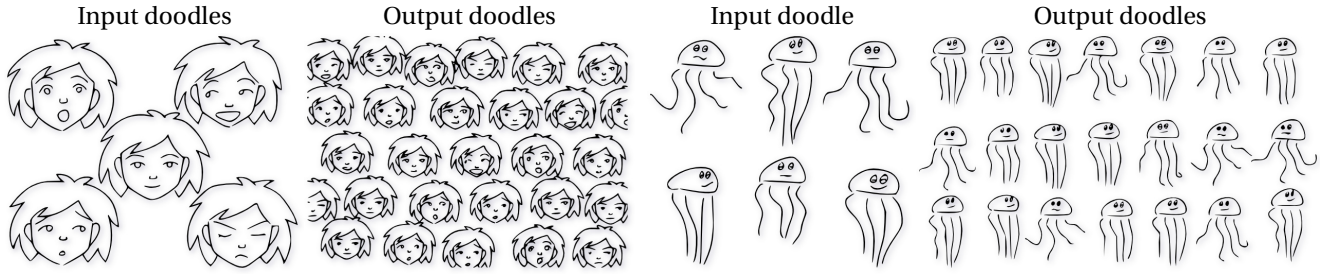


4.15 Approximate "Instancing" in Art. Though "instancing" is an inappropriate term for describing art, repetition plays a critical role in the composition of all three presented drawings. Whether for comical or simply visual purposes, repetition of shapes is used to make any dissimilar element naturally stand out from the myriad of duplicates. The more exact the copying, the slighter the deviation needs to be for immediate localization of the "intruder". In Gomez's drawing however, the most and only distinguishing feature of her mascot character *Méphistofélix* is the color of his black fur. This specific work is especially interesting as repetition is remarkably aesthetically-driven. Therefore "instances", although they still do not get the spot light, cannot be considered as simple filler, unlike in Larson's drawing or Miyazaki's watercolor where the *susuwatari* spirits form some kind of dusty and indistinct background.

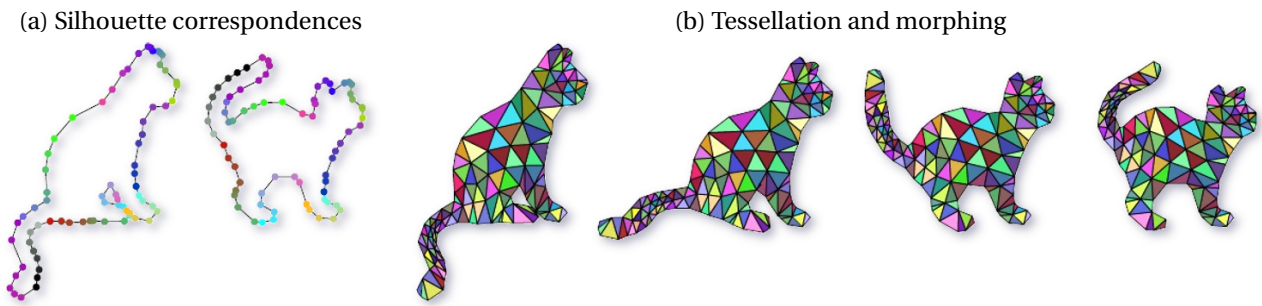
4.16. Conveying Natural Variations Through Line Style [AWI+09]. In the case of line drawings, dissociating a stroke's path from its style is not clear, and introducing variations directly at the line rendering level may prove useful to hide the obvious repetitions of our outputs. AlMeraj *et al.* propose to mimic hand-drawn lines through a supervised learning technique exploiting appearance statistics from a hand-drawn line database. For the simulation of the small stroke orientation glitches often observed in real-line drawings, they use Flash and Hogan's arm movement model to create natural-looking, jerk-minimizing paths.



Interpolating Elements' Shapes As our analysis brings together elements of a comparable appearance together, we could try to "interpolate" between members of a same appearance category for creating new elements in similar way to Baxter's *Latent doodle space* [BA06]. Such work stems from Alexa's *morphing space* that he conceived as an alternative to key framing for animation [AM99]. Poses appear as points in this high-dimensional space, while animations correspond to trajectories running through it. Less tedious than key framing and more flexible than the fitting of a shape deformable model, morphing-based approaches achieve a good compromise between expedience and artistic freedom, and comply to the new needs of animators since the advent of vector-based animations. But again, this ease-of-use comes at the price of overcoming issues such as the handling of the exponential growth of the morphing space as key poses are embedded into it, and the interpolation between an arbitrary number of complex shapes. In his line-drawing centric *Latent doodle space*, Baxter directly uses linear interpolation between pairs of matched strokes, creating unnerving wrinkles or loops in the process [BA06] (*cf.* Figure 4.17). Conversely, approaches involving image-based input poses instead resort to rigid-shape interpolation schemes [ACOL00, BBA08, BBA09a]. Most of such schemes are unfortunately confined to the case of 2-way interpolation, whereas the circulation through the morphing space commonly results in the association of a higher number of poses. The tessellation of the morphing space enables N-way linear interpolation, but breaks when it comes to as-rigid-as-possible interpolation as the symmetry property is not ensured. The lack of symmetry also aggravates the order-dependence of

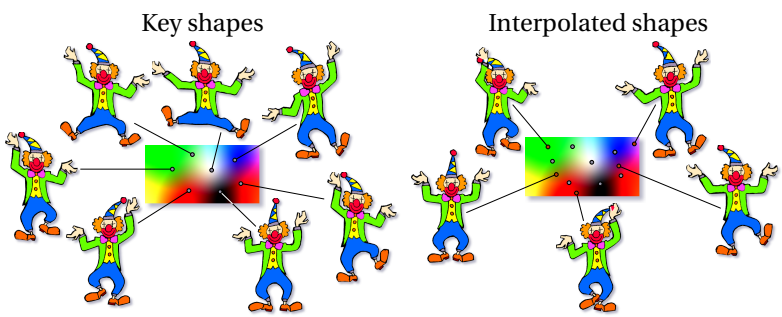


4.17 Baxter *et al.*'s Latent Doodle Space [BA06]. Baxter *et al.* build a high-dimensional *doodle space* inside of which drawings appear as points. By "exploring" it, users introduce natural variations to their drawings. Their technique is divided into two stages: a stroke matching procedure between the input examples, and the construction of the latent space. The first step is cast as a constrained clustering problem considering the strokes' positions, orientation and connectivity. The second stage mostly requires robust dimensionality reduction techniques to provide users with an intuitive exploration interface and satisfactory doodle interpolation.



4.18 Baxter *et al.*'s Shape Embedding for 2-Way Interpolation [BBA09a]. Baxter trades geometric matching for image-space morphing to achieve efficient and robust interpolation between the silhouettes of two shapes. After extracting feature points by considering the scale-space behavior of the local curvature, they find a set of globally consistent correspondences via the spectral analysis of the resulting assignment matrix *(a)*. Next, they compute a compatible tessellation between the two shapes by considering these matches *(b)*. Morphing is finally guided by the end shapes' directly comparable skeletons. The achieved degree of automation is impressive, the user only providing a single pair of correspondences to disambiguate the parametrization of the silhouettes.

results obtained by recursive 2-way blends [XZWB05]. The state-of-the-art solution is currently Baxter *et al.*'s extension of their *compatible shape embedding* for animation [BBA09a] to the case of N-way rigid interpolation [BBA09b](*cf.* Figures 4.18 and 4.19 respectively). Considering back our line-based inputs, Baxter *et al.*'s latent doodle space [BA06] appears as solid starting point to build upon. Along with the need of a better stroke interpolation, their strict requirement of an equal number of strokes between input elements is an hindrance for the method's useability that needs to be lifted. Similarly, their stroke matching step involves a metric which accounts for the lines' orientation and connectivity, but disregards their actual shape and is not invariant to simple geometric transforms. Enriching it with intrinsic information such as normalized curvature-based signatures could therefore be salutary for extending it to less monitored frameworks.



4.19. Baxter *et al.*'s N-Way Morphing [BBA09b]. Baxter *et al.* had the goal of enabling intuitive animation control clear in their mind when extending their animation-friendly shape interpolation [BBA09a] to the more useful case of rigid shape interpolation involving an arbitrary number of key poses. They greatly facilitate the user's experience by enabling the free roaming of their apparently 2d pose space where point trajectories correspond to animations.

Part II

Perceptually-Accurate Image Desaturation and Enhancement

The second part of my thesis leaves the texturing field and embraces the fascinating domain of color perception, but transposes the same idea of prior-free information extraction for synthesis in the context of image enhancement.

Research detailed herein was conducted in pair with Kaleigh Smith at the end of her internship in the ARTIS team during the first quarter of 2008, and Pr. Joëlle Thollot. At this time, Kaleigh was a PhD candidate at the Max Planck Institut Informatik (Saarbrücken, Germany) under the guidance of Pr. Karol Myszkowski. Her research interests resolved around visual perception and its interplay with artistic techniques. She graduated in December 2008 after defending her thesis *Contours and Contrast* which explores the perceptual impact of unsharp masking and its possible applications to various Computer Graphics applications such as tone mapping, grey-scale conversion and 3d rendering enhancement. Her gait as a Computer Graphics researcher is characterized by a strong artistic sensitivity, always drawing inspiration and empirical validation from real-life media and art.

Following a first attempt at the generalized application of unsharp masking for the enhancement of 2d imagery, our collaboration changed direction to focus on the common problem of color to grey-scale conversion. The subtending idea remained though and consisted in exploiting the widely-known Cornsweet illusions and the recourse to the unsharp masking filter to easily and efficiently introduce them. Deeply concerned by perceptual meaningfulness of our conversion process, our approach departed from existing techniques which often neglect perceptual accuracy in aid of color contrast preservation. Our work had the chance to be part of the proceedings of EUROGRAPHICS'08, and an independent study conducted by Martin Čadík attested the quality of our results [Čad08a].

Context and Motivations

Even with nowadays' technology, the need of grey-scale imagery is as strong as before. Either for printing, diminishing publishing costs, or simply driven by an artistic whim, the conversion of color images to grey-scale outputs appears among the most basic and commonly-used image processing operations.

But the profitableness of grey-scale conversion goes beyond these application cases: most Computer Vision algorithms for instance still handle their inputs, static images or video frames, only as scalar fields defined over a 2d spatial domain and disregard the information conveyed by their color distribution[6].

Similarly, various expressive rendering techniques consist in the computational emulation of real-life artistic techniques, and aim at the representation of *tonal values* by way of a spatially-varying primitive distributions. Noteworthy examples include the hatching and stippling rendering techniques which have their densities of lines or points reflect the spatial variations of tone (*cf.* Figure 4.20). All these aforementioned methods could greatly benefit from an improved grey-scale conversion that controls the level impoverishment suffered from the colored input.

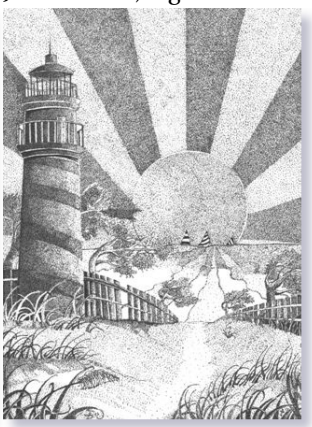
Given an color input image, finding its most suited achromatic version first requires the proper identification of the real objectives fueling the conversion. From our experience, there are two fundamental goals: the preservation of the *entirety* of the image's original content, or the respect of the *perceptual sensation* it elicits in terms of lightness. A grey-scale conversion strictly fulfilling these two conflicting ambitions seems alas impossible.

By the condensing of a three-channeled image to a single-channeled one, information is naturally lost. To make up for it, the lost data must be characterized and detected for applying the relevant processing. Alas, the more missing information need to be brought back to the output, the more artificial distortions need to be bludgeoned into it. Algorithmic complexity left aside, such an approach would indeed meet the first objective.

[6] The usefulness of grey-scale conversions for uses other than direct display or printing is factually illustrated by the resort to Grundland's *Decolorize* algorithm [GD07] by image-based rendering [ZBA+07] and object recognition techniques [ZDDM06].

Robert Doisneau, *La banlieue*.

Joe Pacheco, Lighthouse.



Norman Koren

4.20 Black-And-White in Art. Presented on the left are two examples of extremely different artworks taking advantage of the apparent limitation imposed by their monochromatic medium to create evocative settings. Shown on the right is a benchmark image used to evaluate the quality of achromatic renditions obtained by digital processing for display and printing. It illustrates the intensive amount of care grey-scale imagery, especially photography, receive from artists. With the decline of analog photography in favor of digital photography, the need for image processing algorithms for converting color images to grey-scale is stronger than ever.

But thinking of the output grey distribution as a freely editable waste ground leads to hazardous solutions that clearly contradict the second objective.

Before anything else, we claim a satisfactory grey-scale conversion method is to ensure believable results, not only in terms of overall naturalness, but also when directly confronted to the color input images they originate from. As intuitive as it may sound, this concern for perceptual accuracy is barely mentioned by most previous techniques, and its inclusion to our approach constituted one of its contributions.

The perceptual plausibility requirement appears paramount to us as it governs the preference of achromatic imagery in certain cases. Far from being imposed by technical or budgetary constraints, black-and-white artistic photography attests a strong aesthetical appeal for this specific medium. Gifted with a unique and abstract beauty, monochrome photographs draw their success from their lesser ties to reality. While a color photograph would bluntly reveal all the details of a scene, its achromatic rendition lends itself more easily to the plays of the viewer's imagination (*cf.* Figure 4.20).

Outline

During the course of our demonstration, special emphasis is given to the perceptual aspects of our choices, mostly resolving around our perception of individual brightness, the effects of color and the influence of contours on sensed contrasts. The remaining of the present part is organized as follows. The presentation of most color specification systems is to be found in Appendix A whose objective is to provide non-expert readers with all the necessary notions to fully grasp the perceptual aspects of our work. Direct concurrent work on the specific matter of grey-scale conversion are detailed in Section 5, a couple of these having been published posterior our method. Follows the detail of our approach in Section 6.

CHAPTER 5

State of the Art in Grey-Scale Conversion

Choosing a suitable color specification is intimately linked to our problem as it directly affects the representation of colors and the information we can draw from them. Every single specification presented in Appendix A describes a new color space whose characteristics and strengths dramatically differ from one another. Key features such as additivity, perceptual uniformness or aesthetical uniformity are crucial notions that are to be taken into account. This choice also conditions the actions the method can take. Color spaces hence constitute the theoretical ground upon which any technique involving color images is built, and as such should be incorporated in the early stages of its conception. We now review the literature dedicated to the issue of content-aware grey-scale conversion and show the conflicting nature between its two possibly pursued objectives: the strict preservation of all chromatic information by the produced grey-scale images, or their achromatic accurate portrayal of the input image's color distribution.

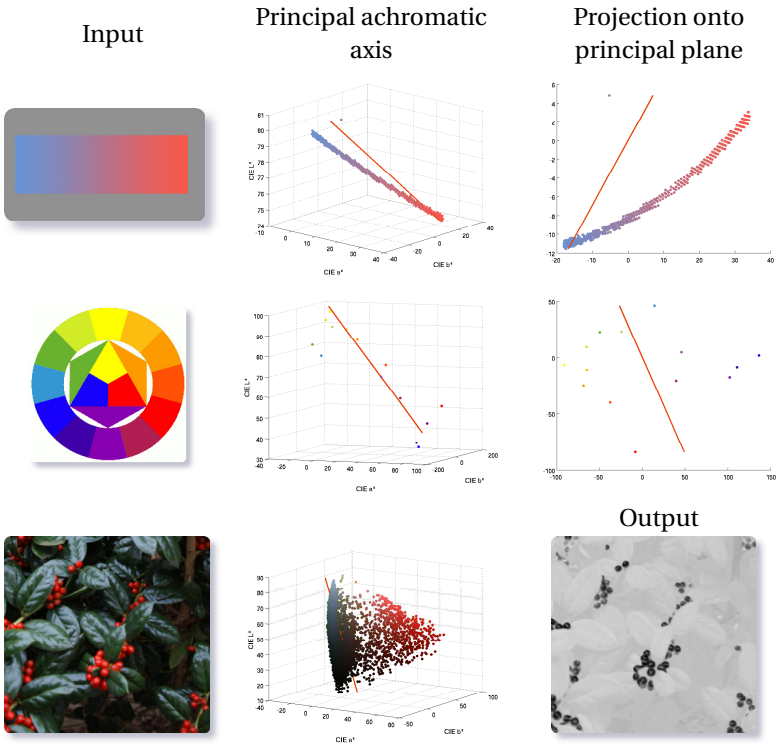
5.1 Mapping Colors to Grey Values

From a data analysis standpoint, color to grey-scale conversion comes down to a dimensionality reduction problem where *three-dimensional* color triples must be mapped to *scalar* grey values. More than the mathematical tools involved in their computation, a first classifying criterion between such mappings lies in their spatial behavior, whether they act globally or locally.

Global mappings guarantee the consistency of the returned results across the entirety of the image plane, and assign to features sharing a same color the very same grey-scale intensity regardless their relative distance throughout the input. Intensive algorithmic and computational care is usually required to fulfil this consistency objective and also often comes at the price of sacrificing part of the finest detail resolution. Conversely, local mappings adapt their output values to the local color distributions. They can therefore better depict small features and produce images with an overall sharper look. The weakness of such mappings lies in the grey values inhomogeneities they can introduce in originally flat-colored regions, and in their frequently surjective nature as their grey value assignments depend on the colors' surrounds.

Effortless grey-scale conversions often disregard the input image's chrominance and consist in global mappings that directly replace the input's colors by the average of their red, green and blue channels, or their luminance component. In such situations, edges between equiluminant directly adjacent colors naturally and irrevocably disappear and cannot be recovered even by gamma post-correction. The disappearance of such visual cues is especially dreadful as the collisions between equiluminant colors are commonly found in natural scenes and artistic imaginary, notably for the illusory motions they induce.

Bala and Braun's early take on grey-scale conversion attempts to *explicitly* alter their results' grey values so that they still reflect the contrasts that a direct mapping to luminance would overlook [BB03]. Input colors are projected onto an achromatic axis following a global non-linear mapping that incorporates the effect of chroma on lightness as dictated by Fairchild's simplest lightness predictor L_1^{**}. The obtained grey values are then re-spaced – either equally or according to the color distances in the CIE $L^*a^*b^*$ color space – while respecting their overall lightness order. Alas, this last step meant to facilitate discrimination of adjacent colors necessitates the explicit extraction of all distinct colors presented in the image, hence confining the use of their conversion to inputs with a discrete color gamut such as vector graphics. As attested by the results present in their report, their approach is dedicated to the case of business graphics and cannot be directly applied to more complex images as it would break in the presence of color gradients.



5.1 Shortcomings of PCA. In addition to its limited flexibility in terms of statistical modeling, PCA suffers from severe issues such as its sensitivity to outliers *(top row)*, its global linear nature that cannot adapt to spread color distributions *(center row)*, and the generation of excessive contrasts when carried out in the context of grey-scale conversion *(bottom row)*.

Given the 3d color distribution of a particular image, Principal Component Analysis (PCA) can unveil a more appropriate luminance axis. The computed principal component is the linear axis that maximizes the variance of the color projections in the least-squares sense. Alas, grey-scale conversion approaches relying on it are directly plagued by the shortcomings of this specific dimensionality reduction technique (*cf.* Figure 5.1). Its most severe limitations are its increased sensitivity to outliers, and even though it could be thought of as a strength, its lack of priors regarding the observations' probability distribution function. As a direct consequence, it can only capture simplistic linear correlations between colors, which is quite a restricting solution in terms of statistical color modeling. Lastly, the relative spacings between colors once projected over the principal achromatic axis do not have any relevant meaning in terms of brightness perception, and directly scaling the color's projections to spread them across the luminance range yields unpleasant results with highly distorted contrasts.

A better adaptation of the conversion mapping to the input's color distribution is therefore necessary to expect complex chromatic contents to stand its application. Following this rationale, Rasche *et al.* find a linear global mapping that ensures the equality of the normalized contrasts between the input colors and their assigned grey counterparts. Privileging discrimination above all other concerns, Rasche only accounts for the perceptual distances between colors, and disregards their spatial location, pixels of a same color thus ending up with the same intensity value regardless of their neighbors' color distribution. Color differences are computed using the computationally simple albeit perceptually relevant CIE $L^* a^* b^*$ Euclidean distance. The observance of the input color contrasts by the sought grey differences is embedded in a cost function similar to multidimensional scaling. They employ a conjugate gradient method to compute the linear transform in CIE $L^* a^* b^*$ that would best preserve the observed contrasts, and initialize their iterative solver with the direct luminance mapping. For alleviating the computational load, Rasche *et al.* suggest to perform quantization of the input's palette color assuming it would not jeopardize the results' quality insofar the remaining colors are weighted according to their relative frequency in the image. The primary goal of Rasche's method is the correction of imagery for visually impaired persons suffering from cone cell disorders such as trichromatopsia or dichromatopsia. Henceforth its main focus lies more in the differentiation between colors than their individual identification. This objective of his is clearly reflected by his results which while excelling in global discrimination, exhibit a highly distorted range and often unexpected contrasts. Rasche's method also reveals

the shortcoming of linear mappings as a global linear map cannot correctly unfold a color distribution consisting of multiple modes of contrasting gradients. Besides, if the input has large portions of black and white colored areas, the linear map cannot deviate from the standard luminance mapping and fails to capture color contrasts.

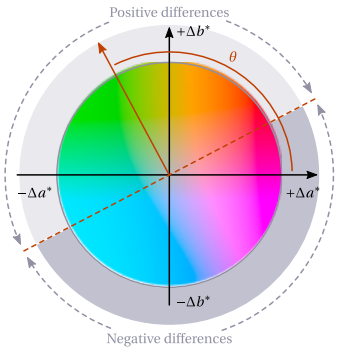
Though aware of the intrinsic limitations of global linear mappings for handling intricate color distributions, Rasche finds in them an alternative to more elaborate dimensionality reduction techniques that he considers as dauntingly expensive and complicated. It is true that analysis techniques such as local linear embedding or Laplacian eigenmap embedding come down to eigenvalue problems that require the spectral analysis of large sparse matrices encoding the data set topology. However, in very recent work, Cui *et al.* manage to successfully apply such a technique to the case of grey-scale conversion [CHRW09]. Thanks to the non-manifold ISOMAP learning technique, they perform complex dimensionality reduction by applying a linear mapping on the image's non-linear embedding. The input image is represented by a graph in CIE $L^* a^* b^*$ space obtained by linking the nearest colors together. The geodesic distances between its nodes are estimated via Dijkstra's shortest-path finding algorithm and stored in a cost matrix. Comparatively to Rasche *et al.*'s color quantization, they suggest to compute exact distances only between *landmark points* which consist of a smaller portion of the input data that spans the entirety of its distribution. Once computed, the values of the cost matrix are mapped to a non-linear scale for controlling contrasts, and the final 1d color space is obtained by the spectral analysis of the matrix.

5.2 Grey-Scale Conversion by the Integration of the Color Contrasts

A possible alternative for preserving the image's chromatic content consists in reconstructing the output from the set of desired contrasts. Such gradient-based techniques implicitly find non-linear local mappings that ensure the observance by the result of all the input's contrast constraints. They differ from one another in the mathematical definition of the color contrasts and the computational ways used to create the final image out of them. They must all confront themselves to the ill-posed issue of the determination of the sign of such contrasts, as the lack of absolute ordering between multivariate data such as colors complicates the estimation of their gradient. Notably, the issue of finding a consistent contrast sign assignment is decisive as spatially inconsistent or carelessly chosen gradient polarities can endanger the output's visual plausibility or very existence.

Bala and Eschbar propose another color to grey-scale transform where chrominance edges supersede Bala's previous lightness reordering step [BE04]. Detected by a high-pass filter run over the image's chroma C^*_{ab} values, edges are adaptively added to the luminance values and locally weighted so as to account for already existing luminance edges and prevent overshooting. While their magnitude comes from the chroma high-frequencies, the sign of the edges is controlled by the luminance gradients that may be ill-defined or unstable at locations of equiluminant chrominance edges. From the results provided in their article and from the ones obtained using Čadík's implementation, the enforced contrasts assume the form of clearly noticeable edges which are somewhat distracting as they are nowhere to be found so markedly in the input.

In her seminal paper, Amy Gooch's *Color2Grey* suggests a more successful conversion method and pioneers the formulation of the generation of the achromatic image from imposed color contrasts as an optimization problem [GOTG05]. Drawing inspiration from the use of Poisson solvers for gradient-domain high dynamic range compression, Gooch gathers the contrasts between pairs of surrounding pixels, and combines them in an objective function favoring the preservation by the grey values of the color contrasts. The optimal image is obtained using an iterative conjugate gradient solver. The



5.2 Gooch's Sign Assignment [GOTG05].

expression of the color *scalar* differences is one of Gooch's strongest contributions. It involves the magnitude of the CIE chroma C^*_{ab} difference if it exceeds a certainty threshold, or the CIE lightness L^* difference otherwise. Instead of arbitrarily determining the contrasts' polarity, Gooch enables users to specify a hue angle θ used to split the CIE $L^*a^*b^*$ chrominance plane in two oppositely-polarized halves (*cf.* Figure 5.2). Only the color difference *vectors* $\Delta C = (\Delta a^*, \Delta b^*)$ pointing in the same direction as the user-provided vector are attached a positive sign. This hue angle parameter has a dramatic impact over the results' visual aspect, making them oscillate between satisfactory to completely unnatural by a simple change of its value. Alas, the computational complexity of Gooch's method ranges from quadratic to quartic in terms of pixels – as it theoretically considers the contrasts of all pixel pairs – and therefore hinders the trial-and-error needed to assess the influence of this parameter over a given image. The performances of Gooch's original implementation unfortunately restricts its use to small inputs, even on nowadays' machines. Mantiuk *et al.*'s multi-scale extension accelerates the evaluation of long-range contrasts by considering the coarser levels of the image's pyramidal decomposition [MMS06]. Likewise, Ruzon *et al.* study Gooch's energy function and solve it by using look-up tables for the storage and interpolation of the most frequently evaluated quantities [RCWN06]. They also resort to color clustering in the CIE $L^*a^*b^*$ color space for trading Gooch's initial summation over the differences between pixel pairs for the summation over the differences between pixels and the input's dominant colors. More than implementation tricks, all these aforementioned improvements turned Gooch's technique into a truly interactive method.

In comparable work, Kuhn *et al.* perform constrained optimization on the luminance values of the quantized image using a mass-spring system [KOF08], where the constraints between colors control the amplitude of the movements the particles can perform. Each quantized color is associated with a particle which moves along the lightness axis and is subject to the forces from the other particles proportionally to their CIE $L^*a^*b^*$ distance. Once reached, the equilibrium state of the system describes the final grey-scale image which is obtained through interpolation between the quantized colors' established brightness. Kuhn empirically guarantees the perceptual relevance of his results by enforcing the closeness of its particle colors' intensities to their original lightness values by adjusting the rest length of their spring. While performing good results, Kuhn's technique requires the extraction of the input's palette which may fail to capture the complete chromatic content of the image and introduce artifacts during interpolation.

Leaving the optimization framework, Neumann *et al.*'s gradient-based method marks its difference by striving for a better perceptual accuracy [NCN07], contrasting sharply with the liberties taken by Gooch's user centric and Kuhn's physics-based approaches. They express their constraining differences in Nemcsics' Coloroid space and count on its considerable experimental background to guarantee an improved perceptual handling. In additional experiments, they study the effect at a given spatial frequency of the Coloroid hue and saturation on the perceived equivalent grey, and combine their findings with the classic luminance difference to yield their final color contrast formula. They use it to compute the image's gradient field that they further refine using an iterative greedy inconsistency correction procedure. Their achromatic image is obtained by the direct 2d integration of the resulting gradient field. By emphasizing the need of perceptual accuracy, Neumann *et al.*'s philosophy is indisputably closer to ours, but their technique suffers from several issues. The debatable resort to the Coloroid space left aside, most of them arise from the unpredictability of their gradient correction especially in detailed regions of the image. Its greedy nature also lends itself to the propagation of gradient signs that strongly differ from our perception of the original color image's contrasts, hence spoiling the authors' claim to propose a perceptually accurate conversion framework.

Very recently, an analogous grey-scale conversion technique by contrast reintegration has been proposed by Drew *et al.* [DCFB09]. They compute chrominance edge-aware luminance derivatives, and process the resulting gradient field so that it may be integrated back to produce the final output. They use Socolinsky and Wolff's algorithm to consider all color channels' derivatives as a whole and extract for each pixel the direction for which its magnitude stands maximal. Similarly to Neumann *et al.* who ensure their field's integrability by forcing the sums of the gradients surrounding each pixel to be null, Drew *et al.* detect locally

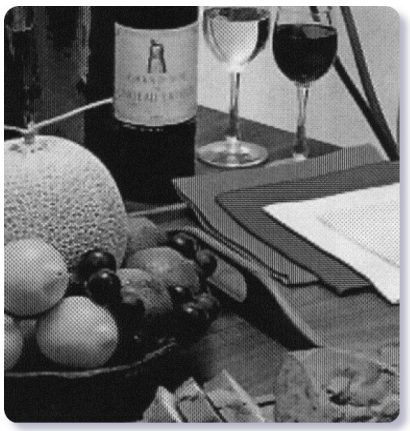
inconsistent gradients by considering the pixels whose gradient's curl[1] is non-null. Such pixels are iteratively corrected using an error-diffusion sign flipping procedure. While mathematically more elegant and robust that Neumann's, Drew *et al.*'s method abdicates perceptual relevance by relying on the p-Minkowski-norms to determine most of their gradient signs, and chooses p such as to minimize the field's global curl regardless of its perceptual meaningfulness.

Addressing most pitfalls such as algorithmic complexity and the lack of perceptual soundness, Grundland *et al.* conceive a global, continuous and piece-wise linear mapping for creating his grey-scale results [GD07]. Attaching special importance to the preservation of the original luminance range and ordering, Grundland's grey-scale conversion stands among the best-to-date techniques and reaches an efficient trade-off between discrimination and the respect of basic perceptual principles. RGB colors are converted to linear YPQ color space[2] and associated together by *Gaussian pairing* which draws the pair-defining displacement vectors from a univariate Gaussian distribution. Grundland's spatial sampling bestows his method with the advantages of both local and global techniques, respectively speed and long-range consistency. Lost contrasts between color pairs are defined as the normalized difference between the RGB Euclidean distance and the difference between their uncorrected luminance Y. They perform the predominant component analysis of these values to find the axis in the PQ chrominance plane maximizing the covariance between the observations' chromatic and luminance contrasts. The chromatic contributions obtained by projection over this axis are added to the pixel's luminance. While Grundland aims at perceptual correctness, the resort to the simplistic YPQ specification, which while reminiscent to Ruderman's linear $L\alpha\beta$ color spaces [RCcC98], is not perceptually validated, and compromises the method's accuracy to a certain extent, even when the authors claim the inclusion of more accurate color conversions would not bring a worthy improvement to their technique. Grundland *et al.*'s results are nevertheless quite pleasing to the eye while delivering the amount of distortion necessary for the lost contrasts to reappear most of the time. Yet, the severe data-dependency of his method hinders its straightforward extension to the case of videos or animated inputs.

5.3 Color-to-Texture Approaches

Original conversion methods using alternative visual cues to make up for the loss of chrominance information have also been proposed. Among them, techniques mapping colors to fine grey-scale textures represent a particularly interesting take on the issue. Naturally, their results are lacking in terms of perceptual faithfulness but are worth mentioning.

De Quieroz and Braun's conversion technique is especially interesting as they propose beside contrast preservation, to make their conversion process reversible and devise alongside their forward color-to-texture converter, the associate decoding algorithm that enables color recovery [dQB06]. For that aim, they decompose the image into a set of bandpass sub-bands using discrete wavelet transform and replace the image's high-pass horizontal and vertical sub-bands by the information carried by the chrominance planes of its YC_bC_r conversion. Performing the inverse transform produces the final grey-scale image where the spatial evolution of its high-frequency patterns reflects the original's color changes (*cf.* Figure 5.3). Among the virtues of their method, its creation of a continuous and smooth blend between textures constitutes a valuable improvement in comparison to earlier techniques that use a texture dic-



5.3 De Quieroz's Color-to-Texture [dQB06].

[1] Given a scalar function f defined over a 2d spatial domain $f : \mathbb{R}^2 \mapsto \mathbb{R}$, the curl operator of f is defined as $\text{curl}(f) = \frac{\partial^2 f}{\partial x \partial y} - \frac{\partial^2 f}{\partial y \partial x}$.

[2] In Grundland's YPQ color space, Y represents the achromatic luminance axis whereas P and Q approximate color-opponent channels

tionary and require the extraction of the input's palette [BHT00, KKLE06]. Unfortunately, while theoretically invertible – some details being irremediably lost during forward transform when the original sub-bands are overwritten –, their decoding method is not robust against changes in the grey-scale image: de-screening and warping caused by successive printing and scanning result in color shifts, whereas blurring causes desaturation.

5.4 Extension to Videos

Prior ours, no methods handled animated inputs. Often involving heavy optimization-based computations, many of them invoke color quantization that would create abrupt color changes and lightness ordering flips if unwisely applied to temporally-varying palettes. Similarly, mappings obtained by the statistical analysis of the input's color distribution also require specific care to remain relevant on videos.

Subsequently to our work, Kim *et al.* suggest a technique geared towards animated color content and find a non-linear global mapping whose parametric form is highly reminiscent to Fairchild's brightness predictor [KJDL09]. Optimal parameters are estimated on a per-pixel basis and are driven by chromatic edge information. Perceptual relevance is ensured to some extent by having this contrast information account for the natural lightness ordering imposed by Nayatani's estimator. Thanks to the simple parametric expression of their non-linear mapping, its analytical solution boils down to the inversion of a 9×9 matrix. Their handling of the temporal dimension is disarmingly simple as they directly incorporate temporally-based color discontinuities to their energy term. This straightforward extension to video is made possible by the local nature of the constraints behind their mapping. But while driven by local edge information, the fitted functional nevertheless aims at the globally consistent assignment of grey values. The robustness of their method to long video sequences exhibiting a importantly varying color palette may therefore not be guaranteed and the perceptual relevance of their ordering not hold. But in spite of these possible limitations, Kim's technique achieves a fairly good compromise between discrimination and perceptual accuracy, and lifts most of its previous work's limitations, including ours.

CHAPTER 6

Apparent Grey-Scale, A Fast Conversion for Images and Videos

Even if nowadays' color printers become widespread and mostly affordable, there is still room for monochromatic imagery. Either for printing cost concerns, limited display set-ups or artistic purposes, initially colored images or photographs often end up being presented as black-and-white pictures. By their richer representation, color images sure offer more freedom and elaborateness in terms of processing than their grey-scale counterparts, but the usefulness of the latter is not to be demonstrated. Grey-scale conversion is actually quite an involving task due to the high level of correlation between the image's RGB channels. But capturing correlations and possibly non-linear dependencies between a color's components is not the only hurdle. Since dealing with colors, we also need for the sake of our application to account for their brightness as perceived by an observer. Our conversion problem hence cannot be cast into a mere dimensionality reduction problem and our output brightness signals have to respect perceptual correctness.

In contrast with previous research, our claim is that for being genuinely successful, grey-scale conversion has to best embody our visual system's processing of the luminance emanating from the interaction of light with the observed scene. Whereas related work mostly focus on discriminability between the grey values assigned to different colors, we strive for the preservation of the colors' perceived brightness and the polarity of the gradients emerging from adjacent colors.

This duality of objectives – of global ordering and observance of local features – is reflected in the work-flow of our method:

- First, a global mapping taking advantage of recent findings in brightness perception is applied to all colors of the image;
- Second, a multi-scale enhancement is proposed in order to best account for the local chromatic gradients that could be lost in the previous step.

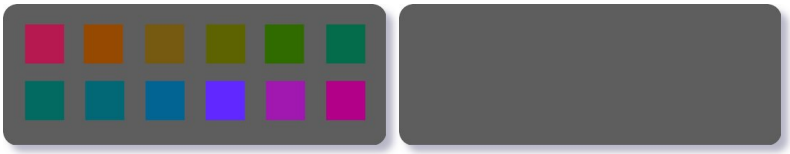
6.1 Influence of Chroma on our Perception of Brightness

The first stage of our technique directly deals with the very core of our grey-scale conversion problem and determines the major part of our final results' appearance. It notably tackles the non-trivial issue of finding a perceptually appropriate mapping from colors to grey values alongside with a relevant color ordering in terms of perceived brightness.

As shown in Appendix A, the modeling of perceived brightness is not new, as monochromatic brightness matching is one of the earliest experimental tool to quantify and specify visual responses. But devising relationships between radiometric measurements (*i.e.* the amount of light traversing a given area), and psychometric quantities (our visual sensation of it) comes down to map objective measures to subjective ones, and is as such a challenging issue. Visual perception modeling has known an incremental growth as new hypotheses were formulated and new effects observed. Among these, the Helmholtz-Kohlrausch effect deserves special attention as it is directly related to our grey-scale conversion problem.

6.1.1 Defining of the Helmholtz-Kohlrausch Effect

The best explanation of the Helmholtz-Kohlrausch entoptic effect is to illustrate it. Take a look at Figure 6.1. Its left part shows colored rectangles over an achromatic background, whereas its right part corresponds to



6.1 Illustration of the Helmholtz-Kohlrausch Effect. All colors share the same lightness and luminous reflectance ($L^* = 50$).

the pixel-wise mapping of these colors to their respective luminance[1]. We see that all colors share the same luminance as their surround. However, to many observers, the colored shapes still appear brighter than the background. This glowing impression – also referred to as *Farbenglut* [Koh35]– is the empirical definition of the Helmholtz-Kohlrausch effect and is also the evidence that luminance by itself does not constitute a trustable predictor for our perception of brightness.

A more formal definition has been proposed by Wyszecki who expresses the Helmholtz-Kohlrausch effect as the phenomenon that *"a chromatic stimulus, of the same luminance as the "white" stimulus will in general appear brighter than the reference stimulus"* [WS00].

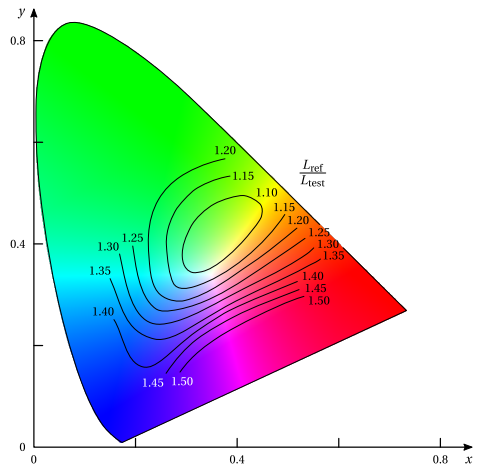
6.1.2 Unveiling the Helmholtz-Kohlrausch Effect

More thorough investigations have been proposed to isolate, formulate and analyze the Helmholtz-Kohlrausch phenomenon. Most of them involve the visual brightness matching heterochromatic stimuli. Such experiments differ from the ones scattered throughout Appendix A in several aspects: first, they involve *heterochromatic* stimuli contrary to the matching experiments that led to the establishment of the CIE standard observers; second, the require observers to match the bipartite stimuli's two-halves in terms of *brightness* only and not simply in terms of visual sensation. It turned out that this specific experimental set-up reveals some of the breakdowns of earlier models and unveiled the existence of the Helmholtz-Kohlrausch effect. Two seminal studies reached a same conclusion, that our perception of brightness of a visual stimulus is linked to its chromatic content.

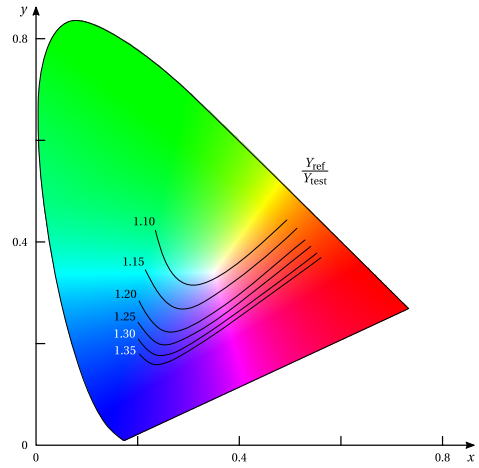
Sanders-Wyszecki 1964 [SW64] First experiments by Sanders and Wyszecki requested twenty observers to brightness match 95 heterochromatic test stimuli of distinct chromaticities against a common reference white stimulus. Stimuli were viewed through a 10° aperture encircled by a white surround field. While the colored test stimulus had its luminance kept constant ($L_{\text{test}} = 20cd/m^2$), observers had to tune the reference white's luminance L_{ref} until a satisfactory equality of brightness was attained between the two halves of the stimulus. L_{ref} therefore stands as an experimental estimate of the observers' perceived brightness sensation. Sanders and Wyszecki averaged the $\frac{L_{\text{ref}}}{L_{\text{test}}}$ ratios across subjects and observed that these ratios almost always exceeded 1. This implies that luminance tends to understate the perceived brightness of a colored stimulus, and hence fails at predicting the sensation of brightness in such cases. By drawing the loci of constant ratio across the xy diagram (*cf.* Figure 6.2), they deduce that the more saturated the visual stimulus is, the more pronounced the estimation error gets. Although variations between observers are important because of the subjectivity of the phenomenon, the consistency of the ratios across trials irrefutably confirms the effect of color on brightness.

Wyszecki 1967 [Wys67] The second set of experiments by Wyszecki focuses on the perception of colored surfaces and is thus complementary to his previous demonstration involving colored light-based stimuli. The increased number of test subjects – reaching out 76 individuals scattered throughout the United States and

[1] In the following, luminance L (the luminous flux per unit of area of light oriented in a given direction) is estimated from the image pixels' colors as follows. First, its estimation requires the expression of the red, green and blue color components in the device-independent, linear RGB space. We assume all our images are encoded in the $sRGB$ color space proposed by Hewlett-Packard and Microsoft in 1995 as a norm for color reproduction on mainstream display devices. Inverse gamma mapping must therefore be performed *prior* the computation of the luminance as the linear sum of RGB values. Since the $sRGB$ reference white corresponds to the D_{65} illuminant, the weighting percentages of the red, green and blue linear responses are 21.2656%, 71.5151% and 7.2186% respectively. These ratios are directly established from the tristimulus values Y of the D_{65} illuminant's primaries.



6.2 Demonstration of the Helmholtz-Kohlrausch Effect with Colored Lights. The irregular shape of the loci of constant ratios between perceived brightness and the luminance of the test stimuli shows the role of chromaticity over our perception of brightness.



6.3 Demonstration of the Helmholtz-Kohlrausch Effect with Colored Surfaces. Similarly, visual lightness matching of color surfaces demonstrates the unsuitability of the luminous reflectance for predicting the perceived brightness and accounting for the Helmholtz-Kohlrausch effect.

Canada – further increases the trustworthiness of his results. Observers were asked to consider 43 colored ceramic tiles and for each of them, pick a grey tile among a set of 10 achromatic tiles whose luminous reflectance Y_{ref} varied from 28 and 52.5 at roughly constant perceptual interval. The selected tile was to best match in terms of lightness the tested colored tile. By considering the averaged ratios $\frac{Y_{\text{ref}}}{Y_{\text{test}}}$ ratio, Wyszecki again came to analogous findings: at equal lightness, a colorful object appears brighter as a duller one. The loci over the xy diagram of the luminous reflectance ratios exhibit a strikingly similar shape to the ones associated with light stimuli (*cf.* Figure 6.3).

6.1.3 Explaining the Helmholtz-Kohlrausch Effect

Finding clarifications behind this phenomenon is of primordial importance as the Helmholtz-Kohlrausch effect indicates that established color measurements do not reflect the reality of our visual perception. Moreover, the conclusions from Sanders and Wyszecki's experiment indirectly question the relevance of the color matching framework upon which most CIE-ratified standards rely on. The non-linear and unpredictable evolution of the luminance ratios is likely to find its explanation in the unsuitability of the early stages of color modeling. At its very core, the conversion from eye-impinging radiant fluxes to perceived luminous fluxes finds its support in Abney's law.

Abney's law states that the luminous flux of several radiant fluxes viewed together can be approximated by the *summation* of their respective luminous fluxes. This assumption is essential for mathematically handling lights whose spectral radiant power follows an arbitrary distribution $L_e(\lambda)$. If it held strict, such a law would allow us to express the reference white's brightness estimate L_{ref} and the colored test stimulus's luminance L_{test} by way of an integration over the visible spectrum's wavelength:

$$L_{\text{ref}} = K_m \int_\lambda L_{e,\text{ref}}(\lambda)\, V^*(\lambda)\, d\lambda, \qquad L_{\text{test}} = K_m \int_\lambda L_{e,\text{test}}(\lambda)\, V^*(\lambda)\, d\lambda, \tag{6.1}$$

where $K_m = 683\, lm/W$ is the maximum luminous efficiency used to normalize the photopic spectral luminous efficiency $V^*(\lambda)$. However, the inequality between these two quantities after brightness match contradicts this perceived brightness modeling. Two main causes may be at the origin of the observed deviations:

- The empirically determined spectral luminous efficiency curves $V^*(\lambda)$ fail to yield faithful approximations of perceived brightness;
- The luminous flux additivity assumption as expressed by Abney's law does not hold.

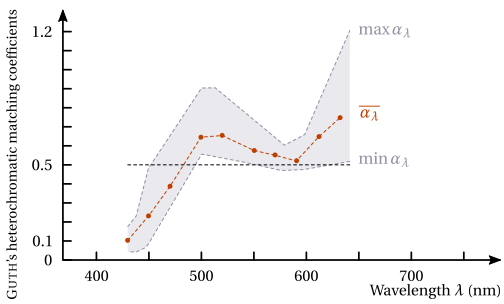
Judd proposed a revised version of the spectral luminous efficiency curve for photopic vision that differs from the original one notably over shorter wavelengths and corrects the former understatement of the brightness of blue lights [Jud51]. Still, even after replacing $V^*(\lambda)$ by Judd's revised function in Equations (6.1), the $\frac{L_{\text{ref}}}{L_{\text{test}}}$ ratios still significantly diverge from unity.

The domain of validity of Abney's law has been intensively investigated, especially in the general case of *heterochromatic* brightness matching where it appears to break. Boynton and Kaiser experimentally contradict the transitivity property of brightness matching implied by Abney's additivity [BK68], and demonstrate that the additive mixture of a "blue" and "yellow" stimuli, $[\mathsf{C}_{\text{blue}}]$ and $[\mathsf{C}_{\text{yellow}}]$ respectively, both brightness matched with a common "white" reference $[\mathsf{W}]$, always appears *less bright* than the same "white" reference whose radiance has been doubled:

$$\left.\begin{aligned}[\mathsf{C}_{\text{blue}}] &\overset{B}{=} [\mathsf{W}]\\ [\mathsf{C}_{\text{yellow}}] &\overset{B}{=} [\mathsf{W}]\end{aligned}\right\} \qquad [\mathsf{C}_{\text{blue}}] + [\mathsf{C}_{\text{yellow}}] \overset{B}{<} 2\,[\mathsf{W}]\,, \tag{6.2}$$

where the B subscript designates equality in terms of sensed brightness. Their experiment is among the first to exhibit additivity failures that could explain the existence of the Helmholtz-Kohlrausch effect and the inappropriateness of nowadays' standards for brightness modeling.

Instead of observing failures of the luminous flux additivity assumption, Guth *et al.* experimentally find the mixing coefficients for which their test and reference stimuli exhibit equal brightness, and observe the deviation from the "ideal" solution if the Abney's law holds [GDM69]. They use monochromatic stimuli $[\mathsf{C}_\lambda]$ to examine the evolution of the deviation across the visible spectrum. They first brightness match their test stimulus with a reference white stimulus, and then consider the heterochromatic stimulus created from the mixture of the test stimulus and the weakened white stimulus whose radiant power has been halved. Observers are instructed to tune the test stimulus' radiant power α_λ to re-establish the brightness match if necessary:



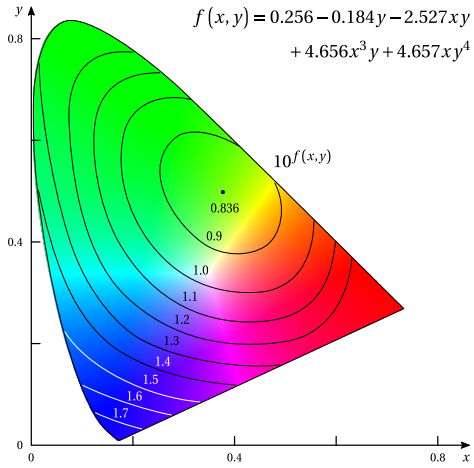
6.4 Additive Failure as Proven by Guth.

$$\begin{aligned}[\mathsf{C}_\lambda] &\overset{B}{=} [\mathsf{W}]\,,\\ \text{find } \alpha_\lambda \text{ such that} \quad \alpha_\lambda\,[\mathsf{C}_\lambda] + 0.5\,[\mathsf{W}] &\overset{B}{=} [\mathsf{W}]\,.\end{aligned} \tag{6.3}$$

If Abney's law stood, the spectral curve for α_λ should always equal 0.5. However, reported results significantly differ from this ideal situation as evidenced in Figure 6.4. These different findings tend to explain the Helmholtz-Kohlrausch effect as a consequence of the breakdown of Abney's law once applied in the general context of brightness estimation of heterochromatic colors.

6.1.4 Overcoming the Helmholtz-Kohlrausch Effect

Until now, all brightness encoding quantities fail to account for the interplay between lightness and chroma. First experiments by Sanders and Wyszecki demonstrate the inappropriateness of luminance in the case of illuminant colors [SW64]. The Helmholtz-Kohlrausch phenomenon is also considered as an anomaly of the Munsell system to the point color ordering in this system must be performed sequentially for each of its axes. Subsequent investigations by Wyszecky also corroborate this concern by reporting pronounced errors of the CIE 1976 lightness L^* as a brightness estimator for color surfaces. The only exception is the CIECAM02 lightness correlate which contrary to its CIECAM97 predecessor, accounts for complex phenomena, the Helmholtz-Kohlrausch effect included. Unfortunately, the modeling of its color appearance attributes is directly based on the precise knowledge of the viewing conditions. This requirement hinders us from relying on this advanced model without oversimplifying assumptions.



6.5 Loci of Constant $10^f(x,y)$ Over the xy Chromaticity Diagram. Ware and Cowan's conversion formula is among the first internationally adopted attempts to account for the Helmholtz-Kohlrausch effect that compromises the perceptual relevance of brightness measurements of heterochromatic stimuli. Although they may not exactly align with the deviation loci as found by Sander and Wyszecki (*cf.* Figures 6.2 and 6.3), the spatial evolution of the Ware-Cowan conversion factor's strength is still in accordance with experimental evidence of the Helmholtz-Kohlrausch effect.

The Ware-Cowan Conversion Factor [WC83]

Research partly explained the Helmholtz-Kohlrausch effect as the consequence of the shortcomings of Abney's laws for handling heterochromatic stimuli. As an attempt to circumvent the hurdle without questioning the experimental laws of color matching, the CIE recommended the inclusion of a supplemental experimental term – called the Ware Cowan conversion factor f – to establish brightness matching. Given a pair of colors of distinct chromaticity coordinates (x_1, y_1) and (x_2, y_2), and of respective luminance L_1 and L_2, the following test should be performed:

$$\log L_1 + f(x_1, y_1) = \log L_2 + f(x_2, y_2), \quad \text{with} \quad f(x,y) = 0.256 - 0.184y - 2.527xy + 4.656x^3y + 4.657xy^4. \tag{6.4}$$

If the equality holds after the inclusion of the correcting term f, then both colors are considered as sharing the same brightness. The visualization of this factor's influence over luminance on the xy chromaticity diagram (*cf.* Figure 6.5) shows it compensates for the previously observed deviations (*cf.* Figures 6.2 and 6.3). However, this alteration of luminance is only to be used in the context of pairwise brightness comparison and empirically incorporates chroma to restore unduly lost brightness matches. Its use in the context of our grey-scale converter is thus neither clear, nor reliable. Finding a proper ordering between color brightness is of capital importance in our case as it strongly affects our perception of the scene, and carefree grey assignments yield surreal looking outputs. Whether of not luminance after correction by the Ware-Cowan produces a satisfactory ordering is doubtful since its primary objective is to enable brightness matching. Additional its validity is only confined to the case of unrelated colors and is then hardly suitable for our purposes.

Broader models for perceived brightness have been proposed by researchers specialized in color appearance modeling, notably Fairchild and Nayatani. The brightness estimates they propose both strive at the proper integration of chroma in our final perception of brightness and therefore take a specific care for predicting the Helmholtz-Kohlrausch phenomenon's influence over it. As a consequence, these predictors constitute good candidates for our global grey-scale mapping and we now need to study their respective effects and scope of use for choosing the most appropriate one with respect to our end application. These enhanced predictors fall into two categories depending on the nature of the handled visual stimuli, either colored lights or surface colors.

Fairchild and Pirotta's Brightness Estimators [FP91]

Fairchild and Pirotta build an improved lightness model out of the measurements gathered by Wyszecki's 1967 study. Thanks to the intensive number of available observations (43 colored ceramic tiles considered by 76 test subjects), they propose more elaborated models than Ware and Cowan's luminance correction via statistical optimization while keeping their predictors as simple and concise as possible. In their models, the most de-

termining factor for predicting the Helmholtz-Kohlrausch effect is the chroma $C^*_{ab} = \sqrt{a^{*2} + b^{*2}}$ as evidenced by the expression of their simplest predictor L^{**}_1:

$$L^{**}_1 = L^* + 0.143 C^*_{ab} \tag{6.5}$$

They later incorporate the effect's hue dependency commonly observed in literature on heterochromatic brightness matching. Indeed, perceived brightness exhibits smaller deviations for yellow hues. However, the limited number of samples *of the same chroma* hinders them from fitting an analytical function of the hue angle $h^\circ_{ab} = \tan^{-1}\frac{b^*}{a^*}$ to the observations. They instead arbitrary choose the absolute value of a half-sinusoid curve assuming a minimum at $h^\circ_{ab} = 90$.

$$L^{**}_2 = L^* + \left(0.116\left|\sin\frac{h^\circ_{ab}-90}{2}\right| + 0.085\right) C^*_{ab} \tag{6.6}$$

Fairchild and Pirotta validate their model on Sanders and Wyszecki's 1958 data set consisting of 106 lightness matches by three observers [SW58], and report enhanced brightness prediction performance in comparison to the CIE 1976 lightness. Because of the low number of test subjects, they conduct their own experiment by instructing 11 test subjects to brightness match 36 colored papers whose colors span the three dimensions of the Munsell system. This additional experiment highlights the influence of the overall lightness on the effect's strength: the larger the overall lightness is, the weaker the deviations from the forecast lightness are. Their final lightness predictor L^{**} hence modulates the chromatic contribution by a lightness-dependent multiplicative factor:

$$L^{**} = L^* + (2.5 - 0.025L^*)\left(0.116\left|\sin\frac{h^\circ_{ab}-90}{2}\right|\right) C^*_{ab} \tag{6.7}$$

Nayatani's Brightness Predictors [Nay97] Further investigations of the Helmholtz-Kohlrausch phenomenon have been led by Pr. Yoshinobu Nayatani who intensively studied its manifestations and proposed unmatched contributions for its formalization and understanding. Nayatani describes two separate and complementary experimental protocols for its study: the VAC (Variable-Achromatic-Color) and the VCC (Variable-Chromatic-Color) methods. Both set-ups involve bipartite stimuli made of a chromatic and an achromatic half that are to be brightness matched, but differ in their matching task. The VAC method – which encompasses Sanders and Wyszecki's as well as Fairchild and Pirotta's experiments – expects observers to adjust the *achromatic* stimulus so that its luminance/lightness coincides with the chromatic half's. Conversely, the VCC approach relies on the adjustment of the *chromatic* half so that it appears as bright as the achromatic reference. Depending on the procedure, the observed Helmholtz-Kohlrausch effect's strength drastically varies.

Unlike Fairchild, Nayatani tackles both solid and luminous colors, and fits his predictors on observations obtained using his two proposed procedures. Expert in color appearance modeling, Nayatani highlights the importance of the adapting luminance L_a over the Helmholtz-Kohlrausch phenomenon's magnitude, and thoroughly studies the effect of the illuminating conditions during the training and validation phases. For the most part, the analytical expressions of his predictors originate from Nayatani's non-linear color appearance model fitted on the impressive number of 305 samples of distinct chromaticities. For the sake of simplicity, we only present their expressions once transposed in the CIE 1976 UCS, as functions of the color's hue $h^\circ_{uv} = \tan^{-1}\left(\frac{v'-v'_n}{u'-u'_n}\right)$, and saturation $s_{uv} = 13\sqrt{(u'-u'_n)^2 + (v'-v'_n)^2}$ [2]. Unlike the CIE $L^*a^*b^*$ space, color spaces derived from CIE $L^*u^*v^*$ enable the approximation of a saturation correlate (colorfulness of a stimulus relative to its *own* brightness), enabling Nayatani to propose equations for the Helmholtz-Kohlrausch effect not only for solid colors:

$$L^*_{N_{VAC}} = L^* + \left(-0.1340q\left(h^\circ_{uv}\right) + 0.0872K_{Br}\right) s_{uv}L^*, \tag{6.8}$$

[2] The n subscript refers to the chromaticity coordinates of the reference white. While Nayatani originally used the C illuminant to normalize its coordinates, it does not constitute an international standard, contrary to the D series illuminants advocated by the CIE. Our implementation resorts to the D_{50} illuminant as it turned out to yield the most satisfying results while still describing comparable lighting conditions to the C illuminant's.

$$L^*_{N_{VCC}} = L^* + \left(-0.8660q\left(h^\circ_{uv}\right) + 0.0872K_{Br}\right)s_{uv}L^*, \tag{6.9}$$

but also for luminous colors:

$$L_{N_{VAC}} = 0.4462\left(1 + \left(-0.1340q\left(h^\circ_{uv}\right) + 0.0872K_{Br}\right)s_{uv} + 0.3086\right)L, \tag{6.10}$$

$$L_{N_{VCC}} = 0.4462\left(1 + \left(-0.8660q\left(h^\circ_{uv}\right) + 0.0872K_{Br}\right)s_{uv} + 0.3086\right)L. \tag{6.11}$$

The first pair of equations predicts the effect of color on lightness, while the second quantifies deviations in terms of luminance. K_{Br} accounts for the impact of the illuminating conditions expressed in Nayatani's non-linear appearance model as the following function of the adapting luminance L_a:

$$K_{Br} = 0.2717\frac{6.469 + 6.362L_a^{0.4495}}{6.469 + L_a^{0.4495}}. \tag{6.12}$$

The $q\left(h^\circ_{uv}\right)$ coefficient encodes the changes in the Helmholtz-Kohlrausch effect due to hue as:

$$\begin{aligned} q\left(h^\circ_{uv}\right) = & - 0.01585 \\ & - 0.03017\cos h^\circ_{uv} - 0.04556\cos 2h^\circ_{uv} - 0.02667\cos 3h^\circ_{uv} - 0.00295\cos 4h^\circ_{uv} \\ & + 0.14592\sin h^\circ_{uv} + 0.05084\sin 2h^\circ_{uv} - 0.01900\sin 3h^\circ_{uv} - 0.00764\sin 4h^\circ_{uv}. \end{aligned} \tag{6.13}$$

Nayatani intensely tested his models on different data sets, under a wide range of illuminating conditions for mesopic and photopic vision (the cones cells being irresponsive in scotopic vision), with stimuli of varying sizes, and concluded to the relative independence of the Helmholtz-Kohlrausch effect with respect to the field of view. Without any specific treatment, his metrics convincingly correct the spectral luminous efficiency curves established from *monochromatic* brightness matching of 2° and 10°-wide stimuli, to make them coincide with their corresponding curves involving *heterochromatic* stimuli.

6.1.5 Embedding the Helmholtz-Kohlrausch Effect in our Grey-Scale Conversion

Considering back our grey value assignment issue, we can draw insightful teachings from the research dedicated to the study of the Helmholtz-Kohlrausch effect. Defined as the brightness of a color judged relatively to the equally bright reference white's, *lightness* already answers some of our perceptual accuracy related concerns such as the visual relevance of the chosen grey values and the preservation of a compelling ordering between colors. But as the non-linear fit of the normalized luminance $\frac{Y}{Y_n}$ [3] to the Munsell value, it simply overlooks the impact of color on our perception of brightness:

$$L^* = \begin{cases} 116\sqrt[3]{\frac{Y}{Y_n}} - 16 & \text{if } \frac{Y}{Y_n} > \left(\frac{6}{29}\right)^3, \\ \left(\frac{29}{3}\right)^3\sqrt[3]{\frac{Y}{Y_n}} & \text{otherwise.} \end{cases} \tag{6.14}$$

Yet, more than a superfluous artifice, the Helmholtz-Kohlrausch effect directly questions the perceptual quality of a straightforward lightness centric grey-scale conversion. Bala and Braun stress the need to account for the contribution of chroma on perceived lightness and incorporate its effect by using Fairchild's earliest brightness predictor L_1^{**}, alas disregarding the influence of hue and overall luminance in the process [BB03]. In addition, while suited to clipart-like images composed of regions of constant color, their explicit color reordering precludes the application of their technique to complex imagery. The inclusion of the Helmholtz-Kohlrausch effect in a grey-scale conversion method aiming at a more generalized field of application is hence left unexplored. We propose to account for the Helmholtz-Kohlrausch effect at an early stage of our conversion algorithm by independently mapping colors to their grey values with the objective of perceptual faithfulness in mind, and rely on one of the aforementioned brightness predictors to carry on this task.

Several considerations now influence our choice for the most appropriate brightness metric:

[3] Y_n denotes the luminous reflectance of the reference white against which the test color brightness is assessed.

- the suitability of the experimental instructions that led to the training observations with respect to our purpose;
- the nature of the involved stimuli as well as their quantity and coverage of the chromaticity diagram;
- and lastly the intrinsic quality of the models.

Since color to grey-scale image conversion presupposes the estimation of the depicted objects' brightness, we can leave Nayatani's estimators for luminous colors behind. Second, following Nayatani's comment on the practical use of his predictors, the VCC procedure, by requiring subjects to brightness match the chromatic stimulus to the achromatic reference, is less compliant to our objective than the VAC procedure, and instead strives for the rehabilitation of heterochromatic brightness matching. The distinction between the two set-ups is not a cosmetic concern, as the VCC correction's magnitude is twice as important as the VAC's in logarithmic space, making the decision between the two quite influential. For its appropriateness to our grey-scale assignment goal and gentler modeling of the hue influence, we favor the predictors obtained with the VAC method.

Beside the experimental apparatus employed to collect them, the amount of observations restrains the complexity of the fitted model and conditions the robustness of the statistical optimization used for its training. The smaller the set of available samples, the less parameters the model can afford to rely on. Bounded by the subset of equichromatic samples from Wyszecki's 1967 experiment, Fairchild and Pirotta can only represent the influence of hue as a sinusoid upon which he performs linear regression (*cf.* Figure 6.6*(left)*). Additionally, though empirically observed, their overall lightness-dependent term $(2.5 - 0.025L^*)$ is not actually validated through experiments, and simply ensues from the linear interpolation that states that the global lightness's influence over the Helmholtz-Kohlrausch effect is equal to 1.0 for $L^* = 60$ (which corresponds to the lightness level of Wyszecki's 1967 data set) and to 0.0 for $L^* = 100$ (no Helmholtz-Kohlrausch effect on a perfectly white stimulus). This lack of experimental validation jeopardizes the perceptual relevance of their L^{**} estimator for images exhibiting various intensity levels.

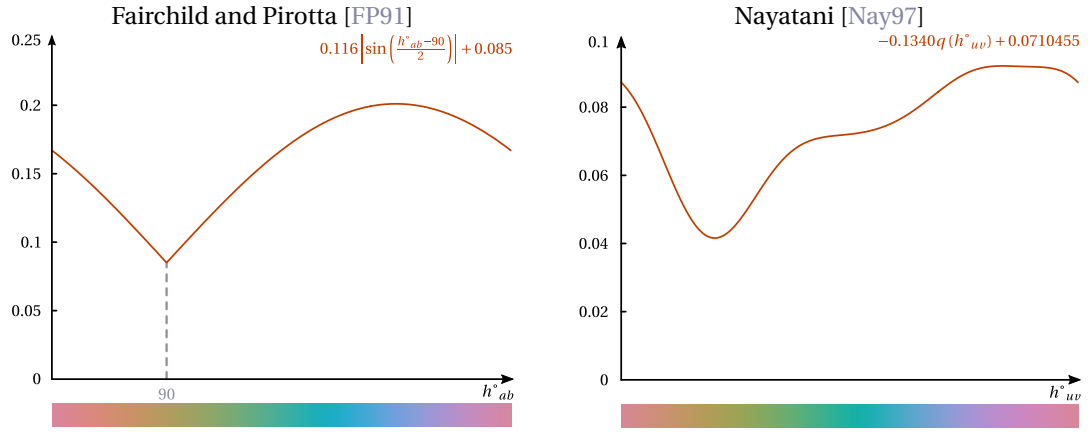
Conversely, Nayatani's non-linear color appearance model was trained over a more substantial number of observations, and allows him to represent the hue dependency using a finer function than Fairchild's sinusoid (*cf.* Figure6.6*(right)*). Additionally, Nayatani's metrics follow the scheme of the Ware-Cowan equations in the sense that they predict the changes in the *ratio* between perceived brightness and measured luminance/lightness, and do not consist in incremental corrections of the deviations by the adjustment of additive terms. They hence better generalize to changes in overall lightness.

The comparison between the corrected lightness plots in Figure 6.7 may give the illusion that the effective differences between Fairchild's L^{**} and Nayatani's $L^*_{N_{VAC}}$ are niggling. Their comparable action indeed finds an explanation in their shared experimental procedure and partly common training data set. But Fairchild's blunter handling of the hue effect makes his chromatic contribution globally stronger, notably for blueish hues, and thus markedly reduces the range of his brightness and diminishes discriminability between equiluminant colors.

Bearing all these considerations in mind, Nayatani's $L^*_{N_{VAC}}$ metric takes the upper hand, both in terms of the quality of its representation and suitability to our grey-scale conversion objective. Our global mapping to apparent brightness L^*_N therefore consists of the following conversion steps. Assumed to have its colors represented in the *sRGB* color space, the input image $\mathscr{I}$ is first mapped to the linear *RGB* space by inverse gamma mapping before having its pixels' tristimulus values XYZ computed. It is then converted to the CIE $L^*u^*v^*$ color space where the Helmholtz-Kohlrausch aware lightness L^*_N can be estimated as indicated by Equation (6.8). Once gathered, the pixels' lightness values L^*_N serve alongside the reference white chromaticity coordinates (u^*_n, v^*_n) to transform the decolorized image from CIE $L^*u^*v^*$ back to linear *RGB* space. Lastly, gamma-mapping is restored to bring the image back to the device-dependent *sRGB* specification and obtain the final grey-scale output $\mathscr{G}$.

Not surprisingly, the apparent brightness mapping may attribute lightness values to distinct colors resulting in the same shade of grey after range quantization. The explanation lies in the dimensionality reduction

[4] In the case of Nayatani's factor, the adapting luminance is set to $L_a = 20cd/m^2$.



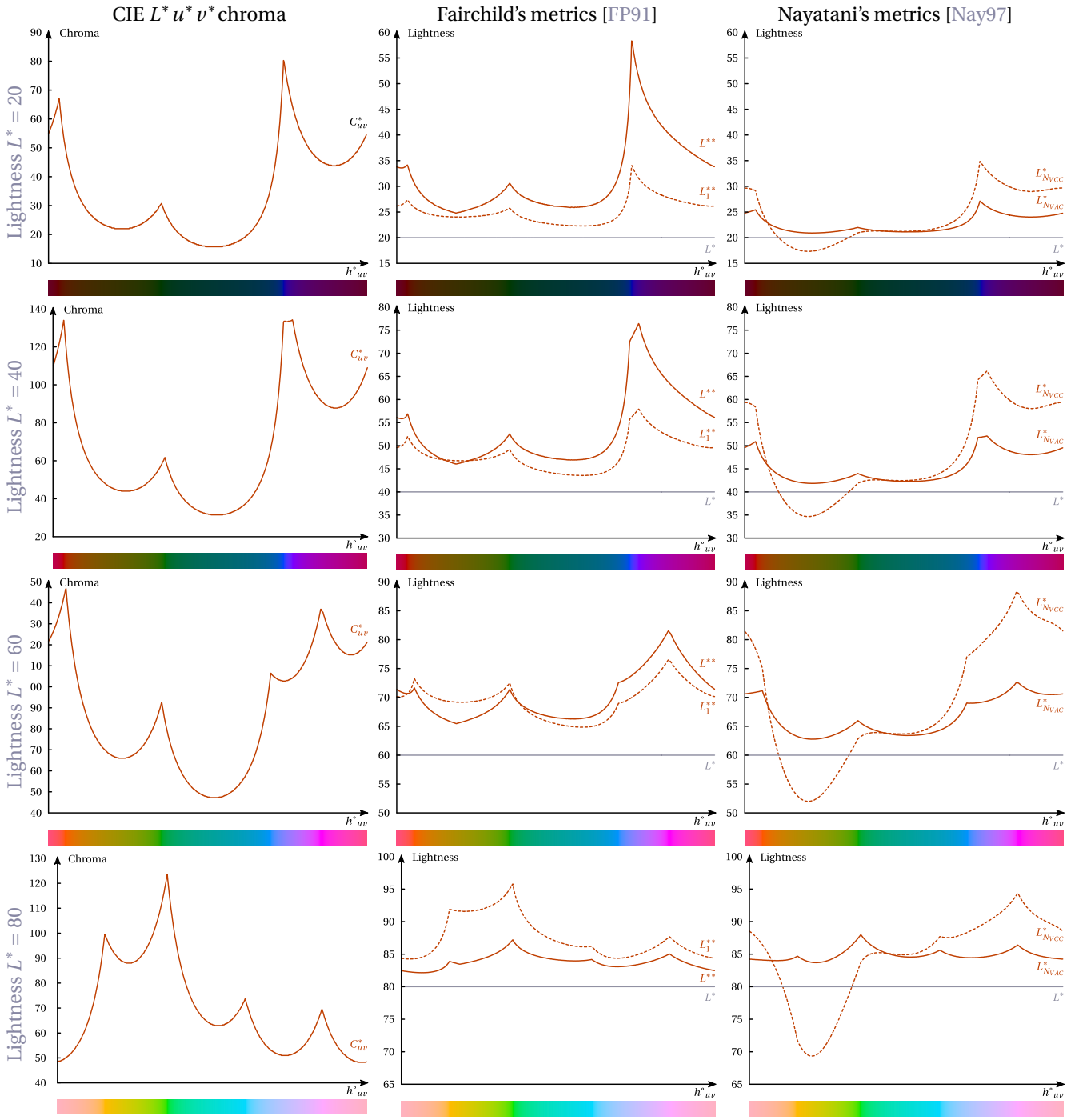
6.6 Fairchild's and Nayatani's Modeling of the Hue Effect . In analytical modeling by regression, the flexibility attainable by the model is directly linked to the size of the training data set. Bearing this unavoidable compromise in mind, Fairchild had no choice but to model hue dependency as a simple sinusoid, whereas Nayatani can fit a more sophisticated function[4]. Despite their differences, both models still inhibit the chroma contribution to lightness around yellow hues.

nature of grey-scale conversion that inevitably engenders information loss, but also in the nature of mapping itself, that we chose to be entirely data-independent. Not only for the sake of speed (global optimization over all pixel pairs representing a staggering computation load), relying on a data-dependent mapping to assign grey values to colors would have compromises its handling of animations, especially if exhibiting abruptly changing color palettes. Subsequent work to ours target this specific issue of extending optimization across the frames of an animation [KJDL09], but at the cost of additional and uncertain calculations. Uncertain in the sense that the optical flow needed to ensure meaningful pixel correspondences across frames is practically in most cases not available, and that the propagation of the grey-scale distances constraints may significantly compromise the output's perceptual relevance which is at the heart of our approach. Moreover, thanks to the intensiveness of the experiments that led to its creation, grey values returned by Nayatani's $L^*_{N_{VAC}}$ predictor are much more accurate than the colors' lightness values, and although discriminability must nevertheless be restored, colors of confusingly similar $L^*_{N_{VAC}}$ values are indeed confusing in terms of brightness perception-wise.

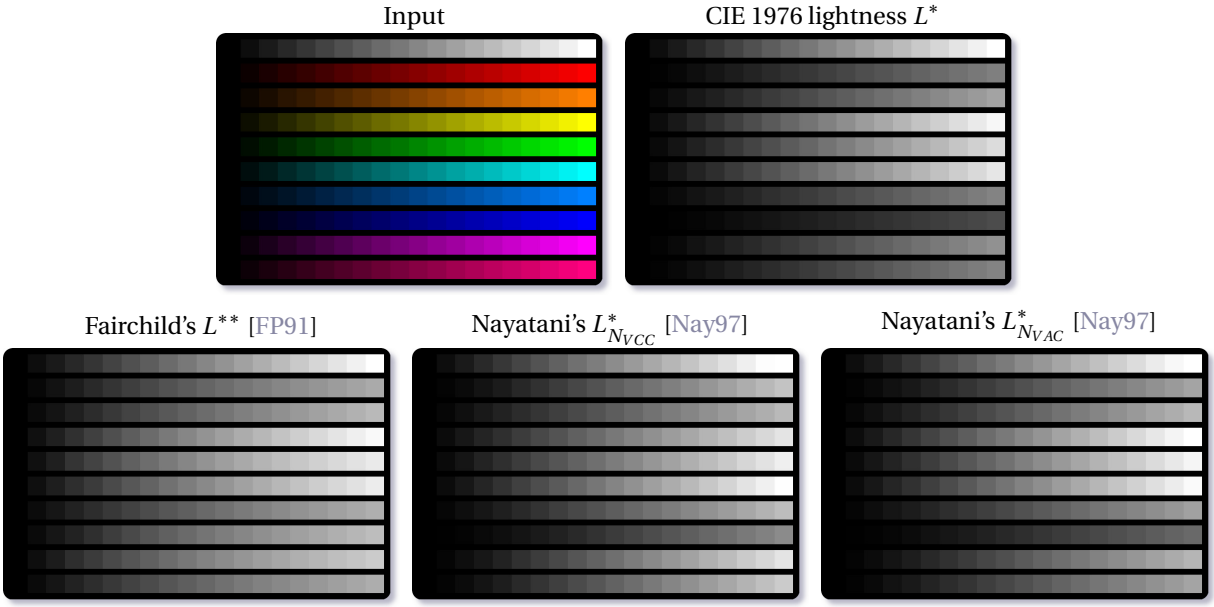
6.2 Chromatic Edges to the Rescue

Our perceptually motivated global mapping outputs visually pleasing grey values with respect to the original color image, and proposes a sensible luminance ordering. However, its fixed nature hinders it from preserving the original image's local chromatic features as faithfully as a locally tuned or fitted mapping. As we previously did, we can take benefit from the insights of research in biological vision, and use local contrast enhancement to satisfy our new-found another perceptually motivated handle to satisfy our newfound discriminability objective. For our contrast operator to be effective and our results to maintain their perceptual pleasantness, this enhancement has to work in unison and as such, takes the previously computed grey-scale image $\mathscr{G}$ as input. While it can be argued whether or not our contrast adaptation is actually more of a post-process than part of our grey-scale conversion [CHRW09], we certainly did not it as an added-on beautification procedure, but as an integrant part of our algorithm.

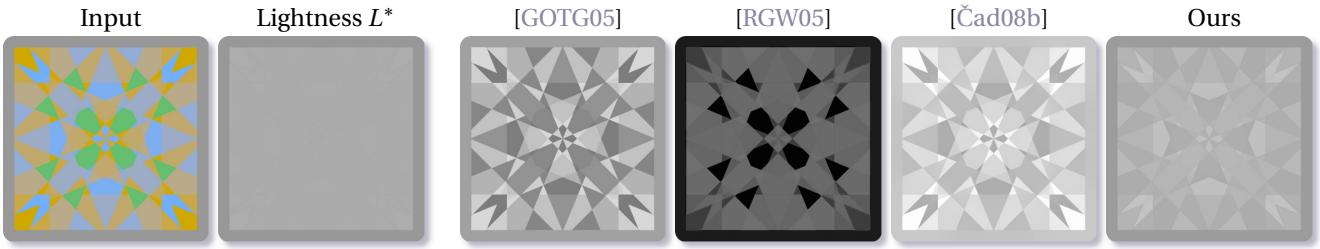
Findings in cognition and biological vision hint that contrasts are fundamental clues governing our percepts of our surrounding world. Important to the point that our intensive reliance to them can become misleading as many examples of illusory brightness assignments have been analyzed by scientists. This later point is inspiring for our perception-friendly conversion: if we could get a hold on such *illusory* brightness effects, we could subtly grant improved discrimination to our results. Moreover, not only has our visual system grown an acute sensitivity to contrasts, but it happens that their sharpness in images also governs our impression



6.7 Visualizing the Helmholtz-Kohlrausch Effect. Above are different plots showing the different brightness metrics at our disposal *(center and right columns)*. Along with the classical CIE 1976 lightness *(in grey)*, the number of alternatives raises to four: Fairchild's L_1^{**} and L^{**} *(center: dashed, solid red curves)*, as well as Nayatani's $L^*_{N_{VCC}}$ and $L^*_{N_{VAC}}$ *(right: dashed, solid red curves)*. Test color ramps span the whole hue wheel at several lightness levels *(different rows)*, with the highest possible chroma C^*_{uv} *(left column)* so that the chromatic lightness' contribution to brightness is accentuated. For devising the best pick between these estimators, several observations can be made: 1) L_1^{**} can be ignored as contrary to the others, it does not account for the hue dependency (lesser impact at yellow hues, increase for blue ones). 2) From their common experimental procedure, L^{**} and $L^*_{N_{VAC}}$ roughly describe the same shape, especially at higher luminance levels, with an improved smoothness level for the latter. 3) Both L^{**} and $L^*_{N_{VCC}}$ have a pronounced tendency for overstating the effect and are prone to brightness overshooting. For L^{**}, this is explained by Fairchild's hasty handling of the overall lightness' influence that he models via an under-constrained linear weighting (this is especially visible for low lightness). 4) Of all mappings, $L^*_{N_{VAC}}$ is the most faithful to the original lightness by proposing the most natural grey values while incorporating the same amount of variations as the others. For these different reasons, we opted for Nayatani's $L^*_{N_{VAC}}$ metric as our global mapping to apparent brightness.



6.8 Assessing the Plausibleness of the Color-Aware Brightness Metrics. Though it does not include the Helmholtz-Kohlrausch effect, standard lightness L^* is nevertheless a perceptually relevant brightness predictor, and its corrected versions should not differ from it too significantly. Fairchild's L^{**} clearly overstates the impact of chroma for most hues, especially the blue ones. Even Nayatani's exaggerated $L_{N_{VCC}}$ metric is unsuited for our application. But the best alternative is obviously Nayatani's $L_{N_{VAC}}$: it does not alter the lightness perception thoughtlessly while incorporating the effect of color *(especially visible red row)*. The overall appearance stays close to what one would expect, and the ordering is satisfactory. Source: SONERA Technologies.



6.9 Importance of Luminance Ordering. This image sporting perfectly equiluminant colors shows the pitfall of optimization schemes only driven by distance preservation. Although our intermediary result lacks other techniques' discrimination power, it is the most perceptually faithful output from all global mappings presented here. The closest concurrent is Neumann and Čadík's less contrasted, "perceptually plausible" conversion. Their grey value assignment could be debated though, as their method flags the light blue regions as brighter as the dull yellow ones. Source: Related work images courtesy of Martin Čadík [Čad08a].

of visual quality. Countless examples can be drawn from art, and constitute empirical proofs of the intangible association between crisp contrasts and visual satisfaction. Manipulating contrasts compensating for the information lost by our global mapping hence appears as promising.

6.2.1 The Chicken and the Egg: Contrasts and Contours

The prevailing claim behind our conversion approach is that a technique that entirely overlooks the human visual system's mechanisms cannot produce truly satisfactory results. It may construct them so that their grey values meticulously depict the short or wide-range chromatic differences found in the original image, but it is likely to be at the expense of visual pleasantness, or even plausibility. It is also arguable whether or not a technique blindly favoring discriminability ends up producing genuinely informative results, especially if its grey-scale outputs elicit the visual impression that they stem from a different color image. In the case of drawings or photographs, shuffling grey values for the sole sake of local color discrimination may actually turn out to be more dreadful than impoverished contrasts.

In opposition to most competing techniques (at the only exception of Neumann and Čadík's converter [NCN07]), we want our technique to preserve the perception of the original image as best as possible. The variety of processing at our disposal is hence limited by the level of tolerance of our visual system to the changes caused by the involved operators. Our acute sensitivity to spatiotemporal changes, and most importantly our uneven sensitivity to different *kinds* of changes incites us to resort to contrast enhancement.

6.2.1.1 Global Contrasts for Enhancing Range

Intuitively, contrasts refer to the local patterns emanating from the juxtaposition or brisk transition between regions differing in brightness or color. They are especially helpful to distinguish between nearby objects, or delineate figures from the background. Since critically important to beings mostly resorting to vision for interacting with their surrounding, evolution gifted us with an increased sensitivity to such visual stimuli. As swift luminance gradients reveal the geometrical shape of surfaces or indicate that objects are observed under a grazing view angle, contrasts are pre-attentively linked our notion of *boundary*, and are prevalent in our vision. Contrast sharpness is therefore one of the most valuable non-content-related attribute[5] of our visual percepts.

But beside conveying crucial information on the surroundings, contrasts also influence our sense of visual appeal. Art across History abounds with edifying examples where contrasts are deliberately used for the sole purpose of aesthetics. In western art, Italian painters from the Mannerist and Baroque art movements whose *chiaroscuro* technique soon spread across Europe from the 16th to the 18th centuries, were the first to attach such specific attention to global contrasts. To the point *Chiaroscuro* which consists in the depiction of stark contrasts between lit areas and pitch darkness is among the most distinguishing features of the paintings from this period. The role of light and illumination is indeed at the heart of their creations' underlying message: often depicting religious scenes or mythical beings, they mostly represent isolated god-like light sources clashing violently against the surrounding earthy darkness, and often have the entirety of their composition governed by the ensuing contrasts. Such dramatic lighting results in shadows revealing the objects' and characters' outlines, and fixes fleeting scenes over the canvas as sharply as photography (*cf.* Figure 6.10). From this flourishing artistic period, paintings considerably gained in realism, the interplay between light and shadows conferring an enhanced 3d sense to the portrayed objects, especially the human anatomy.

Michelangelo Merisi da Caravaggio
Supper at Emmaus
1601. 1606.

Georges de la Tour,
Apparition de l'ange à St Joseph
1645.

6.10 Global Contrasts in Art. *(left)* Are proposed two representations of the same biblical scene executed by the *same artist* and following to the same composition. As subtle as it may sound, the difference only lies in the handling of contrasts which make the mood of the two paintings barely comparable. With these, *Il Caravaggio* marks the advent of the *chiaroscuro* apinting technique, and proposes a masterful illustration of the impact of contrasts on dramatization and visual pleasantness. *(right)* De la Tour also reveals himself as a master of the technique in his trademark candle-lit scenes where contrasts serve for the depiction of realistic lighting which delineate shapes and guide the composition.

[5] in opposition to semantics-related visual attributes, such as people, facial expressions or actions.

Avalon,
Mamoru Oshī, 2001.

Un long dimanche de fiançailles,
Jean-Pierre Jeunet, 2004.

300,
Zack Snyder, 2007.

6.11 Global Contrasts in Movies. With modern post-processing techniques, directors can polish the appearance of their scenes so that they best match their imagination. Among recently released, heavily retouched movies, the above examples exhibit a specific use of contrasts for dramatization at the expense of realism. Interestingly enough, all share a limited color palette made of shades of sepia, and our understanding of the scenes is mostly guided by luminance contrasts. In the Avalon screen shot *(left)*, contrasts contribute to figure-background segmentation, the silhouettes of the black clad characters popping out from the lighter, foggier setting. Back lights also constitute a proven technique for emphasizing light obstructing characters and adorning them with strong contrasts. But contrasts on the still from *Un long dimanche de fiançailles (center)* spread across the whole scene, from the rim lighted characters' faces to the checkerboard pattern on the floor. In 300, contrasts give the illusion that characters are wearing armors *(right)*.

Nowadays, similar considerations are mostly encountered in photography and cinematography where the careful lighting of faces and shadow defined demarcation lines play an intense role not for dramatics, but also for aesthetics and scene understanding. Some extreme cases, such as comics inspired or fantasy movies, often rely on intensive post-treatments to create phantasmagoric and surreal atmospheres. But contrast enhancement now pervades movies beside such creations, and many aesthetically powerful movies now exhibit finely crafted contrasts which seem sometimes at odds with the actual settings, but always end up in eye pleasing results (*cf.* Figure 6.11).

6.2.1.2 Local Contrasts for Enhancing Contours

If we follow Patrick Cavanagh's scheme of thought [Cav05], art can be considered as an open window towards the understanding of our biological vision, and artists, *"similarly to neuroscientists"*, try our visual system's strengths and weaknesses out with their work. The pronounced and steady resort to strong contrasts in art then constitutes empirical proof that they appeal to the eye. Contrasts thus represent an interesting playground to explore for improving our grey-scale results.

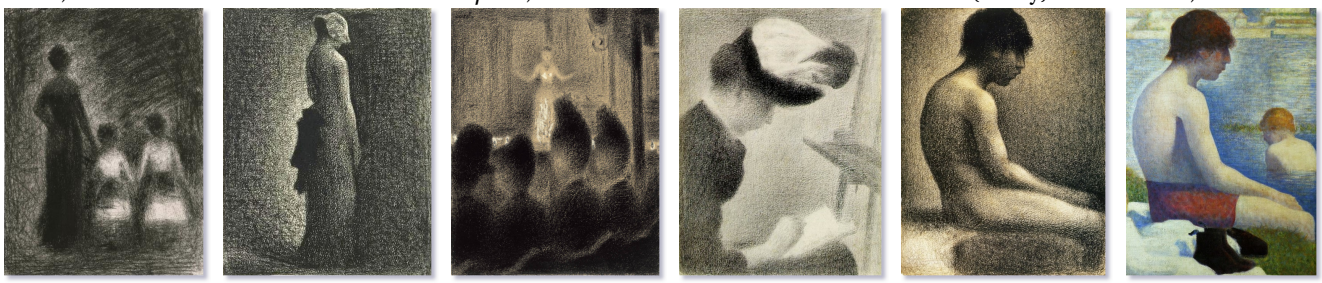
However, though meaningful as aesthetical validation, the plays on contrast presented thus far are misleading in the sense that their intensity and spatial extent take the upper hand over the faithful depiction of the scene. Processing our images in order to mimic such global contrasts would constitute a lunacy conflicting with our initial thrive for perceptual accuracy with respect to our original inputs. Alternatively, *locally* enhancing contrasts appears as a more sensible approach to fulfill our discriminability constraint, and while spatially more subtle, the pleasing effect of localized contrast enhancement remains. Indeed, artificially introducing or reinforcing contrast in images or videos increases the sensation of visual quality as reported by observers, and quantified by recent studies such as Lin *et al.*'s on the perceptual impact of edge sharpness [LGK06].

From the edge-driven contrasts to the notion of outline, there is only one easy step to take, and again edifying artworks exploit the tangled interaction between contrasts and *contours*. Exaggeratedly sharp contrasts are indeed often used to stress objects boundaries, providing assistance to figure-ground segregation, the distinction between objects and the understanding of the scene's spatial layout. Artists explore the effects caused by more localized, path following contrasts to achieve more detailed representations of shapes and enhance the viewer's perception of the objects' smaller features (*cf.* Figure 6.14). They are also strongly manipulated to reinforce objects' silhouettes. In professional photography, the careful lighting surrounding the subjects ensures the artful flair of professional portraits and of most importance is the discrete albeit determining *rim*

6.12 Highlighting Contours. In professional photography, it is commonplace to use fairly unrealistic albeit not visually disturbing lighting to enhance shape perception and confer an artistic mood to the result *(left)*. Along with the shadow casting *key light*, and softening *fill light*, the *rim light* is located behind the subject and enhances the silhouettes by very locally boosting the contrast with the background *(right)*. Source: Photograph by John Seah.

Georges Seurat

Femme avec fillettes, ca. 1882-84. *Le nœud noir*, 1884. *Au concert européen, ca.* 1886-88 *Femme assise devant chevalet.* *Baigneurs à Asnières* (study). *Baigneurs à Asnières*, 1884.



6.13 Local Contrasts in Pictorial Art. The above artworks are perfect examples of the artistic exploitation of the perceptual effects caused by edge-following manipulation of local contrasts. This trend is especially noticeable in the impressionist and pointillism art movements that were driven by the experimental will of artists. The influence of scientific findings such as Helmholtz', Maxwell's, Rood's or Chevreuil's is determinant for the choice of their color palette and arrangement. The stark effect caused by simple plays on contrast is impressive especially on dominantly dark or light mono-chromatic backgrounds, but can be transposed to color. Seurat called his *chiaroscuro* technique *irradiation*, and comforted art critic Charles Blanc's claim that "harmony can be born from the analogy of contrast". As Seurat wrote in one of his letters, "Art is harmony. Harmony is the analogy of contrasts[...] The contrasts are of tone, a brighter for a darker."

light which located behind them to accentuate their delineation over uniform backgrounds (*cf.* Figure 6.12). In pictorial art, French Pointillist painter Georges Seurat's series of drawings are masterful examples of how effectively carefully disposed and shaped contrast lines can enhance contours and alter our perception of brightness via clever intensity discontinuities (*cf.* Figure 6.13). Seurat's artwork *Le nœud noir* has notably been used by Margaret S. Livingstone to illustrate how contrasts alone could be used to create the sensation of a wider luminance range than the medium[6]'s actual one. Seurat's countershading (the background being lightened and directly put into contact with the lady's black silhouette) makes the character almost loom up. His resort to this effect pervades his whole series of sketched feminine figures, either over darker or lighter backgrounds, as well as in his studies or even color paintings (*cf.* Figure 6.13). Many other examples found in paintings of various different art styles also exhibit similar contrast effects as illustrated by Ryder's still life (*cf.* Figure 6.15). This painting is an as interesting illustration of the effect as it aches for a faithful depiction of reality of photographic quality, and is not driven by the same will of visual experimentation as Seurat's.

Research in Computer Graphics also explore the enlightening power of local contrasts for improving the readability or beautifying the computer-generated imagery. Disambiguating confusing shapes, or making more explicit the world-space organization of the scene are formidable challenges faced in expressive rendering. The accentuation of shape-driven contrasts do wonder for such tasks since they are unconsciously related to visual phenomena such as self-shadowing or the compression of light patterns at the grazing angles of glossy surfaces. Many alternatives exist to automatically introduce such contrasts in automatic renderings

[6] Here, simple greasy conté crayon on laid paper.

Ami Talluto,
The Maple Forest, 2008.

6.14 Global Versus Local Contrasts. Talluto's painting illustrates the dual role of global and local contrasts. While contrasts of wider spatial extent define the artwork's composition and induce most of the 3d sensation, more localized contrasts enhance our perception of the objects' surfaces. Here, global contrasts are mainly caused by the maple trees' trunks standing against the clearer, flatter background. Conversely, local contrasts evolve at a different scale and exaggeratedly emphasize their roughness.

Anthony J. Ryder,
Warm and Cool, 1999, Oil on limen.

6.15 Tolerance to Local Contrast Alterations. The subtle darkening around the rose's petals illustrates the kind of contrast enhancements we target in order to reinforce and restore the input's lost chroma edges. But more than an exemplification, it also constitutes an empirical evidence that such contrast changes do not deteriorate the image quality but by sharpening it beautify it instead.

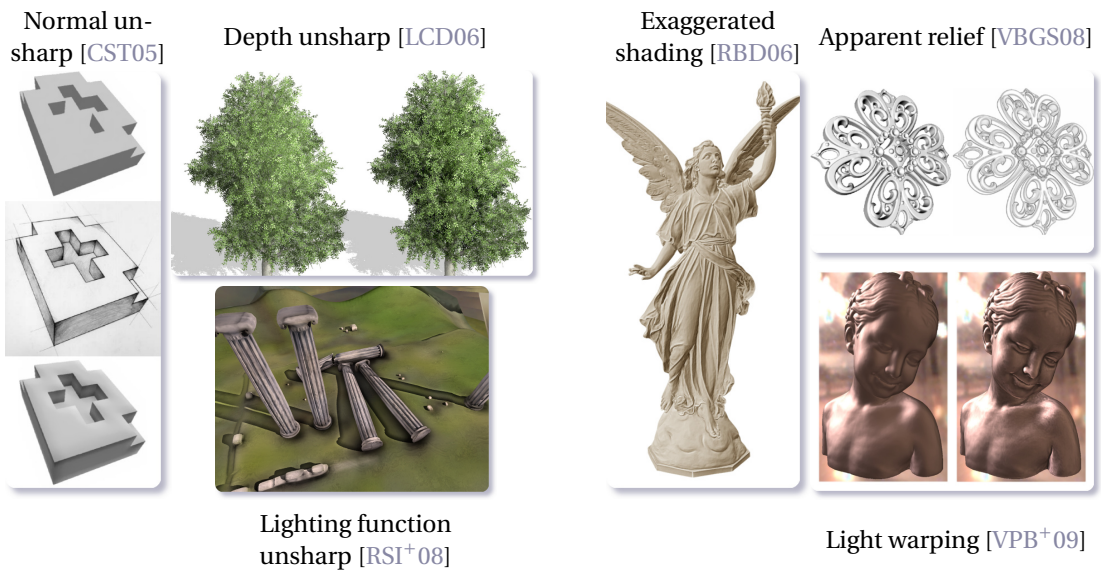
as presented in Figure 6.16. Most shape enhancing techniques ultimately boil down to locally modifying how light is reflected by the objects' surfaces, in other words their shading. A first approach is to perturb the surface normals in order to accentuate shading discontinuities and reflections, and reinforce the perception of their geometric details and curvature such as Cignoni's *normal enhancement* [CST05] or Vergne *et al.*'s *light warping* [VPB+09]. Contrariwise, the dual take on this problem is to alter or process the lighting itself. Examples include Rusinkiewicz's *exaggerated shading* which relies on the adaptive placement of the light source so that it best reveals the surface's local features [RBD06], or Ritschel's 3d consistent sharpening of the reflected radiance [RSI+08] which also enhances shadows and material details as it processes the signal of the camera impinging light. Similarly, Luft's technique for augmenting the perception of depth also relies on the unsharp masking filter [LCD06].

6.2.1.3 Contours for Enhancing Global Contrasts

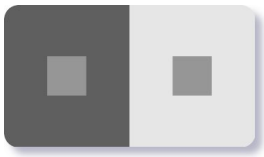
Contrasts can bring out contours in images and make figures stand from the background, but the opposite phenomenon is also witnessed. This *converse effect of contours on contrast* constitutes a even more stunning evidence of our perceptual sensitivity to contrast, to the point that even the slightest direct intensity opposition subjugates our sensation of global brightness. This last observation is interesting for us to reintroduce lost chromatic contrasts with the least intrusive changes possible.

Perceptual Illusions

Perceptual illusions are the most striking evidences of our error-prone perception of brightness and are all the more impressive as they are usually associated with surprisingly simple stimuli. Chiefs among them, *simultaneous brightness contrast* and *simultaneous color contrast* unveil the influence of the surrounding luminance values or colors over a central stimulus of interest. Figure 6.17*(left)* displays the minimal stimulus necessary for the brightness illusion to take effect. Simultaneous color contrast is an even more perplexing



6.16 Enhancing Shape Perception in Computer Graphics. Themes such as visual understandability are especially dear to the expressive rendering community which endeavors to find meaningful ways of omitting superfluous details while maintaining the appropriate visual information. They often draw inspiration from artists' work, mostly from scientific illustrators. Their work notably reveals that emphasizing local contrasts mimicking the interplay of light with sharper surfaces considerably contribute to shape enhancement.

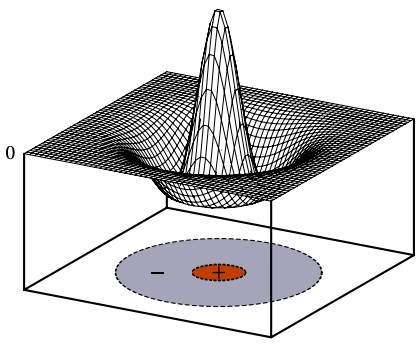


6.17 Contrasts as the Masterminds behind our Perception of Brightness. Many simple illusions hint that our percepts are mostly driven by the information conveyed by contrast more than the absolute light intensities. In this particular example, both squares share the same reflected light intensity *(right)*. However, observers report their impression according to which the left square appears brighter than the right one *(left)*.

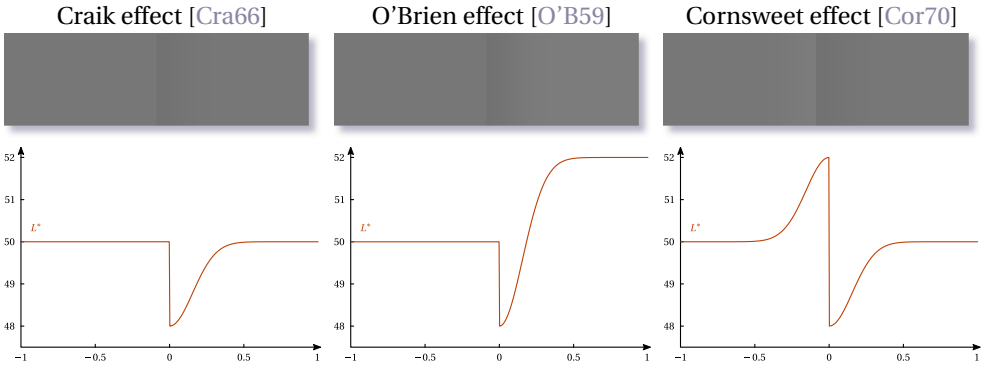
phenomenon as it reveals that surrounding colors induce a significant change in the center stimulus's perceived chromaticity which, moving away from the surrounds' chromaticity values, participates to the clearer distinction of the central stimulus to its immediate surround. Our perception hence appears subjugated to the presence of contrasts, and numerous research attempt to model the altered brightness sensation of an *in situ* stimulus by considering luminance ratios across contrast boundaries. While Nayatani's $L_{N_{VAC}}$ metric incorporates the chroma's contribution to lightness for colors considered *independently*, contrasts may help us to assign additional illusory brightness to our intermediary grey-scale results.

From a Physiological Perspective

The conflict between perceived brightness and the effective quantity of light reflected by an object's surface towards the eye directly refers to the retina's physiology and its constitutive cells'. Receiving signals from the rod and cone cells, some cells of the retina's ganglionic layer (*cf.* Figure A.5) happen to be highly responsive to the stimuli's contrasts. Such cells also exhibit an particularly interesting behavior in terms of neural responses and could be the reasons behind our acute sensitivity to contrasts. Using small light spots, neuroscientist Stephen Kuffler discovered that each of these cell was associated with a specific location over the retina upon which impinging light causes maximal neural response. Beside their spatial locality, he also observed that light falling onto the direct immediate vicinity of such spots of optimal response instead inhibit the considered cell. This particular spatially-dependent behavior of the reti-



6.18 Modeling of a Retinal Ganglion Cell's Finite Impulsion Response as a Difference-of-Gaussians Filter.



6.19 The Craik-O'Brien-Cornsweet Effects. The entoptic effects independently discovered by Kenneth Craik *(left)*, Vivian O'Brien *(center)* and Tom Cornsweet *(right)* are illustrations of the influence of contours on contrast. In all three cases, the lightness at either side of the central discontinuity is equal (if not higher on its right side) and despite this fact, observers report that the left halves appear lighter than the right halves, comparatively to a simple step edge.

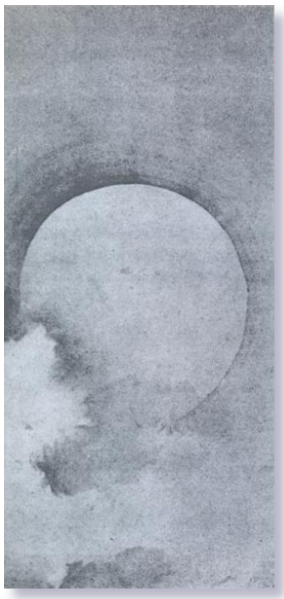
nal ganglion cells' responses corresponds to a *center-surround* organization, and the spatial extent for which a cell exhibits sensitivity is referred to as its *receptive field*. The mathematical formulation of such cells' neural responses is often expressed as the convolution of the eye impinging signal with a difference-of-gaussian filter, for it best emulates their strong response at sharp intensity changes (*cf.* Figure 6.18). Of uppermost importance, this modeling accounts for the *uneven* nature of their sensitivity: the sharper the contrasts (in our case, the steeper the slope of an image gradient), the stronger their neural response. Our vision then pays more attention to the width of an intensity transition than to its actual magnitude and it happens that this unbalanced responsivity can fool our visual system once the spatial integration of the neural responses is done.

The Craik-O'Brien-Cornsweet Illusions

Such tricks include the famous Craik-O'Brien-Cornsweet illusions that Floyd Ratliff considers as manifestations of the converse influence on contours on contrasts [Rat85]. They emerge from the center-surround organization of our visual system's ganglionic layer, but similar illusions can affect any sensory perception for which its receptors share a comparable organization. Earliest examples include Rawdon-Smith and Grindley's "illusory perception of loudness": they discovered that for most listeners, a sudden increase in volume followed by a gradual return to the original tone was perceived as a simple step-like increase, the illusion consisting in the fact that the progressive tone decrease was left unnoticed. Psychologist Kenneth Craik came to the same observations when having observers consider a visual stimulus following a similar spatial distribution as the temporal behavior of Rawdon-Smith and Grindley's stimulus (*cf.* Figure 6.19-(a)). Similarly, only the central discontinuity yields most of our perception of the stimulus, and most people perceive its left half as significantly brighter than its right half. This demonstrates how determining contrasts are to our visual percepts, but also how distinctly changes of different abruptness are treated: while the sharp central discontinuity is easily detected, the progressive return to the original intensity is almost indiscernible until the distracting discontinuity is covered. Physicist Vivian O-Brien and psychologist Tom Cornsweet independently exposed stimuli inducing an analogous perception of illusory brightness caused by a spatial luminance distribution closely following Craik's (*cf.* Figure 6.19*(b)* and 6.19*(c)* respectively). Of the three, O-Brien's is the most outstanding of the three as the gradual return actually exceeds the original intensity level [O'B59]. Notwithstanding, the illusion still holds and is in complete opposition with the physical amount of reflected light. Beside its steadiness, the illusion's spatial extent is also surprising as illusory brightness entirely fills the territorial area flanking the discontinuity. Cornsweet's stimulus may not be as ground-breaking as Craik's or as impressive as O'Brien's, yet it proves to be the most useful for our purposes. By ensuring the introduction of symmetry does not compromise the illusion [Cor70], it hints the achievement of such brightness illusions can be obtained through simple computational means such as isotropic filtering.

Andō Hiroshige

White Heron and Irises, 1833. View of the Tenryū River, 1832-34. Sparrows, Moon and Blossoming Peach, 1832-34.



Keinen
The Autumn Moon, *ca.* 1900.

Katsushika Hokusai,
Illustration of a Dragon from the *Manga*, 1815.

6.20 Contrast Illusions in Eastern Art. Surprisingly absent from most Western art form, the marked use of inks in Eastern art, and especially in *ukiyo-e* favored artists to master the illusory contrasts elicited by edge-like discontinuities. All illustrations presented above show the power of the Craik effect, along with the mastery of Japanese artists who achieve stunning brightness effects to circumvent the limitation of their medium.

Victor Hugo
Souvenirs d'Anvers, circa 1852.

6.21 The Craik Illusion as a Play of Ink. Ink lends itself quite easily to the achievement of the subtle contrasts eliciting illusory brightness. More commonly encountered in Eastern art, this drawing by French writer Victor Hugo nevertheless exhibits similar effects, especially visible at the outlines of the cloud that appears significantly brighter than the foggy sky. While Eastern artists used brushes carrying an uneven load of ink (*cf.* Figure 6.20), Hugo resorted ro stencils to produce comparable effects in his brown ink wash drawings.

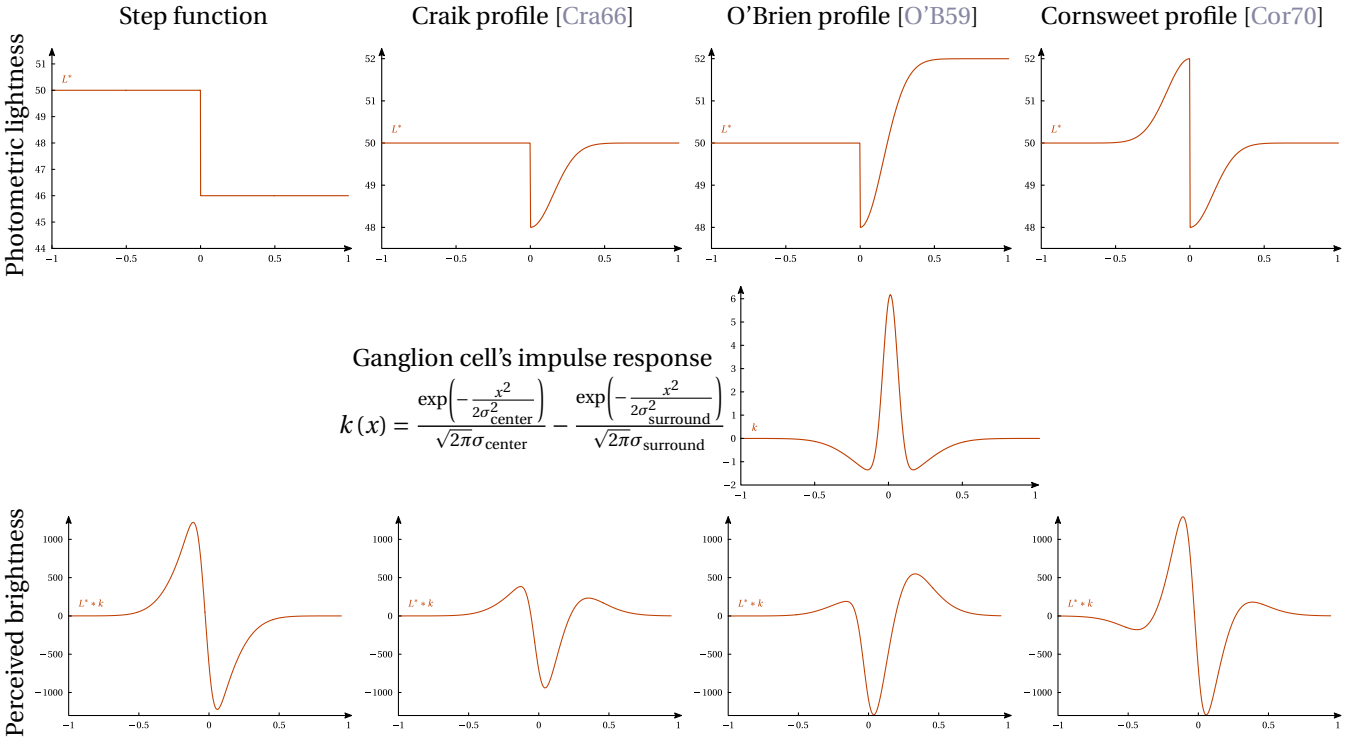
Illusory Brightness in Art

While not as abounding as global contrasts for lighting, or local contrasts for contour enhancement, such local intensity distributions are still observed in some pieces of art [Rat85], most of them being the work of Eastern artists (*cf.* Figure 6.20). While the Eastern sense of aesthetics which emphasizes stylization and achieves beauty from simplicity partly explains this fact, the medium they used also explains this fact. Indeed, ink is especially suitable for achieving the subtle gradients necessary for such illusory brightness to appear, either by the use of carefully ink-loaded brushes (*cf.* Figure 6.20), or stencils 6.21. But the gradual intensity return to its global value does not have much requirement in terms of continuity to elicit compelling contrast illusions: the stippled gradients shown in Figure 6.22 are also effective and take advantage of our visual system's spatial integration ability.

Korean storage jar
Yi dynasty.

Kay Chorao,
The Witch's Egg, 1974.

6.22 Stippling the Craik Effect. These two real-life illustrations of the Craik effect reported by late biophysicist Floyd Ratliff [Rat85] have in common the fact that the exponential return to the average luminance after the contrast fall-off is not continuous, but stippled. Even if the patches of the cobalt-based glaze on the surface of the vase slightly blend during firing, individual brush strokes are still discernable, and yet do not hamper the illusion to take effect.

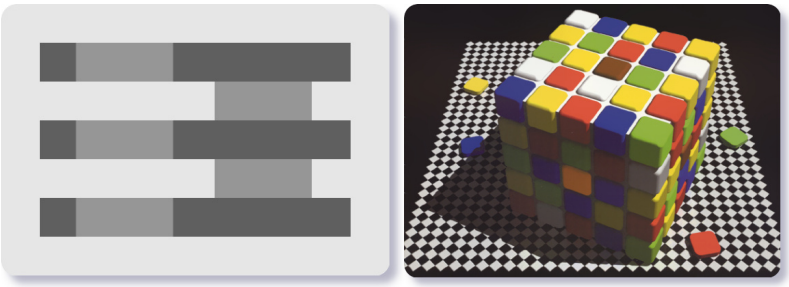


6.23 Different Effects for a Common Percept. Cells from the retinal ganglionic layer are highly responsive to abrupt spatiotemporal changes and their associated impulse response function can be computationally approximated by a difference-of-gaussians *(middle row)*. After convolution by this "filter", we can see that all luminance profiles *(top row)* end up sharing roughly the same neural response *(bottom row)*. While this partly explains why the Craik-O'Brien-Cornsweet effects belong to an equivalence class of stimuli, it does not clarify why they all specifically appear as step-like discontinuities.

Understanding the Craik-O'Brien-Cornsweet Illusions

The mathematical representation of the retinal ganglion cells' signal processing proves to be quite opportune as it provides a convenient explanation for the fact all three kinds of stimuli give rise to the same kind of illusory percepts. Figure 6.23 demonstrates that the approximate neural responses caused by all three signals are clearly comparable, rationalizing why in spite of their effective differences, they all belong to a same family of stimuli. Still, this computational formulation does not elucidate the reason why they all share the specific appearance of a rectilinear change separating regions of constant intensity.

While our technique pertains to the family of methods manipulating illusory brightness via luminance ratios across contrasts, alternative explanations by empirical evidence for these visual phenomena have also been proposed [WMP98, LP99, PSL99, LP00, PWNL04], most of which explain such entoptic effects by our visual system's *prior knowledge*. The predictions by luminance ratio of the percepts caused by the antagonistic interactions found in the retinal ganglion cells' receptive fields have notably been contradicted by numerous counter-examples (*cf.* Figure 6.24 *(left)*).



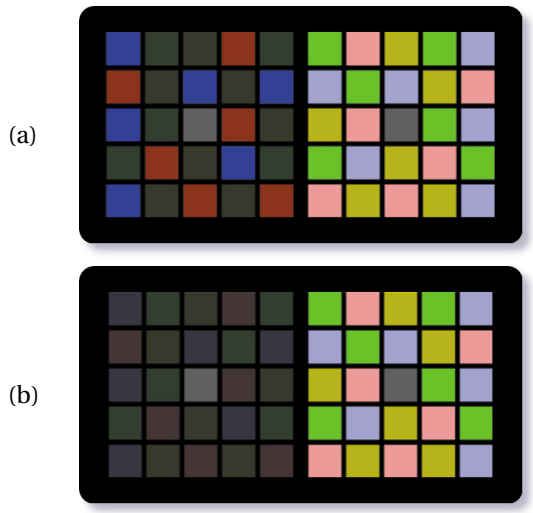
6.24. More Than Luminance Ratios. The distorted sensitivity caused by the specific response of the retinal ganglion cells provides an elegant explanation for the simultaneous brightness contrast phenomenon in most cases. However, subsequent research exhibited stimuli causing comparable surround-driven brightness deviations unpredictable using luminance ratios across contrast boundaries. White's illusion *(left)* notably shows perceptually brighter left targets although they are mostly in direct contact to areas of higher luminance. Lotto and Purves proposed an alternative "empirical" explanation, stating that the apparent brightness is mostly a "manifestation of its most likely provenance" granting a crucial role to its past knowledge [LP99]. They back up their assertion with a set of agreeing experiments showing the determining role of illumination related biases such as the presumption of shadowing or the direction of incoming light *(right)*.

Going back to the source of all visual stimuli, Purves *et al.* consider the two crucial factors conditioning the amount of light reflected by any scene to the eye: the reflectances of its surfaces, and the illumination under which they are observed. Two equiluminant returns are by nature ambiguous, and according to Purves *et al.*, the sole information our visual system can exploit in such cases lies in its past experience and its knowledge of the relative frequency of the possible causes of the observed stimuli. They back up their probabilistic framework hypothesis with experiments consisting in having test brightness match theoretically equiluminant targets whose difference lies in their surroundings. They study the effect of carefully chosen backgrounds over the illusion's strength by considering the magnitude of the observers' adjustments. Their claim is that our perception is mostly driven by the information from the stimulus that allows our visual system to makes "unconscious inferences" about how the stimulus is likely to have been generated. They demonstrate that simultaneous contrast is all the stronger when surrounds concord and hint the fact that the equiluminant targets are surfaces sharing a same reflectance but observed under different illuminations (*i.e.* when one surround gives the impression to be the shadowed version of the other). Simultaneous contrast may therefore have connections with our visual system's brightness and color consistency mechanisms (*cf.* Figure 6.25). The natural assumption according to which illumination comes from above also has a quantifiable impact on the observers' matches as it implies that objects' upper surfaces are better lit.

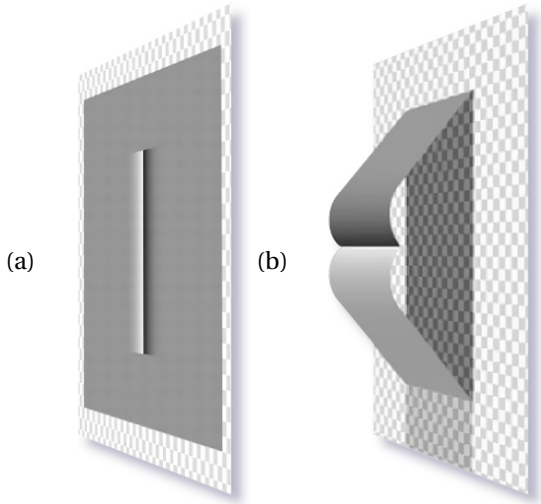
In Purves *et al.*' conception of our visual perception by inference, the Helmholtz-Kohlrausch phenomenon finds its explanation by the empirical significance of a saturated surface which reveals the fact it is well-lit, and therefore appearing brighter than it actually is. They also consider the case of the Cornsweet edge, and identify its potential source in a scene either as a gradual change in the observed surface's reflectance properties, either as a difference of illumination between the two territorial areas adjoining the edge (*cf.* Figure 6.26). Again, differences in the amplitudes of the observer-provided correction factors disclose the influence of visual hints biasing our interpretation of the scene over the illusory brightness it elicits. While hints of the Cornsweet edge being the manifestation of a local change in the surface's reflectance destroys the illusion, clues in favor of the hypothesis according to which the edge coincides with the demarcation between two differently lit surfaces strengthens it.

6.2.2 Restoring Lost Contrasts through Illusory Brightness

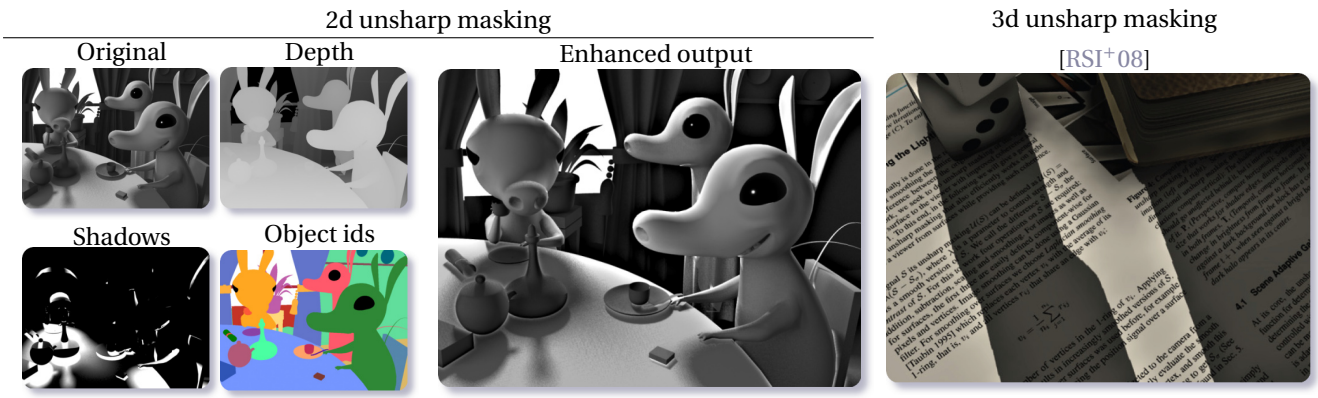
If we consider our conversion objective at the light of these new considerations, we realize that our intermediary results' most severe weaknesses can conveniently be dealt using contrast illusions. Indeed, the most aggravating circumstances where our global mapping fails to prove satisfactory correspond to cases of *directly adjacent* colors of almost equivalent *apparent* brightness. While the lack of resulting grey-scale difference is not that problematical for remotely located colors, our visual system's sensitivity to contrasts resulting from conflicting brightness values or colors forces us to reintroduce the lost chromatic contrasts for our results to be perceptually compelling with respect to their respective input. While Nayatani's brightness predictor ensures the assignment of perceptually convincing grey-values, the independent processing of pixels prevents



6.25 Simultaneous Contrast and Color Consistency. Lotto and Purves demonstrate the magnitude of the simultaneous contrast effect is linked to our unconscious *understanding* of the scene. Contrary to the surrounds in *(a)*, the equiluminant squares of *(b)* concur with the hypothesis that the left arrangement is a shadowed version of the right one, therefore triggering mechanisms close to color consistency and reinforcing the illusory brightness adorning the left target.



6.26 Cornsweet Edges as Changes in Reflectance or Illumination. Whenever clues indicate a Cornsweet edge arises from the graded difference in reflectance, the illusory brightness value assignment vanishes *(a)*. Conversely, evidences of it being caused by two differently lit surfaces of distinct reflectances (shape cues, depth sensation by perspective, cast shadows) considerably reinforces its effect *(b)*.

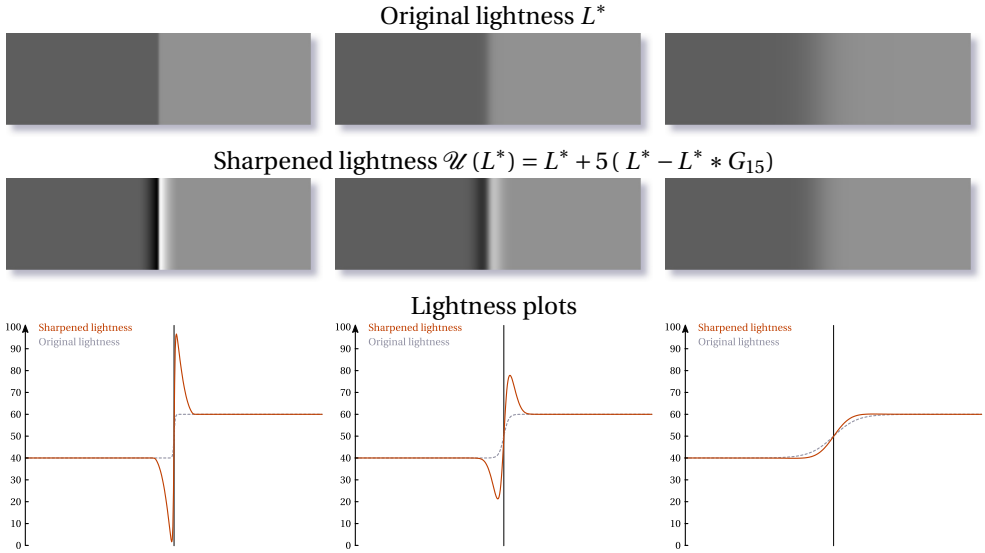


6.27 The Versatility of the Unsharp Masking Filter. Illustrated here some of the many uses of unsharp masking for shape enhancement and help for scene understanding. With Kaleigh Smith, our first take on this operator was primarily motivated by the will to propose a unified framework that would encompass many unsharp-based scene enhancement related techniques [CST05, LCD06]. But the reliance on 2d buffers containing the information to enhance caused artifacts such as halos and conflicting gradient polarities. In subsequent work, Smith along with Tobias Ritschel lifted these inconveniences by directly performing unsharp masking over the lighting function in 3d [RSI+08].

the capture of higher-order effects such as contrasts. But our natural tendency to spread illusory brightness from local intensity discontinuities may enable us to have our grey-scale results nevertheless depict the input's color contrasts, and unsharp masking thanks to its ability to subtly boost contrasts along thin discontinuities (*cf.* Figure 6.27) appears as the perfect pick for the task.

Unsharp Masking

Originally an image acutance improving technique from analog photography, unsharp masking has now become a popular digital image processing technique that is commonly used to grant a higher quality, sharper



6.28 The Effect of Unsharp Masking over Different Kinds of Edge. Depending on the sharpness of the original edge, unsharp masking is more of less pronounced: the steeper the edge's slope, the higher and closer to its central axis the enhancement peaks. While its effect over a step edge resembles a boosted version of a Cornsweet contour with two directly adjacent peaks of opposite polarity *(left)*, its aspect over smoother edges looks more like an explicit Mach band *(middle)*. Another appealing aspect of unsharp masking is its visual unobtrusiveness when the width of the edge becomes smaller than the scale of the filter's blurring operator *(right)*.

look to an image $\mathscr{I}$ by way of combination with a blurred version of it, denoted low $(\mathscr{I})$:

$$\mathscr{U}(\mathscr{I}) = \mathscr{I} + \lambda(\mathscr{I} - \text{low}(\mathscr{I})). \tag{6.15}$$

Unsharp masking straightforwardly emphasizes the image's finest features via the weighted addition to the original image of the enhancement signal yielded by the subtraction of a blurred version of the image to it. This signal hence conveys image features of spatial frequency exceeding the low-passed ones. Its addition to the original image then locally amplifies the magnitude of its first derivatives and strengthens sharp tonal discontinuities, making details naturally stand out. While its strength is controlled by the contribution λ attached to the sharpening signal, the enhancement signal's spatial extent and visual impact is directly dependent on the blurring step: if blur involves the averaging over smaller neighborhoods, edge sharpness increases and very locally alters contrast, yielding crisp results without a noticeable alteration of the image's range. On the other hand, as blur grows wider, global contrasts are enhanced as the conflicting contrast polarities across edges propagate deeper into their surrounding regions.

Aesthetical considerations left aside, the visual impact of unsharp masking directly resonates with our vision's processes in charge of the formation of our brightness and contrast percepts (*cf.* Figure 6.28):

- In the case of sharp edges segregating regions of strongly distinct intensities, unsharp masking further stresses this disparity by lightening the clearer region while darkening the dimer one. Its visual impact is much reminiscent of the Mach bands which designate the small regions of illusory brightness often observed on both sides of steep edges. While pretending to directly manipulate such perceptual entities via image filtering would be a non-sense, this observation confirms the unobtrusive, if not pleasing effect of unsharp masking on images.
- For smoother edges, the Mach band analogy still holds to some extent, but the peaks of additional intensity progressively depart from the apex of the edge as its blurriness increases, thus continuously weakening its effect.

Reintroducing Discriminability using Color Contrast-Based Unsharp Masking

In our case, we aim at the reintroduction to our grey-scale output of lost chroma contrasts, and unsharp masking its lightness with an enhancement signal embedding this missing information satisfies our discriminability objective. As seen before, the computation of such signals boils down to the approximation of the image's smoothed 2nd derivatives by the subtraction of a low-passed version of the image to it, and can be generalized by a difference-of-gaussians filtering operation. Subtracting two differently blurred versions of the image acts as an edge detector over the image features whose spatial frequency lies between the range of frequencies preserved by the blurring passes. Beside its intuitive multi-scale extension, this modeling provides us with a natural blur control parameter via the standard deviation of the initiating gaussian filter.

But more importantly, the convolution with a difference-of-gaussians makes the shape of the enhancing signals strikingly close to the mathematical expression of the visual stimuli associated with the Cornsweet illusion: a sudden intensity change flanked with two exponential returns of opposite sign to a common intensity value. Henceforth, by having the sharpening signal encode the input's *chroma* edges, we can enforce them back to our globally mapped grey-image via the introduction of Cornsweet contours where colors of a same apparent brightness meet in the original image. The approximation of the image's second derivatives by the convolution with a bank of difference-of-gaussian filters directly refers to the finite impulse response of the retina's ganglion cells in charge of our pre-attentive perception of contrasts. We claim that having our contrast enhancement rely on a corrective signal close to our perceptual response to the original image is to best guarantee our results' perceptual accuracy.

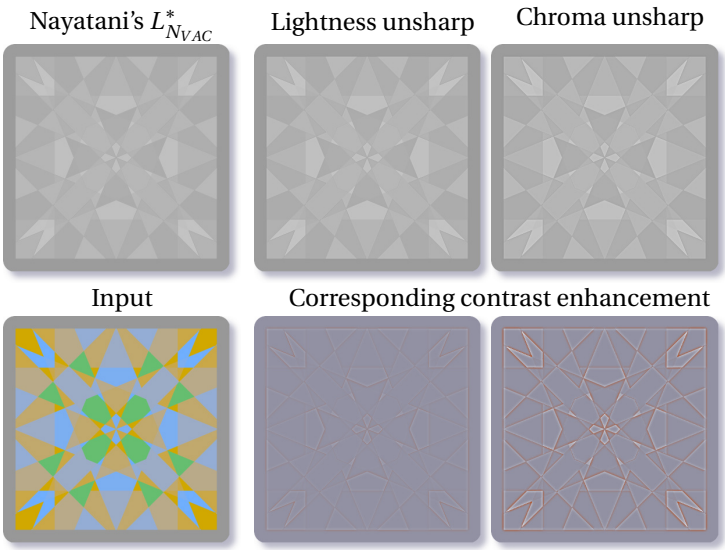
The appeal of such a filter-centric approach for handling discriminability is two-fold. Beside the simplicity of the involved processes, the linear nature of their underlying filters blazes the trail for improved performances and a straightforward handling of temporal coherence. Then, the locality of the changes applied to the globally mapped grey-scale image facilitates the preservation of its perceptual accuracy which is the prominent feature of our approach. Bearing in mind this initial objective, our local discriminability restorative step still requires some attention. Of most important considerations, the hue consistent lightness ordering and the relevant polarity of gradients established by the $L_{N_{VAC}}$ predictor (*cf.* Figure 6.29) must be preserved by our enhancement.

6.29. Versatility of Gradients. Thanks to Nayatani's perceptually relevant brightness mapping, the gradients straightforwardly computed over our globally mapped grey-scale image are as adequate *(right)*, whereas the estimation of gradients for multivariate data is nowhere as easy as illustrated by Neumann's result *(middle)*. In spite of the care they deploy for perceptual accuracy, the reintegration of their corrected color difference field produces here an almost inverted result.

Following Burt and Adelson's algorithm [BA83], we build the Laplacian pyramids out of the series of gaussian planes obtained through the repeated blurring of the original image $\mathscr{I}$ and our intermediary grey-scale image $\mathscr{G}$. Each level of such pyramids therefore contains the amount of (color or brightness) contrast at a specific scale. Although low-pass filtering needs to be performed in linear additive *RGB* space[7], contrast information arising from the difference between successive pyramid levels is to be evaluated in a perceptually uniform color space. We therefore evaluate color contrasts on a per-pixel basis as the CIE $L^*a^*b^*$ Euclidean distance between a pixel's color before and after gaussian filtering[8]. Our chroma-driven multi-scale unsharp

[7] Discrete convolutions involving the weighted local averaging of colors.

[8] Unline Nayatani, we choose the CIE $L^*a^*b^*$ color specification to assess chromatic contrasts over CIE $L^*u^*v^*$ for its better integration of the color consistency phenomenon.



6.30 Sharpening Chroma Only. For our discriminability objective, special care must be provided to the weighting of the sharpening signal. Blindly running an unsharp filter over the globally mapped grey-scale image fails to recover original contrasts and can only emphasize the ones that survived the $L^*_{N_{VAC}}$ mapping *(center)*. By weighting brightness high frequencies using local chromatic differences from the color input, we can make our apparent grey-scale output exhibit the same contrasts as the original image *(right)*.

masking can thus be summarized as:

$$\mathscr{U}(\mathscr{G})_{L^*} = \mathscr{G}_{L^*} + \sum_i k_i \lambda_i \text{high}_i(\mathscr{G})_{L^*} \quad \text{where} \begin{cases} \text{low}_0(\mathscr{G}) = \mathscr{G}, \\ \forall i > 0, \quad \text{low}_i(\mathscr{G}) = \text{low}_{i-1}(\mathscr{G}) * G_\sigma, \\ \qquad \text{high}_i(\mathscr{G}) = \text{low}_{i-1}(\mathscr{G}) - \text{low}_i(\mathscr{G}). \end{cases} \tag{6.16}$$

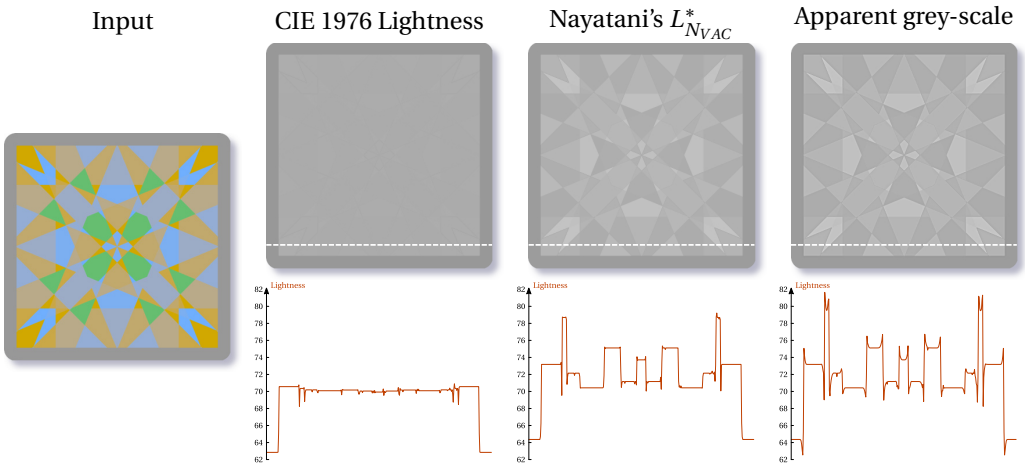
The L^* subscript indicates that all operations only involve the grey-scale image $\mathscr{G}$'s lightness, leaving its bland chroma-related channels untouched. The high frequency content is iteratively isolated through the application of the $\text{high}_i(.)$ operator consisting in the subtraction of two successively blurred versions of the image obtained by the convolution with a gaussian function of standard deviation σ, G_σ.

The different high-frequency bands from the pyramid are adaptively weighted by the (λ_i) factors whose role is to make the achromatic contrasts from $\text{high}_i(\mathscr{G})$ best match the color contrasts at their corresponding scale, $\text{high}_i(\mathscr{I})$. At each pyramid level, the λ_i weighting function is evaluated on a per-pixel basis and reflects the amount of missing contrasts from the grey-scale image when considered with respect to the original color input. Ideally, it should prevent overshooting of already present contrasts and behave continuously across scales and image locations for our Laplacian pyramid's enhancing signal to elicit compelling Cornsweet illusions. Expressed as the ratio of the color contrasts from $\mathscr{I}$ over the already available lightness contrasts in $\mathscr{G}$ at the ith level of the pyramid, it is computed as:

$$\lambda_i = \left(\frac{\sqrt{\text{high}_i\left(L^*_{\mathscr{I}}\right)^2 + \text{high}_i\left(a^*_{\mathscr{I}}\right)^2 + \text{high}_i\left(b^*_{\mathscr{I}}\right)^2}}{\left|\text{high}_i\left(L^*_{\mathscr{G}}\right)\right|} \right)^p. \tag{6.17}$$

Comparisons with unsharp masking directly performed over the grey-scale image $\mathscr{G}$ clearly demonstrates the relevance of this formulation (*cf.* Figure 6.30).

Following our initial thread of thought, we expect this corrective signal to enhance the output's brightness contrasts in accordance with the input's missing color contrasts, and introduce illusory brightness via Cornsweet contours when existing contrasts are initially too mitigated (*cf.* Figure 6.31). But for the illusion to be effective, we give the control back to users through a minimal set of as relevant as possible parameters. While fully automatic up to this point, our technique must have the final integration of its local contrast enhancement manhandled since subtle stimuli such as Cornsweet edges require its adjustment depending on the displaying devices' resolution or range. The set of (k_i) parameters (*cf.* Equation 6.16), as well as the p exponentiation factor (*cf.* Equation 6.17) serve that purpose. The (k_i) factors control the spatial extent of the introduced contrasts by enabling users to give an uneven emphasis on the different pyramid levels. Their influence over edge sharpness and territorial contrast enhancement is shown in Figure 6.32. The p parameter directly



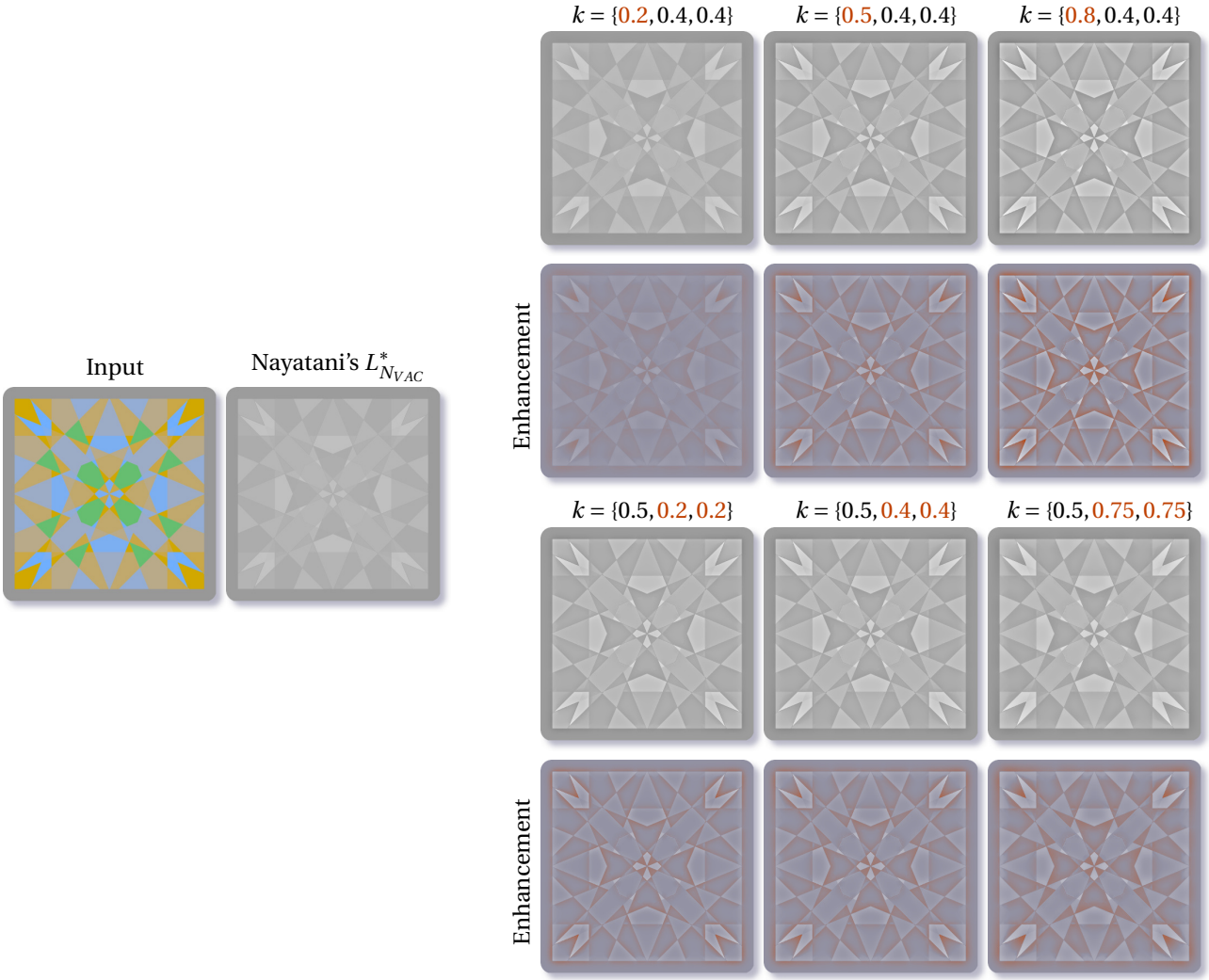
6.31 Improving Discriminability by Enhancing Contrasts. The plots displayed above show lightness graphes established over the same row *(white dashed line)* of different grey-scale versions of the input presented on the left. As evidenced by the CIE 1976 lightness plot, this test image's colors are all close to perfect equiluminance. We observe that our global mapping to apparent brightness lifts this ambiguity to some extent by accounting for the chroma-dependent Helmholtz-Kohlrausch effect. By additionally performing color-driven unsharp masking, we slightly perturb the lightness local profiles and make them resemble the theoretical profiles of a Cornsweet edge, eliciting illusory brightness in the process.

affects the computation of the added contrasts. It is a scalar value in the $[0,1]$ interval mapping ratios between color and lightness contrasts to a non-linear scale that can serve to account for weaker chromatic contrasts while preventing the overshooting of existing lightness contrasts. Contrary to the previous set of parameters, the tuning of p is more dependent on the content of the converted image than its displaying conditions. If the input exhibits subtle, purely chromatic contrasts, resorting to higher values of p can prove to be very beneficial. Its effect is illustrated over one of Claude Monet's most famous painting *Impression soleil levant* which draws its fame from Monet's unexpected and witty use of such chromatic contrasts (*cf.* Figure 6.33), and over a real-life photograph which demonstrates the applicability of our contrast enhancement to complex imagery (*cf.* Figure 6.34). Lastly, the entirety of our conversion approach is graphically summarized in Figure 6.35.

6.3 Additional Results and Discussion

Current section presents additional results of our approach and also discusses its intrinsic limitations. Prior any of these, recall our primary objective is perceptual relevance. Therefore, introducing undue distortions to our inputs' overall brightness distributions for emphasizing discrimination between mapped grey values is avoided at all costs. The price to pay for our choice is the fact our introduced modifications with respect to a standard mapping to luminance may appear pretty subtle for the inattentive viewer, especially in comparison with the severe changes caused by optimization-based approaches. But while not as drastic, our enhancement is present and despite its subtlety, yields satisfactory outputs in terms of accuracy and user preference. All of the subsequently presented results (*cf.* Figures 6.36, 6.37 and 6.38) assume the provided inputs are specified in the device-dependent *sRGB* color space and are meant to be viewed on a calibrated monitor, or printed once mapped on the appropriate printer gamut.

Performances One of the most easily quantifiable virtue of our technique is its computational simplicity which ensures improved performance. Its linear complexity makes it hardly comparable to the extremely expensive optimization-based conversions which have to resort to color sampling or quantization to run at acceptable rates. If any, the bottleneck of our technique is the multi-scale decomposition of the image for the computation of the enhancing contrast signal which ekes out computations a bit. However, none of the results presented here required more than a couple of seconds to be generated. In addition, all involved oper-

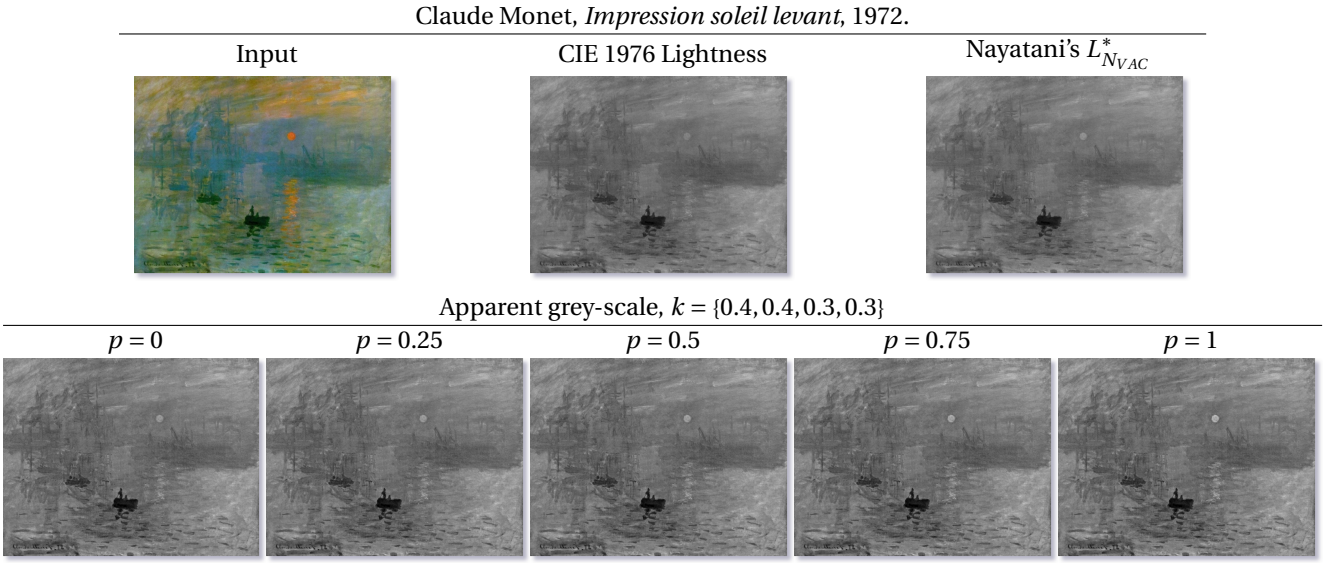


6.32 Controlling the Sharpening Signal's Shape. The set of parameters (k_i) provides the user with intuitive handles for altering the enhancing signal's profile. Since each of them is associated with a frequency band, they notably allow to give variable emphasis on specific bands and directly govern the spatial extent of the chroma unsharp masking. The first set of images shows the effect of tuning the high frequency contributions of the sharpening signal, therefore modifying fine details and greatly improving the introduced Cornsweet illusions. Conversely, altering its lower frequency components affects the spatial extent of the enhancement which goes further into the territorial areas flanking the chroma edges as demonstrated on the second set of results.

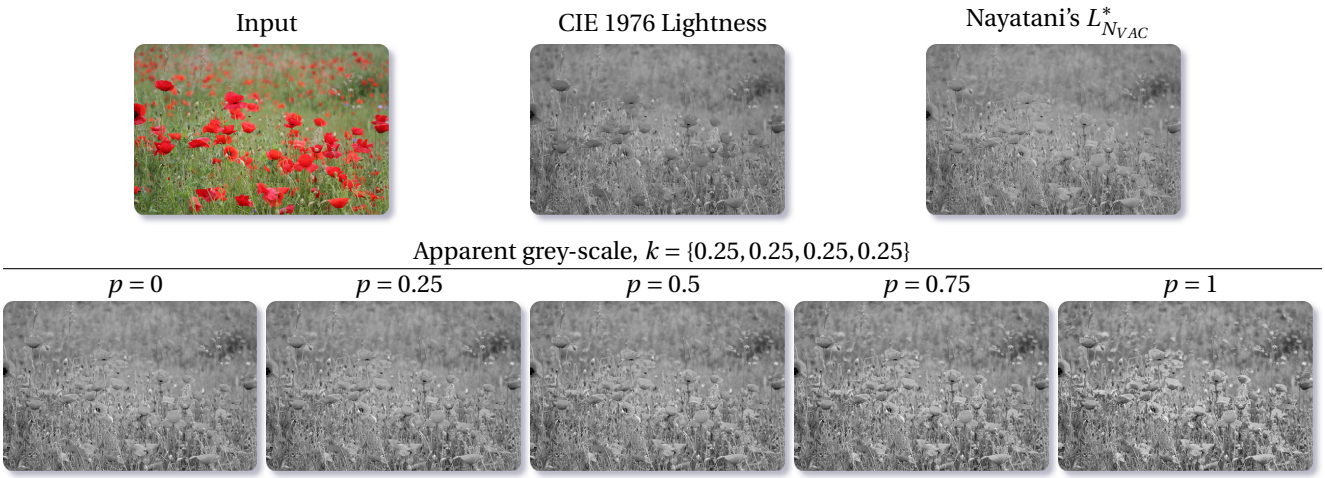
ations from the global mapping to our multi-scale sharpening translate fairly well to programmable graphics hardware architectures.

Robust Lightness Ordering Achieving a proper and consistent assignment of grey values across scenes and parameter changes is a crucial aspect and our method ensures a perceptually correct and consistent ordering of lightness. This is shown in Figure 6.39, our method achieving the most satisfactory results in terms of accuracy. The dynamic range of our results is not overstated for artificially increasing discrimination and stays constant across applications. This feature of our method, which is an indirect consequence of our foremost strive for perceptual accuracy, constitutes a key advantage for its extension to videos as detailed below.

Coherence over Temporally-Varying Palettes Prior our take on grey-scale conversion, no previous method even considered the treatment of animated inputs. This is partly explained by the algorithmic complexity and slow performances of optimization-based techniques. But the main reason is that since they mostly aim at increased discrimination, they often propose heavily data-dependent mappings. This aspect directly compro-

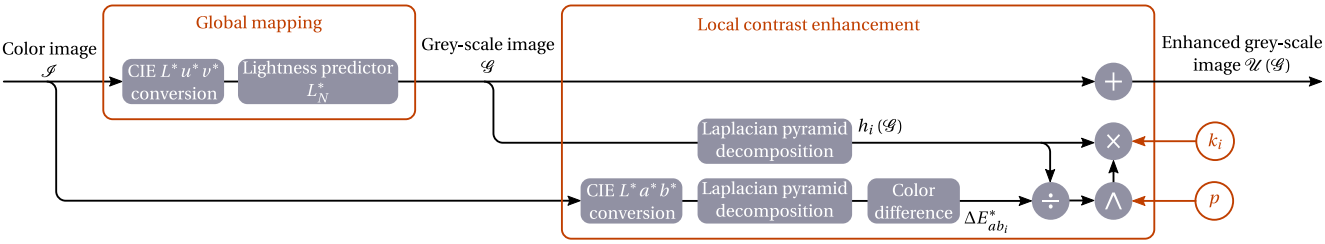


6.33 Boosting Chroma Unsharp Masking. Parameter p is a scalar ranging from 0 to 1 that maps the chroma-driven enhancement to a non-linear scale used to control the emphasis given to the sharpening signal's chromatic component. The higher its value, the steeper the contribution of the chroma-related contrasts is with respect to lightness-dependent contrasts, as illustrated here by the increasingly bright sun glow and its reflections on the water. Note that edges already reflected by the lightness component alone (such as the boat's outlines) are left untouched and do not suffer from undue overshooting.



6.34 Boosting Chroma Unsharp Masking (Continued). Amusingly enough, this photograph of a field of poppies that amusingly enough, exhibits the very same kind of equiluminant color contrasts as the ones impressionist artists were so fond of. Amplification of noise already present in the image only appears for the highest values of p and in spite of that, results are perceptually compelling. Notice the determining role of the integration of the Helmholtz-Kohlrausch phenomenon using Nayatani's predictor that recognizes the flowers' petals as brighter as the grassy background.

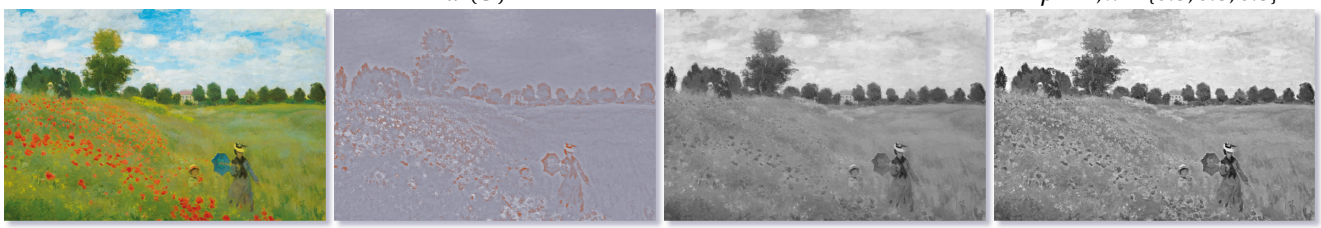
mises the possibility of independently applying such procedures to the frames of a video: without additional processing for the incorporation of the temporal color contrasts appearing at a same pixel location along the course of the video to their objective function, their mapping is likely not to follow a smooth or even continuous evolution. This limitation of most earlier approaches is illustrated on a simple vector-based animation presented in Figure 6.40. On the other hand, our decomposition of the grey-scale conversion problem into two independent steps is especially useful for straightforwardly dealing with such concerns. For the reasons stated in the previous paragraph, our global grey value assignment can naturally be performed on the frames of the animation thanks to its constant, data-independent nature. Our subsequent local contrast enhancement, may be data-dependent but thanks to the linear nature of its filtering operations and locality of its effects, does not



6.35 Overview of our Apparent Grey-scale Algorithm. Our technique proposes a two-stage approach for handling grey-scale conversion, this choice being the direct consequence of our aim at perceptual faithfulness. Global color ordering in terms of perceived brightness precedes local chromatic contrast enhancement for improving discriminability. Only the later stage can be controlled by the user by way of the parameters p and $(k_i)_i$ displayed in red circles.

Claude Monet, *Coquelicots à Argenteuil*, 1973.

Input	Signed differences: $\mathscr{U}(\mathscr{G}) - L^*$	CIE 1976 lightness L^*	Apparent grey-scale $\mathscr{U}(\mathscr{G})$, $p = 1, k = \{0.3, 0.3, 0.3\}$

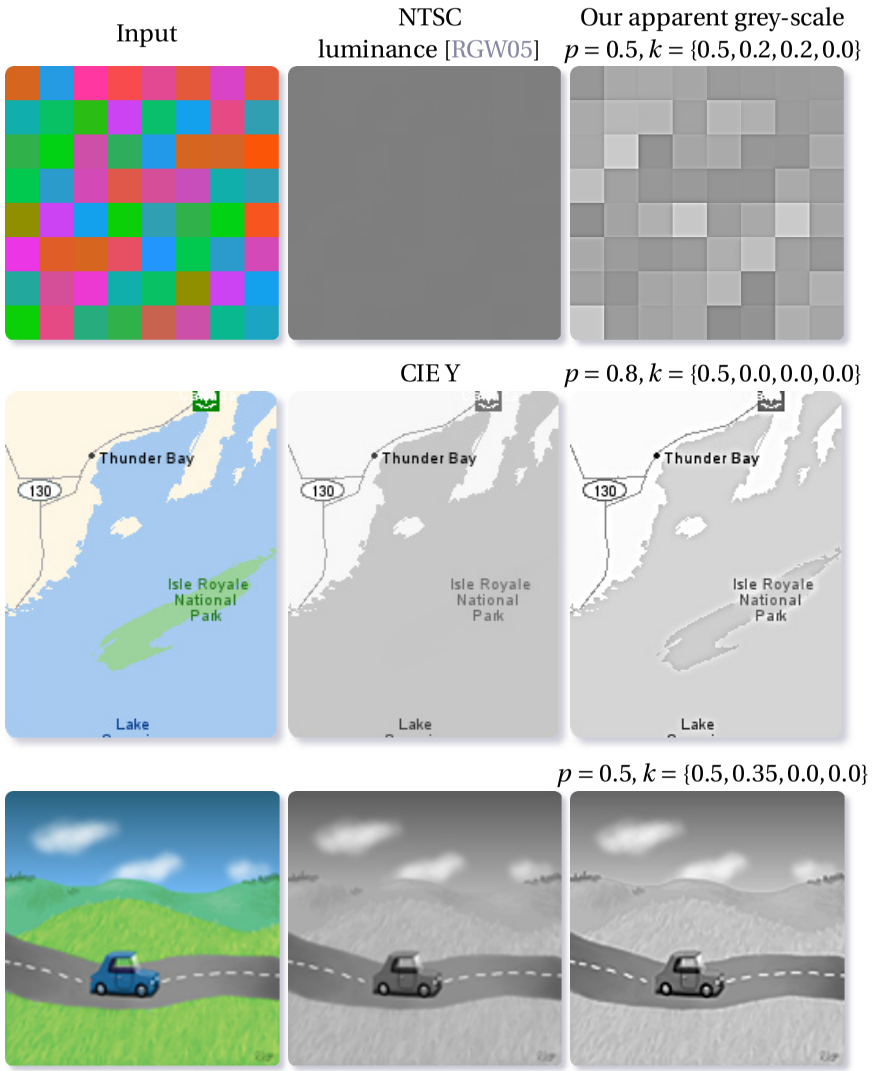


6.36 Restoring Color Contrasts. Paintings from the Impressionist movement are particularly challenging examples when it comes to their determining their achromatic rendition as their equiluminant colors were purposefully chosen for the illusory motions they create. Here, the poppies' red matches in luminance the fields' green *(top left)* and vanishes if the pixel colors are replaced by their respective lightness *(bottom left)*. Our apparent grey-scale approach by accounting for both perceptual phenomena of chroma-driven brightness and chromatic contrasts manages to make the flowers stand out again without distorting the painting's original overall appearance *(bottom right)*. The sky blue is also brightened and is much more faithful to the perception we have of Monet's work than standard lightness.

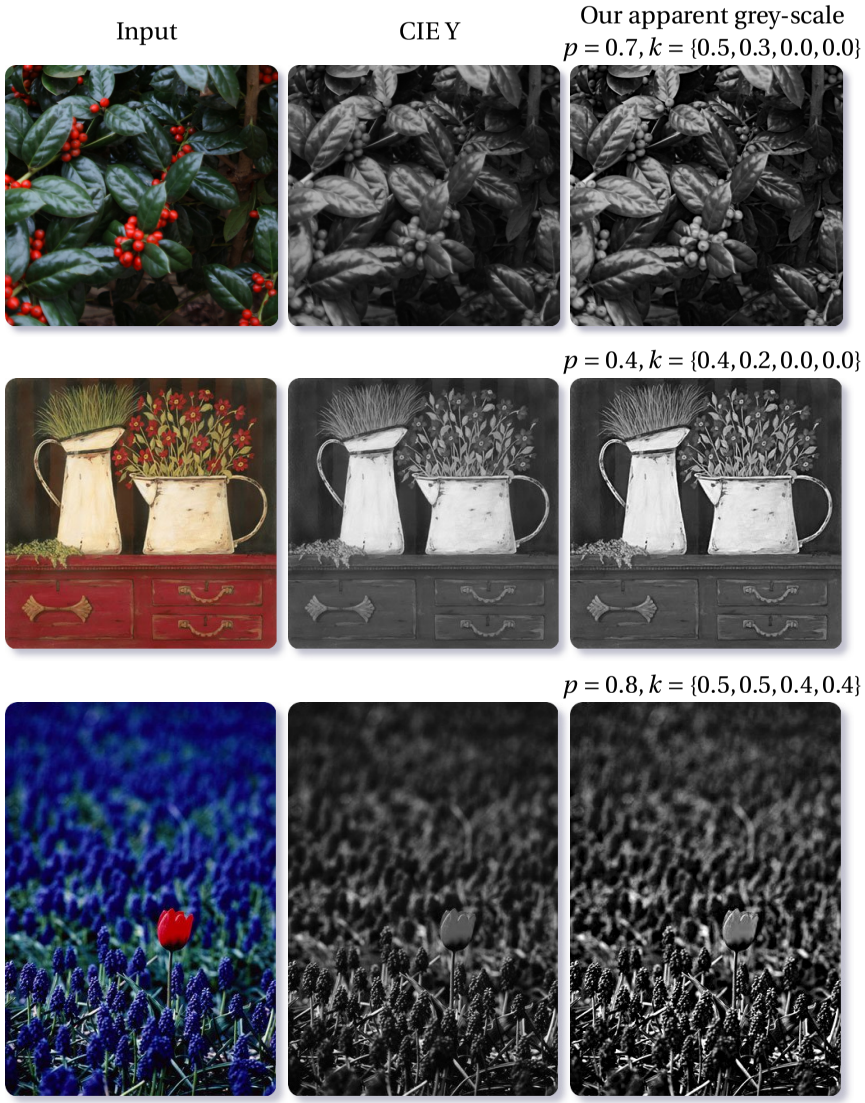
introduce any disturbing artifacts when appropriately tuned for the considered video (*cf.* Figure 6.41).

Detail Preservation Thanks to the computational simplicity of our conversion algorithm, our method does not invoke any data simplifying steps on the input's color distribution such as color range discretization [GOTG05, RGW05] or sparse color sampling [GD07]. It therefore treats the input's as an untouched whole and is not to miss the influence of any present color over the output's perceived brightness, nor any local color contrasts. Additionally, unlike Gooch's and Neumann's grey-scale converters, the core of our technique does not evolve in the image gradient domain and therefore prevents the apparition of artifacts such as the loss of spatial precision or gradient polarity flips when the output is constructed from the input's color contrasts. Indeed, though robust, Gooch's global optimization sometimes tends to alter the input's spatial content as evidenced in her achromatic rendition of Monet's *Impression: soleil levant* (*cf.* Figure 6.42) where the sun and its reflection on the water surface appear dilated and the overall result exhibits an excessive level of blur hiding most of individual brush strokes. Similarly, Neumann's iterative correction of the input's gradient field sure enables the obtention of the final grey-scale outputs by way of a simple 2d integration (therefore considerably accelerating its reconstruction in comparison to Gooch's and Rasche's iterative optimization), but this process seems quite unpredictable as it sometimes propagates unexpected gradient polarities through the field.

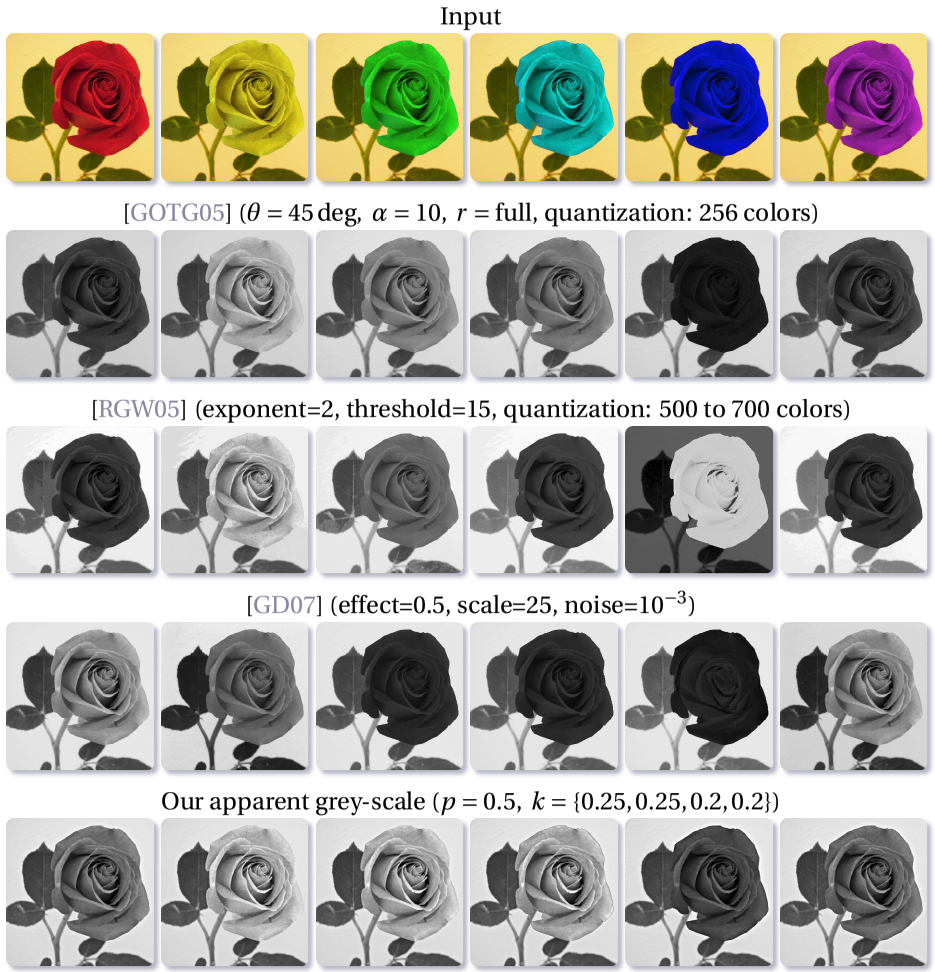
Noise Amplification by Sharpening Alas, our method is not free from artifacts, most of them being generated at the local enhancement step. Recall that it aims at the reintroduction of lost chromatic contrasts by adaptively adding back the lightness high-frequencies. For that aim, we rely on unsharp masking which provides a convenient and elegant way of introducing Cornsweet edges. However, our technique also directly inherits



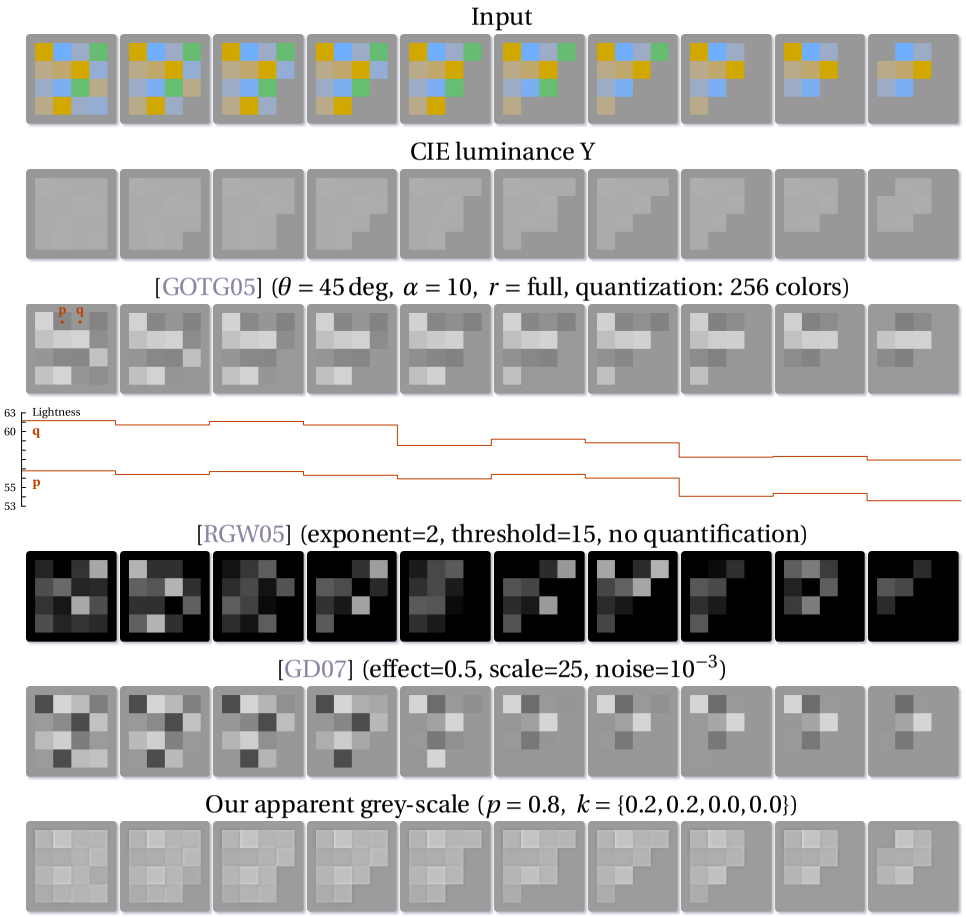
6.37 Our Achromatic Renditions of Images with Equiluminant Colors. In such pathological cases, the Cornsweet edges added by our contrast enhancement constitute essential visual clues for discrimination. The induced illusory brightness values are perceptually in concordance with the gradient polarity imposed by Nayatani's brightness predictor and markedly propagate inside the adjacent territorial regions. This is particularly appreciable on the map example where the island reappears.



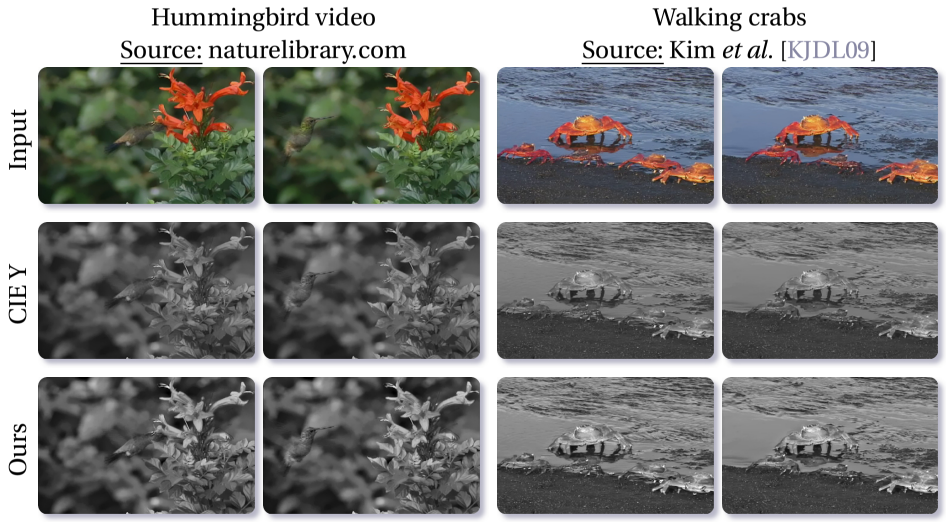
6.38 More Grey-Scale Results. These examples highlight the determining role of our global mapping step that incorporates the Helmholtz-Kohlrausch effect and accounts for the increased perceived brightness of saturated colors. Vivid colors appear brighter in our outputs as they would if directly mapped to their luminance. The berries, and vibrant red flowers stand out more from their respective duller backgrounds.



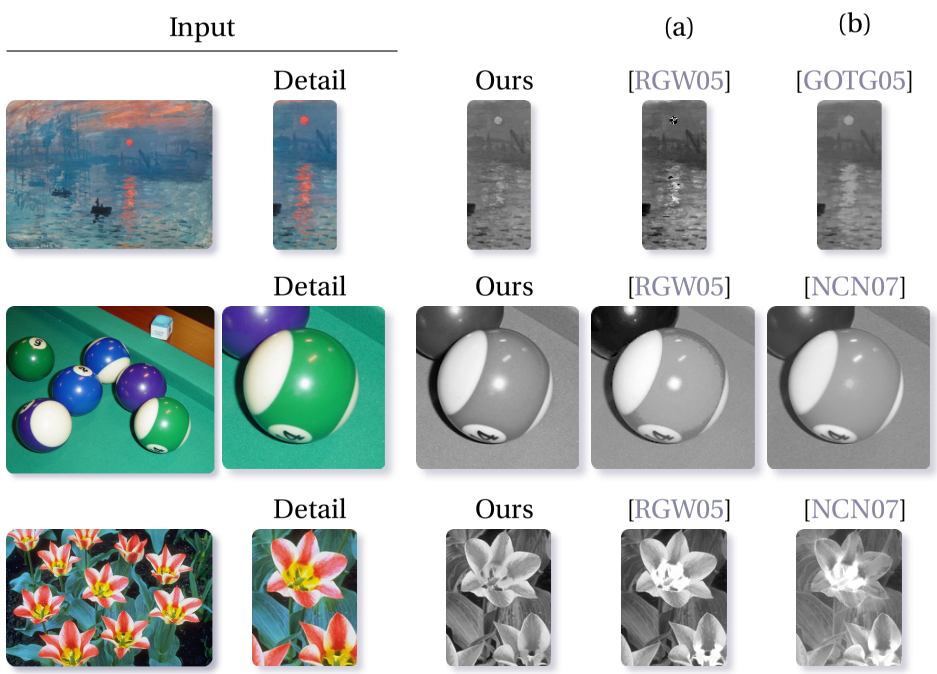
6.39 Lightness Ordering. Our conversion ensures a consistent handling of colors across applications and chromatic content, and the ordering of the assigned grey values is guaranteed without conflicts or sudden rearrangements. Here, only the rose's petals change colors, which is reflected by our inputs where its leaves and stem are left untouched. Gooch's *Color2Grey*, while proposing a more perceptually relevant grey value assignment than Rasche's or Grundland's, slightly alters the background's sharpness and intensity across the different results. Rasche's assignment is particularly dramatic, sometimes deviating significantly from the colors' luminance. But since targeting visually deficient people, his method can afford such lightness flips and range distortions for preserving the information conveyed by the input. Note that color quantization was needed for Rasche's algorithm to run at reasonable times.



6.40 From Perceptual Consistency to Temporal Coherence. The proposed animation consists in the progressive disappearance of the equiluminant color squares. The quality of the results is thus directly dependent on the handling of the color contrasts present in the input. Discrimination centric approaches cannot ensure the consistency of their grey value assignment as they rely on the input's color distribution either for the computation of their objective function [GOTG05, RGW05], either for the creation of an intermediary, data-driven color space where discrimination is best achieved [GD07]. While more mitigated in Gooch's results, a similar shimmering effect is observed in the sequence of outputs reconstructed by optimization as evidenced by the graphed luminance temporal profiles at pixel positions **p** and **q**. Our approach overcomes the challenge posed by temporal coherence insofar its parameters its parameters are kept constant. We also claim that our results' dynamic range best depicts the color input's.



6.41 Grey-Scale Conversion of Real-Life Videos. Since our method does not involve any specific sampling scheme either in the spatial dimension or color range, it is especially suited for the conversion of animated inputs. The vivid red flower gets additional brightness and the bird stands more sharply in front of the background *(left)*. Similarly, the shells of the walking crabs end up as brighter in our results for correctly accounting for their vibrant colors which contrast with the duller sea blue *(right)*. When watched in motion, our sharpening-based local contrast enhancement does not introduce any noticeable artifacts at the top half of the second video where ripples on the water surface along the perspective foreshortening introduce fast-moving high frequencies. All results were obtained with the following: $p = 0.8$ and $k = \{0.2, 0.8, 0.0, 0.0\}$.

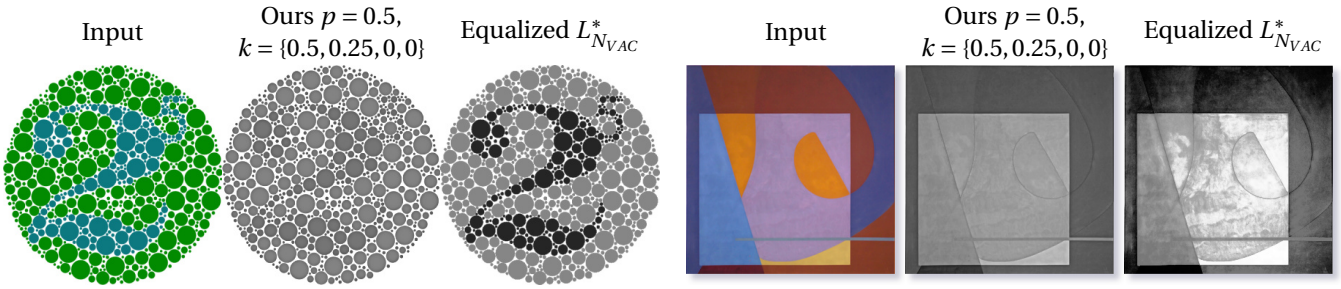


6.42 Preservation of Spatial Details. Another virtue of our method is that it preserves the input's fine spatial details. In contrast, existing methods tend not to be that respectful. Some require the simplification of its color distribution in order to decrease the number of contrast constraints to respect *(a)*, whereas gradient-domain techniques can introduce visible artifacts at the final reconstruction stage of their output *(b)*. Source: All results courtesy of Martin Čadík [Čad08a].

6.43 2d Versus 3d Unsharp Masking. By performing contrast enhancement directly in 3d object space, Ritschel *et al.*'s unsharp masking operator can have its parameters adapted to the scene geometry and viewing conditions. This improved consistency with respect to the scene greatly improves the quality of their results in comparison to the flatter looking 2d sharpened image and notably enables the prevention of the common 2d related artifacts that are halos *(a)* and overshooting in detailed regions *(b)*.

its weaknesses, the most aggravating ones being noise amplification (especially visible when the input image has undergone multiple compression passes for storage) and the creation of halos. By its formulation, unsharp masking provides two handles for its improvement: a better estimation of the image's high-frequencies, and an content-aware scaling factor. The concerns risen by noise reinforcement and edge overshooting come down to finding computational ways of appropriately distinguishing between noise and the edge content of an image. The easiest way to limit noise is to trade high-pass filtering used to compute the contrast signal for a band-pass filter, but this hinders the proper enhancement of most images. More elaborated attempts replace the linear Laplacian filter with non-linear filters embedding local image statistics, such as Lee and Park's improved Laplacian which considers the differences between local means and median values [LP90], or Mitra *et al.*'s generalization of Teager's algorithm which approximates a high-pass filter weighted by local means and accounts for our acute sensitivity to high-frequency variations in darker regions [MLLY91]. Quadratic filters and their approximations usually give more emphasis on brighter pixels and are thus perceptually more relevant than their linear counterparts at the light of Weber's law. Guillon *et al.* suggest the resort to non-linear quadratic filters that smoothly weight the neighboring pixels' contribution depending on their luminance proximity to the center pixel's [GBN96]. Ramponi advocates the use of a cubic non-linear operator locally modulated by the local image gradients to isolate edges from mere noise [RP98]. Long considered as too computationally expensive, the adaptive scaling of the correction signal has also been explored for incorporating local statistical measurements of the image's local "dynamics" and detail level. Polesel *et al.* consider local variances for treating highly- and moderately-detailed areas differently [PRM97]., and in subsequent work, Guillon *et al.* use a Lagrangian function evaluated over the local neighboring gradients to prevent the overshooting of hard edges and the enhancement of falsely detected details [GBNK98]. Already adaptive to chromatic contrasts, our method could obviously benefit from one of these alternatives for artifact prevention.

Creation of Halos Handling halos is more involving, as such artifacts get especially distracting when conflicting with our *understanding* of the scene's layout (at occlusion boundaries typically). While easily unnoticed in detailed, natural scenes, they become unnerving on simpler inputs exhibiting flatter looking areas (*cf.* Kim *et al.*'s cartoon animations [KJDL09]). Alas, with only having a single image at our disposal, we cannot possibly truly anticipate such artifacts. In recent research, Kaleigh Smith addresses these pitfalls by considering the extension of unsharp masking to the 3d case, and performing contrast enhancement on the *lighting function* which is the spatial signal of the light reflected by the mesh surfaces [RSI+08]. With the exact knowledge of the viewing conditions and the scene's geometry at hand, they can have the size of the kernel upon which the local details and associated strength are determined, account for depth discontinuities, or the distance to the viewer. Similarly to our method, they hence introduce Cornsweet edges to their renderings, and by enhancing illumination cues in perfect concordance with the scene's 3d layout substantially strengthen their illusory power as predicted by Purves *et al.* [PSL99].

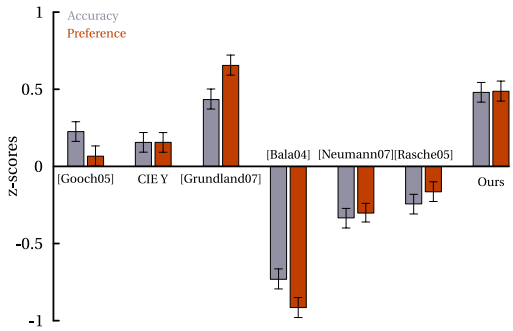


6.44 Failure Cases. The main grief against our outputs is the diminished discrimination between the grey values assigned to clearly different input colors. The first row illustrates the failure of our local enhancement to restore long-range color contrasts, the boundaries of the colored circles against the white background compromising the expected discrimination between them. Yet, our global mapping still yields a perceptually relevant ordering as revealed by an automatic contrast stretching by histogram equalization. The second row highlights a more critical issue where Nayatani's metric does not ensure a satisfactory brightness assignment in terms of information preservation because of the slight color variations scattered across the solidly-colored areas of the painting.

Long-Range Color Contrasts and Discrimination Failures While issues directly related to our contrast enhancement can be alleviated by resorting to more elaborated operators, there are intrinsic limitations to our method. Of these shortcomings, the locality of our second step is the main reason behind our method's failure cases which are always characterized by a poor discriminative power when directly compared to other techniques. Supporting our objective of perceptual accuracy, we do not exaggeratedly stretch the dynamic range of our achromatic outputs and instead favor the restoration of color contrasts of directly adjacent colors. Henceforth, our algorithm cannot reflect contrasts involving distant colors as shown by our achromatic rendition of Ishihara plate No.2 in Figure 6.44*(left)*. Similarly, as a data-independent color transform, our global mapping step may fail to guarantee the necessary degree of distinction between grey levels so that our local enhancement can bring color contrasts back (*cf.* Figure 6.44*(right)*).

6.4 Conclusions

In a recent attempt to go beyond the authors' claims for truly assessing the quality of grey-scale converters, Čadík conducted two series of experiments for evaluating the *accuracy* and *preference* of seven of the most widely known grey-scale conversion algorithms, ours included. To the credit of its author, it is true that no independent and thorough perceptual evaluation of existing techniques had ever been proposed, the sole exceptions being Bala and Eschbach's and Rasche *et al.*'s studies that involved a fairly limited number of test subjects – below 20 – and only compared results from their authors against standard luminance mapping. In stark contrast, Čadík's comparative study relies on the responses of 117 observers with test images spanning from photographs to paintings to cartoons with significantly differing gamuts. His first experiment is meant to estimate a method's *accuracy* whose main requirement is for observers to "select the grey-scale image that better matches the colors of the original color image". By definition, it therefore appears related to Nayatani's VAC experimental set-up at the only difference it involves more visually complex and meaningful stimuli. The second experiment assesses its *preference* and test subjects are charged with the task to "select the preferred grey-scale image from a given pair of achromatic images". From the 20,328 gathered observations, Čadík converts the choices of all observers into standard scores for each test image and displays the results obtained by all seven methods over a 2d graph parameterized by accuracy and preference. Examples of such plots are displayed on the second column of Figure 6.46. These per-image results are especially valuable for us to reach a deeper understanding of the perceptual effect of available techniques on different kinds of imagery. It turns out our concern for perceptual accuracy pays off particularly lucratively in the conversion of complex, real-life images such as photographs, but this comes to the price of showing more mitigated success of inputs whose conversion would require more care for discrimination than our local enhancement pass does. The fact our technique performs that well in terms of preference for complex images is particularly rewarding.

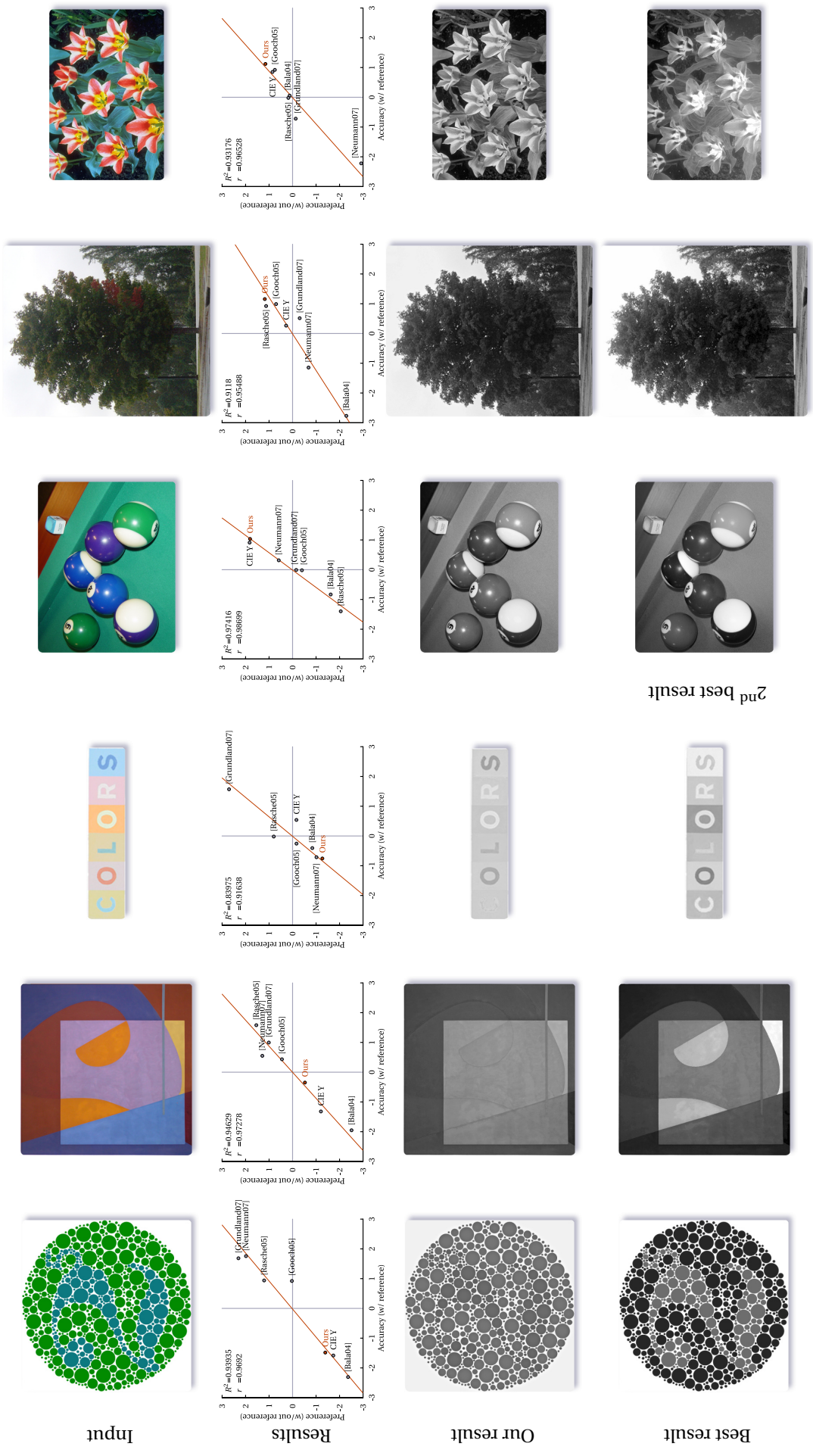


6.45 Perceptual Evaluation of Color-to-Greyscale Image Conversions [Čad08a]. Čadík's experiments for assessing the accuracy and preference of grey-scale converters reveal that our method yields more than satisfactory results. While it excels mostly in terms of accuracy (which was our primary goal to begin with), it also performs well in terms of preference, our technique standing in second position behind Grundland's *Decolorize* algorithm [GD07][9]. As stressed by Čadík himself, our approach also exhibits a consistent performance across images and experiments.

After performing multiple comparison tests, Čadík finally establishes an overall ranking of all seven techniques he considered and finds out that our method achieves the second best performance of 0.487 just behind Grundland's *Decolorize* algorithm topping at 0.544. Overall results for all tested techniques are plotted as error bars in Figure 6.45. More surprisingly, Gooch's *Color2Grey* does not seem to bring the expected level of improvement over a standard conversion, and presents comparable performances with the direct mapping to CIE luminance. Nevertheless, it should be reminded that in spite of these unexpected low performances, Gooch's technique is of the first approaches to cast grey-scale conversion into a non-linear optimization problem for accounting for lost color contrasts.

Further statistical analysis of the observation data at his disposal shows that none of the tested methods can pretend to produce universally good results. This fact is notably illustrated by the obvious complementarity of our method with Grundland's: while our method yields best results on images with wider color gamuts, Grundland's *Decolorize* takes the center stage on images exhibiting a limited number of colors. Far from being either a let-down or a perplexing conclusion, this observation is on the contrary quite motivating as room for a better alliance between perception and discrimination remains.

[9] After the publication of vCad'ik2008's study results, Grundland pointed out that the default parameters used for the comparisons involving his method corresponded to values exaggerating the contrasts it produces, and therefore diminishing its accuracy with respect to the original images: `http://www.cl.cam.ac.uk/~mg290/Portfolio/TurnColorsGray.html`.



6.46 Weaknesses and Strengths of Our Approach. Displayed above are test images and accuracy-versus-preference plots from Čadík's comparative study of the main grey-scale converters available back in 2009 [Čad08a]. Have been incorporated to the study Gooch's *Color2Grey* method [GOTG05], Grundland's *Decolorize* algorithm [GD07], along with Bala's [BE04], Rasche's [RGW05], Neumann's [NCN07] and ours [SLTM08]. Čadík's trials clearly highlighted the pros and cons of our take on grey-scale conversion. By relying on unsharp masking to restore local color contrasts, our method cannot account for distant color disparities lost after the global mapping step (1st *left row*). In addition, by emphasizing perceptual accuracy over discrimination, our technique sometimes fails to deliver the needed amount of contrasts without overshoots when the involved colors are deemed too close by Nayatani's $L^*_{N_{VAC}}$ brightness predictor (2nd *and* 3rd *rows*). Yet, our perception centric approach also represents our method's main strength as it performs particularly well on real-world complex imagery which cannot stand much grey-scale distortions, observers' judgements in terms of accuracy and preference advocating the use of our method in such cases *(right)*.

Part III

Artistic Line Rendering by Example

The research material presented here is the fruit of the collaboration with Amit Shesh[10] who is currently an assistant professor of Computer Sciences at the School of Information Technology, Illinois State University. In its current state, our research does not constitute a finished and published work, and the following will instead constitute a proof of its concept. Its inclusion in this thesis cannot however be questioned as it takes an integral part to our reflection on the conception of efficient example-based content generation methods, and the appropriate balance between analysis, prior knowledge and user guidance that needs to be assessed on a case-by-case basis for that aim.

We present a method for inferring an artist's style in the case of feature line-based renditions of 3d meshes. For that aim, we use well-studied machine learning techniques in order to capture the possible correlations between the differential surface geometry of the 3d scene and its ongoing 2d depiction by the artist. Our work aims at finding the best possible trade-off between automation and ease of manipulation for the user.

Context and Motivations

"The essence of drawing is the line exploring space." **~ Andy Goldsworthy**

Tracing lines, delineating forms using their contours indeed sound like the very basis of any form of visual creation and even with the amount of machinery at one's disposal, starting off with a rough sketch made of a few albeit carefully chosen lines still appears as mandatory as it is natural.

The best illustration of the naturalness of the line based depictions is indisputably cave painting. Before the advent of scripture, first attempts at communication and the transmission of symbolic thoughts involved drawings of the simplest forms, and some of them remain as impressive as ever in spite of their 17,000 year-long lifespan. Indeed, despite the simplicity of their silhouettes and their awkward perspective, most of these paintings represent easily identifiable animals and understandable hunting scenes. The first pictographic stages of most alphabets is also another proof of the easiness and naturalness of communicating through line-based images: from explicit mnemonic images to abstract shapes representing syllabuses to signs evoking the sounds of our vocal chords, any text comes down to an extremely stylized line drawing.

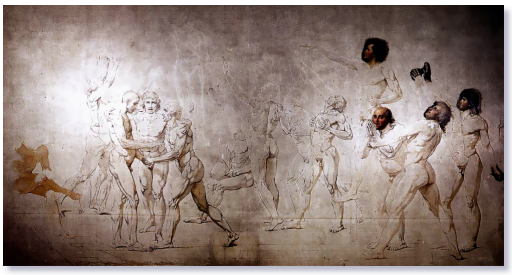
Another fascinating aspect of line drawings is that this form of representation barely changed across centuries. Although artists' skills constantly increased, line drawings still abound nowadays and amusingly enough, the penlight drawn elephant displayed in Figure 6.48 still bears an uncanny resemblance to its prehistorical ancestor. As alluded by Goldsworthy, lines are the most basic graphical primitives for visual representation and therefore the easiest to manipulate. This explains the pervasive nature of line-based representations in art as no matter how elaborated its final aspect, most artworks emanate from rough, line based sketches at

[10] His professional web page can be accessed from http://www.itk.ilstu.edu/faculty/ashesh.

6.47 17,000 Year Old Representation of a Bull from the Lascaux Cave, France.

6.48 Penlight Illustration for the 2008 Annecy International Animated Film Festival. Source: Félix Création

Jacques Louis David,
The Tennis Court Oath, 1789-1772.



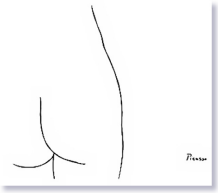
Tatsuya Yoshikawa,
Devil May Cry 4, 2008.

Masahide Tanaka,
Final Fantasy IX, 2000.

6.49 At the Heart of Most Visual Arts. *(left)* Due to funding issues and the political instability plaguing post-revolutionary France, David's majestic rendition of the Tennis Court Oath remains unachieved, revealing the line drawing guiding his painting. Only the portraits of some of the deputies were finalized, exhibiting a high level of realism under which the underlying lines completely disappear. *(middle, right)* Thinking of the arising of mainstream 3d graphic productions as the fall of 2d representations, and more specifically line drawings, would be a mistake. Prior their three-dimensional incarnations, all creations involving design research from characters to settings, start off as a rough line drawings.

Edouard Manet,
Five jockeys galloping,
1867-1871.

Pablo Picasso,
Femme.



Malcolm Sutherland,
Still from *La fête*,
2010.

6.50 Simple and yet Complex. Making possible our mental reconstitution of a horse from a set of lines broken by the illusion of motion, transforming three single curves into an almost erotic figure, suggesting a setting from disparate visual clues hinting the presence of trees and a fence, these are formidable proofs of true artists' mastery of lines.

the earliest stages of their conception. This is particularly true for paintings, regardless of their final degree of photo-realism, but still holds for seemingly more remote forms of creation such as architecture, interior design, sculpture or 3d modeling (*cf.* Figure 6.49).

"Be it the edge of space or time, there is nothing so awe-inspiring as a border." ~ Yukio Mishima

But line drawings should not be reduced to an efficient albeit archaic or unfinished means of visual communication. Beside their improved understandability, they also elicit an indisputable aesthetical appeal.

Following the artistic revolution set in motion by the Impressionist art movement, many artists played with the evocative power of the lines, letting details up to the viewers' imagination. Some line drawings can also dramatically differ from the reality they depict, and yet remain surprisingly recognizable. Edifying examples from the end 19th century to present-day are proposed in Figure 6.50, and although disparate in terms of content, execution and context, all attest the mastery of their respective artist. While not as predominant as colors, lines also contributed to the Impressionist artists' quest for a less controlled, more emotional and instinctive perception of art. By controlling their shape and their geometrical aspect, lines lend themselves very easily to abstraction and stripping down their art to their purest form became the objective of some artists, especially Henri Matisse (*cf.* Figure 6.51).

In accordance with Mishima's thought, lines can actually steal the spotlight from the subjects they represent and take the center stage. Lines sure are fascinating for several reasons: their efficiency as a consequence of their remarkable evocative power; and their simplicity and controllability a skilled artist can put in the service of aesthetics and abstraction. But lines also impress by the artistic margin and stylistic variations they offer. Thinking of quintessential line drawings only as highly abstracted artworks would definitely be misleading as demonstrated by lush art pieces such as Stephen Wiltshire's depictions of city scenes (*cf.* Figure 6.52).

Henri Matisse, *Thèmes et variations*, Variations F1 to F7, 1941.

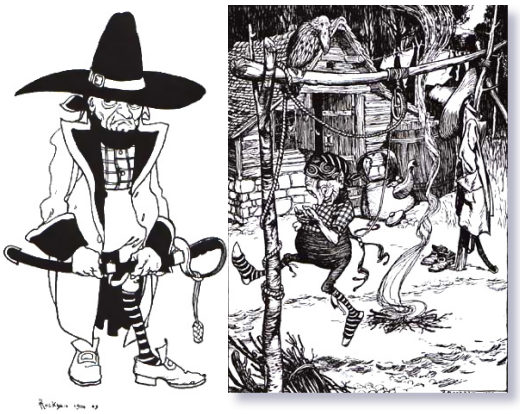
6.51 Aching for Purity. Hard if not impossible to mention line drawings while passing over French painter Henri Matisse's creations in silence. Matisse is internationally renowned for his line drawing collections and numerous variations on a same theme. By progressively discarding superfluous lines while polishing the shapes of the remaining ones, he aimed more at purity than simplicity. Friendly rival painter Pablo Picasso was openly sarcastic about Matisse's way of proceeding, claiming that "in drawing, nothing is better than the first attempt".

Stephen Wiltshire

The Ritz Hotel,
Piccadilly, London, 2008.

6.52 Wiltshire's *Floating Cities*. Nicknamed the "human camera" for his ability of drawing cityscapes with an astonishing level of accuracy from memory, British artist Stephen Wiltshire draws his impressive visual acuity and memory skill from his autism.

Arthur Rackhamm,
The Fairy Tales of the Brothers Grimm, 1932.



6.53 The Versatility of Line Drawings: the Density of Lines. These drawings by British illustrator Arthur Rackhamm constitute a stunning example of the variety of artworks a same artist can achieve with lines. The stylistic contrast lies here in the difference of visual complexity between the two illustrations, the *Rumpelstiltskin* character portrayed on the left appearing much more uncluttered than the scene presented on the right and enjoying a finer graphical treatment as evidenced by the apparition of patterns on his clothing.

The contrast between Matisse's and Wiltshire's artworks in terms of their execution and regardless of the aesthetical motivations of their creator, reveals the extent of the stylistic variations attainable by line drawings, and almost unveils the two ends of their spectrum. Only loosely tied to the shape they represent, lines prove to be in spite of their apparent primitiveness a considerably malleable medium as attested by line drawings differing in terms of visual complexity (*cf.* Figure 6.53), line disposition (*cf.* Figure 6.54) and rendition (*cf.* Figure 6.55).

Artistic "Shortcuts" as Connections to our Brain's "Alternate Physics"

Line Drawings and Pictorial Space

One could marvel at the robustness and consistency of the perceptual inferences we draw from line drawings. Jan J. Koenderink's analysis and understanding of our visual system's pictorial space is quite enlightening in that sense. As the perceptual counterpart of our visual field, the *pictorial space* is the 2d space resulting from the processing by our cognitive processes of the retinal image, and is thus a strictly mental entity. Koenderink thoroughly studies the question of the integration of a scene's visual cues and perception of shape in the con-

Elenore Abbott, *Grimm's Fairy Tales*, 1920.

Arthur Rackhamm, *Fairy Tales by H. Andersen*, 1932.

6.54 The Versatility of Line Drawings: the Choice of Lines. Similar in their subject and treatment, these two drawings present a less conspicuous stylistic difference. Here, Abbott's and Rackhamm's artworks mainly differ in the choice of their lines which gets especially visible in the depiction of the mermaids' hair: while Abbott draws locks confounded with the waves' foam, Rackhamm use many more lines as if representing individual hair. Similarly, Rackhamm's lines on the octopuses' tentacles (visually comparable to the ones on Abbott's waves) elicits a much more pronounced impression of 3d depth.

John Flaxman, Homer's *Odyssey*, 1810.

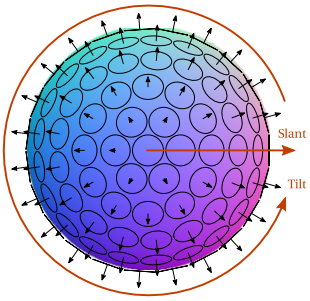
Hayao Miyazaki, Nausicäa of the Valley of the Wind, 1984.

Akihiro Yamada, The Twelve Kingdoms.

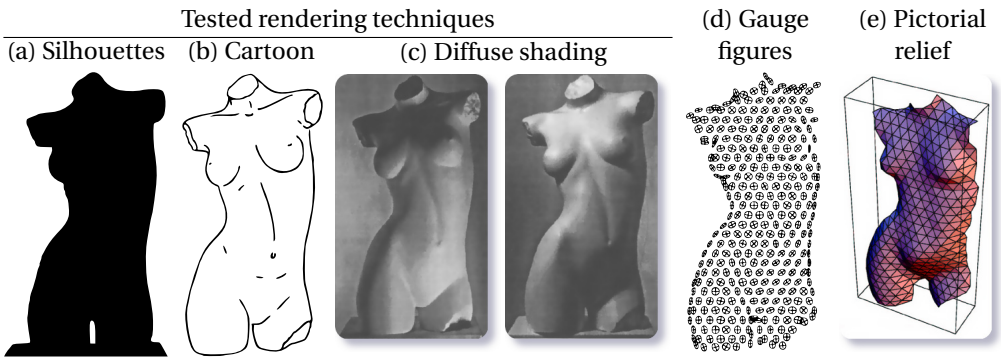
6.55 The Versatility of Line Drawings: the Shape of Lines. If we leave aside technical illustration whose style is more driven by legibility than aesthetics, line drawings offer much artistic freedom through the sole shape of their lines. Flaxman use long and regular lines for describing the folds of the goddess' and the woman's tunics, whereas Miyazaki resort to shorter and rounder lines to give a juvenile look to his character. Finally, in his 2002 illustration for the *The Twelve Kingdoms* series, Yamada makes his angular lines especially sharp to mimic old Chinese woodcuts.

text of monocular vision. Since animals devoid of binocular overlap present depth perception abilities, Koenderink considers monocular depth estimation as perfectly feasible and examines the visual cues extractable from a single to assess their individual influence over the correctness of our inferred *"pictorial reliefs"*. In spite of its 2d nature, the pictorial space still retains parts of the 3d character of the environmental space it emerges from. Koenderink specifically dwells on the mental processes in charge of the recovery of the "pictorial depth" values that fill the 2d barriers corresponding to the perceived projections of 3d opaque objects, as well as the effect of several rendering techniques on the quality of the interpreted pictorial reliefs [KDCL96]. For that aim, he employs the now widely used gauge probing technique detailed in [KvDK92] where subjects are required to orient a gauge figure in such a way that it appears as the projection of a circle painted over the object's surface (*cf.* Figures 6.56 and 6.57*(d)*). Once performed for a sufficiently dense set of positions scattered throughout the surface, the pictorial relief can be reconstructed by integration of the resulting constraint field (*cf.* Figure 6.57*(e)*) and its quality with respect to the actual object assessed. Of special interest, the cartoon rendering (*cf.* Figure 6.57-*(b)*) is generated by manually highlighting salient linear features using black lines over a solid white background, and resembles a line rendering of the shape. Lines are selected by tracing out the brightness discontinuities apparent on differently-lit photographs (*cf.* Figure 6.57-*(c)*) and encompass the exterior silhouettes, the continuation of the contours in the interior of the shape as well as a couple of creases.

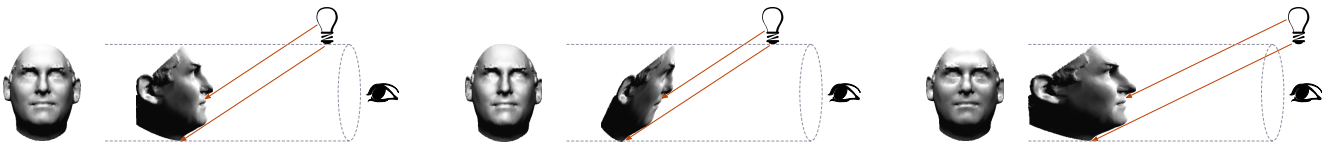
Koenderink's experiment reveals the stunning faithfulness of the pictorial effects established over such cartoon renditions, which mark a real heap in quality from the impoverished perception from the silhouettes to the percepts elicited by fully-shaded images. Koendering sees in diffuse shading only a "weak" visual cue whose addition does not bring that much more information than a set of carefully selected feature lines. Shading cues are also only locally consistent when illumination emanates from a non-unidirectional light source,



6.56 **Koenderink's Gauge Figure Protocol.**



6.57 **Koenderink's Pictorial Relief Unveiling [KDCL96].** By having observers adjust a dense set of gauge figures so that they seem to lie over the object's surface *(d)*, Koenderink reconstructs the object's perceived shape *(e)*. By confronting the obtained "pictorial relief" to reality, he assesses the amount of perceptually relevant information conveyed by a specific kind of rendering *(a)-(c)*.



6.58 **Bas-Relief Ambiguity [BKY97].** Under parallel projection and illuminated by a distant light source, a 3d shape's diffuse-shaded rendition is by essence ambiguous. This corroborates Koenderink's claim that shading is a weak visual clue for complete shape understanding.

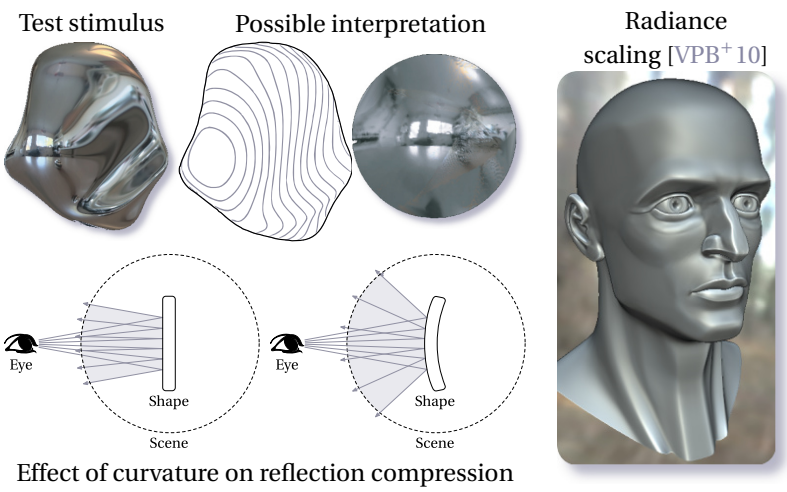
and their efficacy is then also dependent on its main direction (a classical, already encountered bias being that light usually comes from *above* the scene). Diffuse shading is finally known for its ambiguity and can prove misleading when objects are viewed through parallel projection as evidenced by the "bas relief ambiguity" (*cf.* Figure 6.58). In absence of constraints over the light sources' positions, the shape can only be defined up to a certain category of transformations, rightfully called generalized bas-relief transforms. In the context of a monocular depth field, this hurdle for shape assessment adds to the lack of proper radial ordering and the confusion relative to absolute depth estimation.

Where Could People Draw Lines?

Discontinuities again play a crucial role in the creation of our percepts: local disparities yield most of the needed information for our understanding but fail to enable us to attach proper global absolute values, may it be depth or brightness as studied in Part II. Koenderink's experiment suggests that the combination of silhouettes, contours, and curves following shading discontinuities constitute a good set of candidate lines for producing meaningful if not pleasing line drawings. But Koenderink *et al.* "picked out the important linear features" manually from variously lit photographs [KDCL96]. The computational solution for the generation of informative line renderings from a general shape still appears as an open problem.

Independent studies by Roland W. Fleming *et al.* explore the interplay between shape perception and specular reflections, and it happens that although their visual aspect drastically differs depending on the object's surroundings, they enable an as robust shape estimation as textural clues. While the mental recovery of shape from texture is possible thanks to the local foreshortening of their patterns at grazing angles and thus is governed by the depth's first derivatives, the distortion of the reflections' light patterns are dictated by the surface's curvature. The robustness of our percepts to changes in environmental conditions (which lead to completely different reflections) strengthens the intuition that curvature also represents an important, if not foremost intrinsic surface property that should be accounted for finding meaningful lines.

Discontinuities of in their shading, local gaps in depth, specific behaviors of the surface's curvature, these quantifiable properties all convey valuable visual cues that dictate our perception of shape. It then comes nat-



6.59 Shape Perception from Specular Reflections [FTA04]. While depending on the object's surroundings, the reflections' specific distortions are characteristic of the object's intrinsic surface properties such as curvature which govern their local compression. This insight has been recently used by Vergne *et al.* who empower their renderings with an acute shape enhancement [VPB+10] by modulating the reflected amount of light for specific lighting and viewing directions.

urally that stressing out these surface locations across the surface using lines may carry enough information for our brain to reconstruct the complete shape. This ease of interpretation from such lacking depictions is still surprising as explicit lines are nowhere to be found in our daily perception of the real world. Even more perplexing is the seemingly limited influence of stylization conventions over shape understanding and recognition, several experiments reaching the conclusion that certain primates such as chimpanzees are perfectly capable of recognizing objects from line drawings.

Cognition and Art

Following Patrick Cavanagh's analysis of the intangible interplay between visual arts and perception, artists naturally seem to pick the lines that best convey the shape of their subjects and reproduce the minimal set of contours for enabling viewers to effortless recognize them [Cav05]. This recent original examination of art from the neuroscientists' perspective puts artists at the center of the analysis where their erroneous depictions potentially reveal intricate cognitive biases. Many psychological errors committed by artists attest of the selective character of our percept in terms of physical correctness. Famous examples include our tolerance to highly approximate shadows, as well as the famous *"Venus effect"* which refers to our highly erroneous expectations of what reflective surfaces look like.

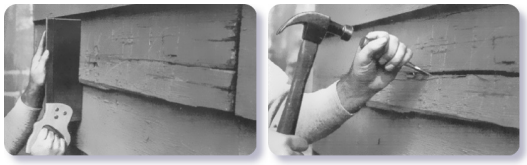
An Alternate Form of Realism in Computer Graphics

Until recently, Computer Graphics mostly ached for the perfect imitation of reality notably by the addition of details that give the illusion of reproducing the complexity of its physical phenomena. Growing alongside the force power of computers, this accumulation of visual cues played a crucial role in the plausibility if not accuracy, of the appearance of computer-generated content. But from the profusion of details to clutter, there is only one easily overstepped limit. When the appearance of rendered objects get distracting to the point of confusion, quality of form supersedes the efficacy of content and impedes the transmission of the information communicated through visuals. The occasional choice of resorting to visually simpler albeit as meaningful renderings earned it the title of *"functional realism"*, promoted to the same rank as the long-exclusive *"photo-realism"* and the currently illusory *"physical realism"* in James A. Ferwerda's analysis of realism in Computer Graphics [Fer03] (*cf.* Figure 6.60).

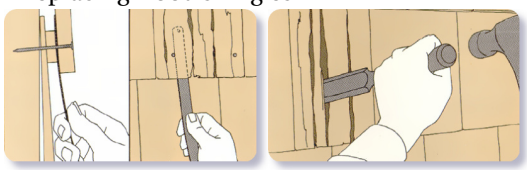
This has long been explored by illustrators and designers fully aware that a clever illustration is often more desirable than a photograph for complex communication. Besides, in spite of their deceiving simplicity, line drawings do elicit a vivid sensation of 3d shape to the point of creating impressively strong illusions (*cf.* Figure 6.61).

At this point of the explanation, we need to clarify what is our exact understanding of the *line drawing*. We follow Hertzmann's classification splitting line art into two groups of increasing visual complexity: *silhou-*

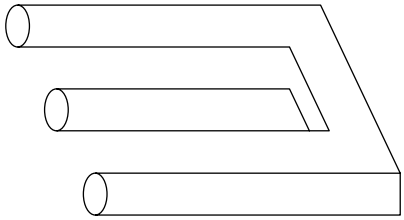
How to replace clapboard



Replacing wood shingles

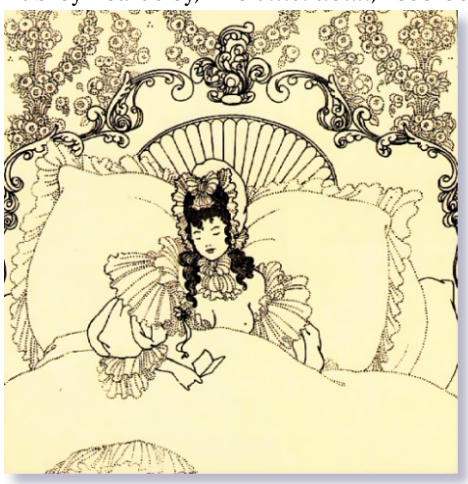


6.60 Ferwerda's Functional Realism [Fer03]. James A. Ferwerda derives three interpretations for *"realism"*: *physical realism* qualifying images eliciting the same stimulation as the light reflected from real-world scenes, *photo-realism* describing images indistinguishable from their real-world counterpart, and *functional realism*. Unlike the first two kinds, functional realism is more driven by the preservation by the images of the visual information conveyed by the scenes than the faithfulness of their appearance.

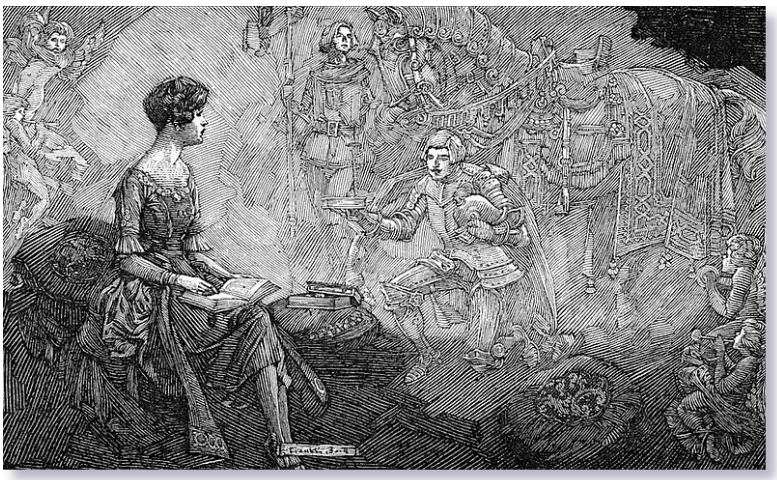


6.61 The Power of Line Drawings. Though simple, this drawing nevertheless locally yields a powerful sense of 3d shape even if it depicts an impossible shape when considered in its entirety. Similar to René Magritte's *Carte blanche*, it draws its strength from visual cues hinting occlusions that disrupt the global consistency of our pictorial depth perception.

Aubrey Beardsley, The *billet doux*, 1895-96.



Franklin Booth, 1927.



6.62 Silhouette Drawings *Versus* Pen-And-Ink Illustrations.

ette drawings and *pen-and-ink illustrations* [HZ00] (*cf.* Figure 6.62). Our concern mostly revolves around the former kind of drawings made of sparse lines meant to best convey the rendered objects' shape. Conversely, pen-and-ink illustrations are characterized by the presence of filling patterns whose varying density accounts for changes in brightness caused by surface variations, shading or textural cues. While the silhouette drawing term is by essence misleading (most sparse line renditions include more than sole silhouettes), we focus on this category of renditions as its more stripped down aspect takes full advantage of our natural shape inference abilities.

Sparse line renditions best exploit the clarifying power of lines while simultaneously conveying shape-related information. Their simple individual appearance also allows for effective manipulation that can serve communication: superfluous details caused by shading, shadows, texture are naturally omitted; shape is more amenable to clarification and simplification; a varying level-of-detail representation can draw attention to specific or salient geometrical features; occluded surfaces can be hinted by way of dashed lines.

These aforementioned techniques and "shortcuts" are all tricks mastered and skillfully used by artists for improving their drawings' functional realism, but achieving convincing results using computers is much more involving than what Koenderink's experiment would imply. Blindly tracing out brightness discontinuities for instance is not adapted in most cases: details in shadowed regions are likely to be missed while extraneous

lines would delineate shadow boundaries and abrupt reflectance changes.

Outline

Slowly but surely, more and more techniques for the automatic line based depiction of 3d geometry have been proposed, and more than the number of methods itself, the fact that they become hardly comparable both in terms of results and methodology is becoming somewhat of an issue as attested by the recent urge for experimental validation [CGL+08, CSD+09].

It turns out these aforementioned user studies did not reach any finite and clear conclusions. More than the perplexing choice of determining the best algorithm in terms of efficacy or pleasantness, the concern mostly revolves around the validation of the actual relevance of each line rendering techniques, or lack thereof. It also happens that no existing method proposes truly satisfactory results on all kinds of geometry. This is dreadfully true for specific and familiar geometries, typically faces whose line-based renditions are subject to strong expectations and aesthetical conventions. This observation hence begs the question of the *controllability* of such methods, and the appropriateness of their control. We claim that bringing users into the rendering loop is mandatory for such techniques to be successful, and decide to find ways to do so. Our work is therefore similar in spirits to Eric B. Lum's take on the problem which to the best of our knowledge, is the unique attempt to include user feedback, and aims at ultimately surpassing it. We propose herein an interactive approach where user feedback, editions and corrections of the ongoing line drawing are processed and anticipated thanks to machine learning. Line drawings are thus not the outcomes of algorithmic "black-boxes" more or less predictably controlled via global thresholds anymore, but the results of the joint collaboration between users and the machine. In contrast with the different techniques mentioned in Chapter 7, we believe that while it should not be entirely abdicated, automation has to give way to controllability and present our approach for example-based line rendering in Chapter 8.

CHAPTER 7

State of the Art in Line Rendering

Despite their seemingly simplicity, lines yield one of the most powerful means of shape depiction and even if highly stylized or abstracted, their consistent understanding among viewers attests their relevancy with respect to the mechanisms of the human visual system.

These benefits were soon identified and exploited by automatic methods in expressive rendering that ditches photo-realism for the intelligibility of their outputs. The present section details most of the research conducted for that aim, and attempts to best highlight the pros and cons of all methods. We follow the broad classification between image-space and object-space methods, presented in Section 7.1 and 7.2 respectively. Proposing a final ranking of all these techniques is difficult if not impossible because of the disparity of their reasonings. Perplexity has grown alongside the ever-increasing number of proposed approaches, and has notably begged the need of studies for their careful validation.

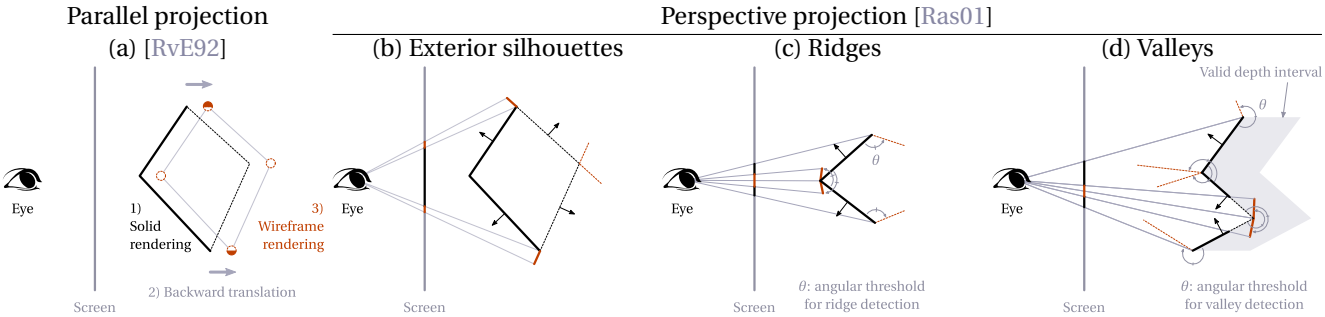
7.1 Tracing Lines in 2d Image Space

Subsequent methods differ in terms of complexity of their handling of the geometric information they can extract from the scene, may it be from its complete 3d representation or a collection of images captured from it. We introduce them in order of increasing complexity, from the hijacking of the rendering pipeline's z-culling procedures, to the filtering of geometric information storing buffers, to edge detection in shaded renditions of the scene.

7.1.1 Exploiting the Projected Scene Depth

Building upon Catmull's solution to the visibility problem and the removal of hidden surfaces [Cat74], work by Jareck Rossignac *et al.*, later extended by Ramesh Raskar *et al.*, consist in exploiting the information stored in the scene's z-buffer to create line drawings from polygonal scenes [RvE92, RC99, Ras01]. Mostly concerned by the legibility of geometrically complex renderings, Rossignac *et al.* strive to the explicit display of tessellation and silhouette edges present in the scene and having their style reflect their visibility [RvE92]. *Silhouette edges* refer to edges adjoining front-facing and back-facing polygons (polygons whose normal respectively points towards and away from the camera). Rossignac proposes four different simple procedures depending on the extracted edges and the treatment reserved to the hidden ones. For silhouettes, scenes have to be drawn twice: the first pass draws them in solid white and initializes their z-buffers; scenes are then rendered in wire-frame mode with thick lines after translation away from the viewer so that only exterior silhouettes that go over the actual geometry survive the z-test (*cf.* Figure 7.1*(a)*). Similarly, Raskar *et al.* conceptualize silhouette drawing as the accentuation of intersection between the set of front-facing visible polygons and the second layer of back-facing polygons behind them [RC99]. By adjusting the successive z-tests, they directly draw edges in screen-space but the resulting raster lines are stuck with a one-pixel width only.

In order to trace lines with a user-specified screen-space width, Raskar *et al.* propose to draw additional geometry: for rendering silhouette edges, they "fatten" back-facing polygons by drawing triangle fans whose edges are pushed outwards the original primitives' edges by a view-dependent amount. This amount is first established so that the silhouettes appear of the specified width under parallel projection [RC99], and is later adjusted to account for its dependency on the screen space position under perspective projection. They also propose routines for highlighting intersection edges, ridges and valleys by way of additional geometry [Ras01] (*cf.* Figures 7.1*(b)*-*(d)*). *Ridges* and *valleys* are defined in terms of the dihedral angle of their associate edges:



7.1 First Image-Based Approaches. Rossignac and Raskar manipulate the depth values of the scene via z-culling procedures. Rossignac performs a second drawing pass that renders the geometry in wire-frame mode *(red dots)* after backward translation *(a)*. Raskar ensures constant screen-space line width under perspective projection by attaching geometry to the scene's polygons *(in red from* (b) *to* (d)*)*. Fragments surviving z-tests are displayed in solid lines whereas hidden geometry is drawn in dashed lines. The case of the valleys is more complex as it requires the existence of an "in-between" depth test function and necessitates two separate z-buffers.

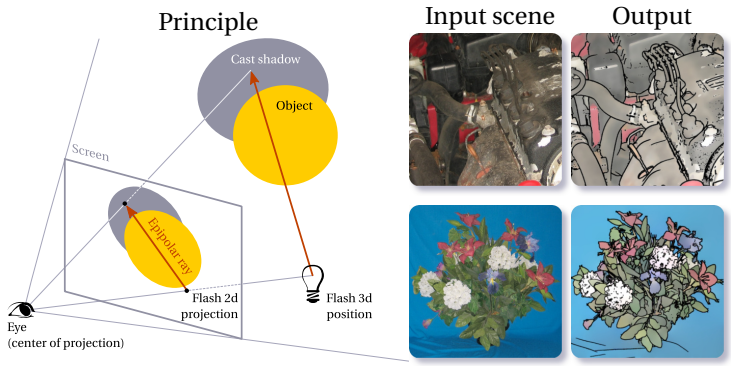
an edge is a ridge, or a valley, if the angle formed by its two adjacent faces' planes lies below, or above, a given threshold. Supplemental quadrilaterals are added to front-facing and back-facing polygons in ways their visible parts highlight ridges and valleys respectively.

These witty methods shine by the fact they only require a consistent orientation of the facets' vertices and do not rely on connectivity-related information. Since primitives can be processed independently, such techniques show impressive performances and lend themselves to graphics hardware acceleration. Their authors notably had this end prospect in mind, Raskar envisioning his geometry appending approach as a shader upstream from the programmable graphics pipeline. Alas, they suffer from the classical issue of z-fighting due to the quantization enforced by the z-buffers finite resolution, and gaps appear when angles between successive line segment are too sharp.

Raskar *et al.* further limit their line rendering method's prerequisites by proposing a "self-contained stylized imaging device", more naturally called NPR camera [RTF+04]. By having a camera equipped with a set of carefully disposed flashes around its projection center, line-based depictions of scenes can be acquired without even having to explicitly handle their geometry. The collection of images obtained by the successive shootings with distinct flashes is sufficient to localize and emphasize depth discontinuities by tracking down the negative intensity transitions caused by shadows along the epipolar ray from the light source associated with each image (*cf.* Figure 7.2*(left)*). They indeed reveal local depth discontinuities, the foreground foreshadowing the underlying background, and are easily distinguished from intensity edges caused by changes in the materials' reflectance as they are appear in only one image of the collection. Only depth-related edges hence survive the intensity normalization process that discards such spurious responses, and the false positives caused by specular reflections are removed by considering the scene's intrinsic image reconstructed from the median gradient field estimated from all images. One can resort to a hierarchy of flashes placed at increasing distances from the center of projection to capture edges when the background lies at a too important distance from the occluding foreground object (the cast shadow ending up being detached from its generating depth edge). Among other contributions, Raskar *et al.* suggest an impressive range of non photo-realistic stylizations (*cf.* Figure 7.2*(right)*), and even extend their approach to video by combining successive differently-lit photographs of the sequence.

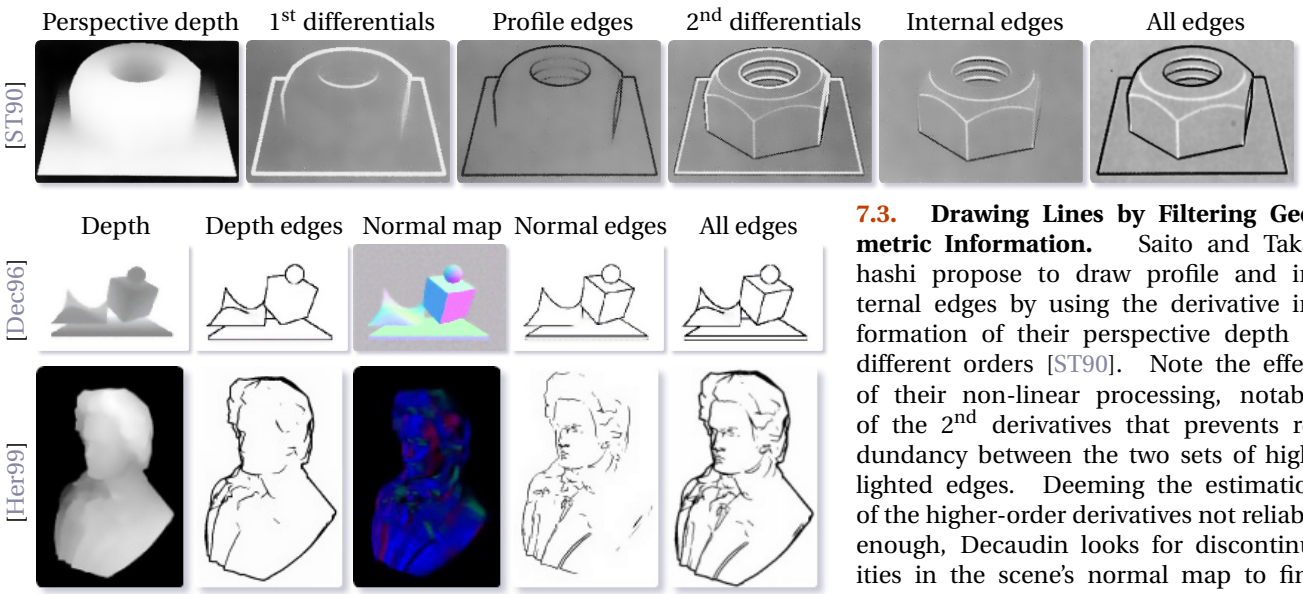
7.1.2 Processing more General Geometric Information

Prior to Rossignac *et al.*'s approach, Saito and Takahashi proposed a more general approach for shape enhancement to improve the comprehensibility of renderings of intricate geometries [ST90]. They hence pioneered the resort to alternative displaying methods for increasing the functional realism of confusing imagery. For that aim, they explicitly manipulate a wider set of geometric information than depth alone, that they store in "G-buffers" consisting of 2d arrays recording the visible surfaces' geometric information at each



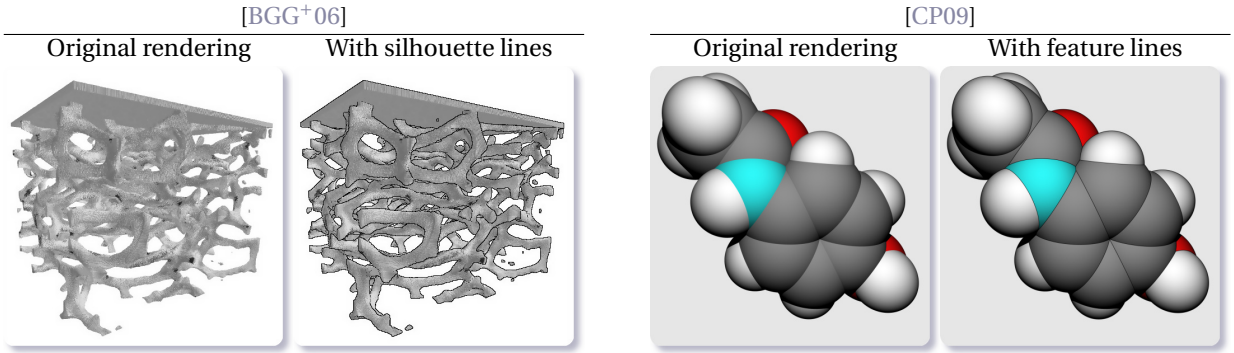
7.2 Raskar's NPR Camera. By walking along epipolar rays in screen-space, Raskar *et al.* find all intensity transitions corresponding to depth discontinuities present in the real-world scene *(left)*. By performing the same operation for all different point lights, line drawings can be obtained in raster format and composited to an abstracted rendition of the input scene *(right)*. Abstraction is carried out by reconstructing images from the gradients lying behind the highlighted lines via a Poisson solver.

screen location after perspective projection. They perform their enhancement as a post-process, independently of the geometric (projection, visibility) and physical processes (shading, texture), and already provide users with controls for adapting it to the scene, task or preference. Chief among the possible enhancement alternatives, the drawing of the profile and internal edges is carried out as the detection of the perspective depth's discontinuities estimated by the convolution of the depth associated G-buffer with a 3×3-wide mask. Filtering is made non-linear by locally normalizing gradients and correcting the 2nd derivatives to limit artifacts and line overdraw (*cf.* Figure 7.3 *(top)*). In their terminology, *profile edges* and *internal edges* are comparable to silhouettes and crease lines, and are composited with conventional renderings to improve their legibility. By trading the perspective depth G-buffer for an image containing the lengths of the rays cast by each of its pixels, boundaries caused by reflection and refraction can also be emphasized. Alas, their method shows pronounced aliasing artifacts and breaks with transparent materials. Saito and Takahashi also consider objects' *ids*, normals, 3d surface positions and 2d patch coordinates to propose further enhancements closer to pen-and-ink illustration such as *contour lines* and *hatching curves*.



7.3. Drawing Lines by Filtering Geometric Information. Saito and Takahashi propose to draw profile and internal edges by using the derivative information of their perspective depth at different orders [ST90]. Note the effect of their non-linear processing, notably of the 2nd derivatives that prevents redundancy between the two sets of highlighted edges. Deeming the estimation of the higher-order derivatives not reliable enough, Decaudin looks for discontinuities in the scene's normal map to find crease lines [Dec96]. In spite of the artifacts due to the raster nature of the approach, Hertzmann illustrated successful renderings of more complex models in his course notes [Her99].

For his cartoon-inspired rendering technique, Decaudin draws inspiration from Saito and Takahashi's approach and proposes a cell-shading rendering characterized by visible feature lines, solid colors and hard shadows. Similarly to [ST90], silhouettes are found as edges in the scene's z-buffer. Yet, the detection of crease lines using the depth's 2nd derivatives is deemed too unstable, and Decaudin finds in discontinuities of the scene's normal map a more robust alternative although he disregards the normals' multi-dimensionality during filtering 7.3-*(middle)*. The targeted rendering style however imposes the thresholding of the estimated differ-



7.4 Incorporating Line Drawing to Simulation Visualization. Silhouettes are very efficient cues for communicating the shape of materials simulated using particles *(left)*, and clarifying the spatial layout of the components of complex structures *(right)*.

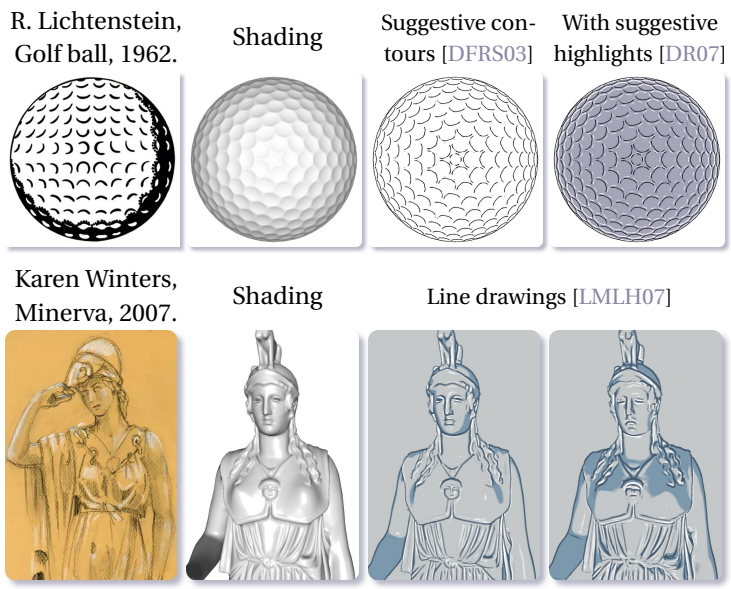
entials' magnitudes so that to mimic the sharpness of ink-drawn curves, henceforth reinforcing the aliasing-bound flickering. Line computation is thus conducted on higher-resolution G-buffers to alleviate such artifacts.

As hinted by Saito and Takahashi's reflection and refraction enhancement, image-based line rendering solutions are conveniently amenable to ray tracing. Their underlying principle which relies on the filtering of 2d buffers aligned with the projection screen, lends itself quite easily to the ray tracing framework where the output's pixels are traversed by scene exploring rays. Along their number of bounces, rays can record information concerning their travel and encountered surfaces, and once halted, store it at their corresponding image locations. By their naturally structuring albeit abstractive ability, lines are valuable clues for the visualization of complex particle systems, and have thus be recently applied to enhance renderings obtained by Material Point Methods[1]. Lines, and particularly silhouettes, are all the more useful to convey shape information as the involved particles are part of a greater whole, and then facilitate the interpretation of the macroscopic structures emerging from their simulation. Bigler *et al.* dedicate a separate thread for the computation of silhouettes in the context of their interactive, massively parallel ray tracer [BGG+06], and compute them as the zero-crossings of the depth's second derivatives unveiled by a Laplacian-of-Gaussians filter. The thresholding of their magnitude provides indirect control over the amount of clutter once composited to the rendered simulation. Bigler's requirement for interactivity comes at the price of aliasing issues and his approach is later extended in Choudhury *et al.*'s renderer embedding the feature line computations into the ray tracing algorithm itself [CP09]. Choudhury proposes a variation of cone tracing for overcoming the lack of explicit geometry dedicated to the infinitely thin lines. By considering the geometric information lying at the vicinity of each primary rays in screen-space and smoothly quantifying relevant discontinuities (in terms of object *ids*, depth or normals), he can trace naturally aliasing-free silhouette, intersection and crease lines with a constant screen space width directly dependent on the sampling fineness of his ray stencils (rays surrounding the central sample disposed in concentric circles in image space).

7.1.3 Finding Edges in Shaded Images

A last trend in image-based line detection works on the final renditions of the scenes once the physical processes (notably local illumination) have been completed. By assuming the Lambertian nature of the objects' material and carefully disposing the illumination (by typically having the unique point light source coincide with the camera's center of projection), intensity variations caused by shading reveal valuable surface information such as its *slant* (dot product between the surface normals and the viewing direction). Processing the information conveyed by the reflected light intensity therefore leads to the emphasis of geometrically salient features. For instance, DeCarlo *et al.* propose image-space algorithms for approximatively detecting their *sug-*

[1] MPM are finite-element methods which represent the solid materials they simulate by a collection of particles whose individual motions caused by applied forces deform the material's overall shape.



7.5 Line Drawings as Abstracted Depictions of Shading. Shown here are line drawings obtained using image-space algorithms that trace lines in 2d by following specific intensity variations of the diffuse-shaded shape. As studies in visual perception established, shading is a double-edged sword as it can convey a formidable amount of information concerning the shape of objects, but easily break down and is subject to restricting conditions for it to be effective (uniform albedo, unidirectional illumination, convex objects) [KDCL96]. Similarly, specular reflections are also demonstrated to be of much interest for shape perception [FTA04]. These insights, along the factual evidence brought be creations of artists, fueled DeCarlo *et al.*'s and Lee *et al.*'s approaches to line drawings by the processing of shaded images of the scene [DFRS03, DR07, LMLH07].

gestive contours [DFRS03], and *suggestive highlights* [DR07]. Theoretically defined in object-space, it happens that such lines roughly line up with intensity valleys and ridges under the aforementioned lighting conditions. Their locations can thus be detected by considering local intensity statistics, and spurious responses and irregularities removed through median filtering (*cf.* Figure 7.5*(top)*).

But resorting to shading is not only a roundabout for processing geometry through fast 2d image processing. Shading can also be considered for aesthetics and the dramatic effect caused by lighting. Lee *et al.* formalize their line drawings as "abstracted" forms of shading, and argue that relevant lines correspond to linear features conveying most of the shading related intensity variations, from contours underlying shadows to highlights stressing specular reflections [LMLH07]. Their lines coincide with the ridges of the height field defined by the grey-scale shaded image of the freely-lit scene. They extract them by robustly computing the curvature principal magnitudes and directions by fitting a degree-2 polynomial to the local intensity variations around each image pixel. Pixels gifted with a principal curvature of sufficient magnitude are flagged as candidates for being part of a line. Candidates lying in smooth regions or at step edges' borders are removed, and only the pixels at the vicinity or at the apex of ridge are kept. Lee's drawings hence have their lines provide shape, but also lighting cues. But this direct dependency on lighting represents both a feature and a limitation as certain lighting set-ups may hide geometric salient features in shadowed regions (*cf.* Figure 7.5*(bottom)*). Nevertheless, the performances of Lee's approach profiting from hardware accelerations allow users to interactively tune more complex lighting schemes to cope with such cases.

7.1.4 Image-Space Approaches: The Pros and Cons

The aforementioned techniques have several desirable characteristics but these come at the expense of as crucial aspects. The following enumeration lists them out and points out attempts to alleviate their limitations. To put it in a nutshell, here is how these different drawing approaches can be termed:

Simple. This is the most conspicuous aspect of these image-space approaches. Their theoretical simplicity and intuitiveness that entails their algorithmic clarity and ease of implementation. Indeed, most of them relies on simple image processing routines such as filtering or non-parametric valley detection that can fairly easily be put to the test and assessed [ST90, Dec96, Her99, DFRS03, DR07, LMLH07]. Perhaps the most involving of these approaches may be the ones indirectly making use of the culling procedures that are automatically carried out by the z-tests [RvE92, RC99, Ras01]. While maybe more prone to implementation mistakes due to

the fiddling necessary to handle the multiple rendering passes, they still only require the appending of supplemental geometry around polygons that follows simple analytical rules.

Fast. Attendant to the previous point, the brightest appeal of image-based approaches is their speed.

Relying on the independent, local processing of their primitives, such techniques are directly amenable to programmable graphics hardware. This feature is paramount notably from the perspective of their integration to video games where timings allocated to superfluous rendering effects are thoughtfully and only sparingly granted.

In addition, since they are directly carried out by the graphics pipeline or constitute enhancements posterior to the rendering of the scene, they do not have to cope with concerns such as line visibility. This contrasts with the more involving computations needed to assess 3d object-space line segments' local visibility. But this casualness comes at the price of the impossible straightforward handling of transparent objects.

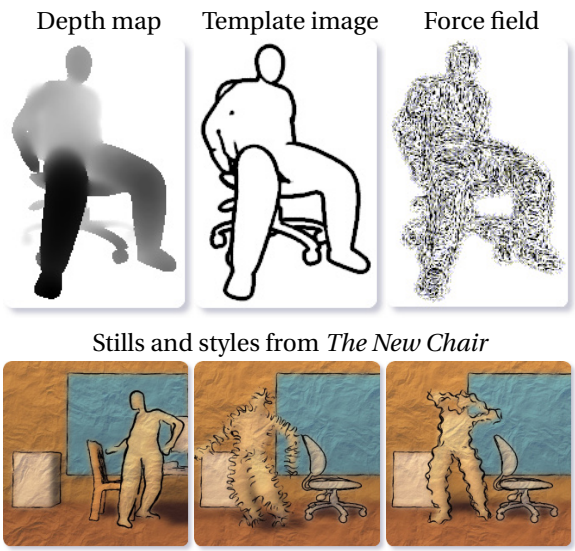
Another appeal is the dependency of their complexity to the end image's resolution, regardless of the scene's geometric complexity. A desirable side-effect is the natural level-of-detail handling such methods ensure. While Raskar *et al.*'s requirement for constant screen-space line width contradicts the previous statement, techniques relying on sampled geometric or shading information whose sampling scheme follows the screen's pixel grid seamlessly simplify their drawings and prevent clutter by having distant discontinuities falling at the same screen positions.

General. One usually underrated advantage is the fact they can directly be applied regardless of the computational representation of the scene's geometry, and work on deformable, key-framed or procedurally animated models. This strength stems from the fact lines are computed at the latest stages of the graphics pipeline (either once parametric surfaces have been tesselated, or once the geometry has already been projected and stored in G-buffers).

Aliased. Their most obvious drawback is the pronounced aliasing that characterizes lines produced by image-space approaches. Either due to z-fighting [RvE92, RC99, Ras01] or the limited precision of the processed data caused by the storage's sampling and quantization, these lines often exhibit disturbing aliasing-related artifacts which get even more alarming when viewed in motion, evoking a shimmering effect at the object's boundaries. Immediate solutions for these issues involve the local averaging of the test responses for producing anti-aliased lines such as Decaudin's recourse to higher-resolution and smoothly minimized buffers [Dec96], or Choudury *et al.*'s cone tracing that enables the modulation of their lines' local strength [CP09].

Stylistically-impaired. Certainly the most frustrating shortcoming of image-space methods for the creative constraints it imposes on the final line appearance, the raster nature of the lines hinders most stylization freedom. Indeed, defined as sets of independent pixels, the handling of spatially consistent variations of these lines' visual attributes requires additional processing. Only rudimentary styles can be natively achieved such as the attribution of a constant color or screen-space width [RC99, Ras01], or assigning different colors to different kinds of lines [ST90], or locally modulating the lines' color with geometric-driven cues such as local shading intensity [LKL06].

This limitation is especially unnerving when the detected lines are meant to imitate a drawing as the exactness of their shape strictly following the underlying geometric features reveals their computational nature and conflicts with the liveliness of actual artworks. A fairly sophisticated and convincing stylization for such raster lines has been proposed by Curtis for his "loose and sketchy" animations where characters have their silhouettes' appearance reflects their emotional state [Cur98]. For overcoming the lack of explicit lines, Curtis relies on a particle-based system where the particles are guided to a vector field pushing them towards the edges to stylize until they reach a region filled with ink (*cf.* Figure 7.6).

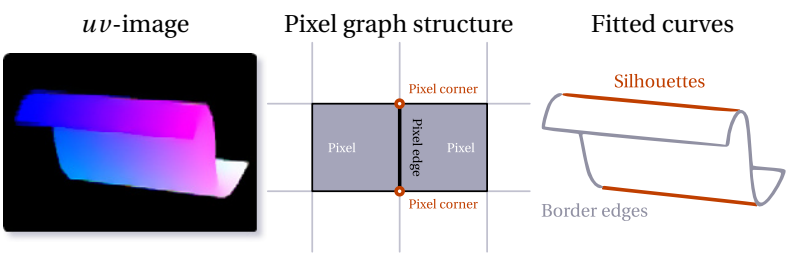


7.6 Curtis's Loose and Sketchy Animation [Cur98]. The per-pixel depth gradient magnitudes constitute the scene's *template image* which records the amount of ink needed at each pixel location to create the final drawing. The end result is obtained as the equilibrium state of a physical particle system guided by the scene's *force field*. It specifies the particles' acceleration values and prompts them to move towards the locations of the silhouettes (it usually corresponds to the orthogonal field of the depth gradient directions). Particles attaining an ink-filled region after random walk are rendered, while the ones falling in ink-deprived areas disappear and are replaced by new ones. By controlling the walk's randomness and drag parameters, users can achieve quite an impressive variety of styles given the very low-level representation of the original lines.

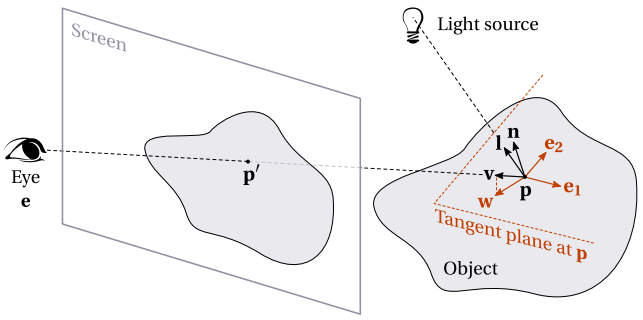
For having image-space lines provide a comparable range of stylization alternatives as their object-space counterparts, one would need to explicitly parameterize them in order to find their *brush path*. Corrêa *et al.* propose to compute a vectorial representation of raster lines found as the discontinuities of their uv-image of the scene by processing the directed graph made of pixel edges to recover polygonal chains and fit B-splines to them [CJTF98]. While overcoming the issue of the lack of parametrization of the original lines, their solution then fails to retain the real-time appeal of former, raster-exclusive solutions. As an attempt to find an acceptable trade-off between speed and stylization, Lee *et al.* suggest in their real-time pencil rendering system to have raster lines "shaken" and drawn several times [LKL06]. This operation consists in having the final rendering of the scene texture mapped to screen aligned quads, and perturbing their texture coordinates via sine functions to confer a wavy effect to the final rendition. Over-sketching is similarly achieved by compositing several renderings of the scene with different coordinate alterations. This trick remains compatible with graphics hardware but since only controlled by screen position, is also prone to the infamous "shower-door".

But even parameterized, image-based lines still could cannot offer the same variety of stylization as object-space 3d lines since the complete 3d information is lost during projection and cannot be fully recovered with images alone. The often encountered line stylization consisting in having line thickness account for the 3d surface's curvature hence seems compromised [Dec96].

Tessellation-bound. At the exception of shading-based solutions [DFRS03, DR07, LMLH07], image-based solutions can usually only highlight *edges* of the geometry. This is especially noticeable in Rossignac *et al.*'s and Raskar *et al.*'s work who base their solution on multiple rendering passes of the scene or the slight alteration of part of its geometry. While not that dreadful an issue if the tessellation is fine enough, this aspect is radically different from object-space approaches that can interpolate their measurements across the objects' surface and find the best possible locations for their lines.



7.7. Fitting Parametric Curves to Raster Lines. Built upon the user-specified matching between a simple 3d textured model's feature lines and a 2d doodle's curves, Corrêa *et al.*'s method for texturing hand-drawn characters requires the vectorization of their initially raster edges extracted from the geometry [CJTF98]. Silhouette and border edges are identified as the outputs of a color discontinuity detector run over the 3d model's uv-image *(left)*. Edges are then classified to either belong to a silhouette or a border, and a directed graph of pixel edges is incrementally constructed and updated *(center)*. Its traversal finally enables the fitting of B-splines to the continuous polygonal chains made of the pixel edges of a same category *(right)*.



$\mathbf{e}$	Position of the camera's center of projection
$\mathbf{p}$	Point sitting on the object's surface
$\mathbf{p}'$	Projection of point $\mathbf{p}$ onto the screen plane
$\mathbf{n}$	Unit normal vector of the surface at $\mathbf{p}$
κ_1, κ_2	Principal curvatures s.t. $\lvert\kappa_1\rvert \geq \lvert\kappa_2\rvert$
$\mathbf{e}_1$	Principal direction of maximal curvature at $\mathbf{p}$
$\mathbf{e}_2$	Principal direction of minimal curvature at $\mathbf{p}$
$\mathbf{v}$	View vector at $\mathbf{p}$
$\mathbf{w}$	Projection of $\mathbf{v}$ onto the tangent plane at $\mathbf{p}$
$\mathbf{l}$	Light vector at $\mathbf{p}$

7.8 Notations. Unit vectors belonging to the tangent plane of the surface at $\mathbf{p}$ are displayed in red *(left)*. It should be noted that the definition of the view vector $\mathbf{v}$ can differ from one paper to another. Unless contradictory notice, we assume that the projection to the screen is a perspective projection. In such a case, $\mathbf{v} = \mathbf{e} - \mathbf{p}$, whereas $\mathbf{v}$ remains constant regardless of the screen position in the context of an orthogonal projection.

7.2 Tracking Lines in 3d Object Space

If we consider the issue of line extraction from the perspective of our perception of shape, it comes rather naturally to cast it as a 2d "edge" detection problem. The edge term is voluntarily restrictive as this is not mere raster images we analyze here, but rather the pictorial space that emerges from our perception of the considered scene. "Finding edges" in 2d image-like buffers storing local information about the scene geometry and layout is just a computational roundabout to localize and process salient features in pictorial space from an approximative cognitive point of view. As we saw in Section 7.1.3, such methods already provide acceptable results and in spite of their intrinsic shortcomings and failings theory-wise, constitute the easiest if not only solutions for *real-time* line generation of animated models.

Conversely, if we consider the same issue of effectively depicting an object's shape through a set of carefully chosen lines from a geometric perspective, object-space methods appear as intellectually more satisfying and exact alternatives. Thus, while image-space approaches may find their value in their promptness and assumed proximity to our cognitive processes, object-space methods draw their strength from the accuracy with which they assess geometric saliency.

This strength paradoxically also turns out to be the "weakness" of this category of techniques as for each possible definition of saliency, a new technique arises, sometimes difficult to evaluate with respect to the others. More than a fundamental weakness, this aspect represents more an hindrance to their wide-spread use. Object-space methods can roughly be subject to two complementary classification criteria: the order of derivation of the surface geometry required by their analysis, and the attached importance to the viewing conditions. We begin this section dedicated to object-space approaches by detailing approaches that only consider the objects' geometry, followed by techniques that also account for the viewing conditions. All necessary concepts and notions relative to the differential geometry of 3d surfaces can be found in Appendix B.

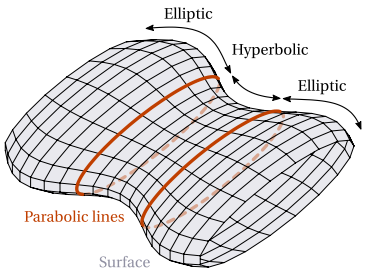
7.2.1 Automatic Detection of View-Independent Lines

We now review the different flavors of lines directly extractable from 3d geometry, but before diving into the subject, we briefly introduce in Figure 7.8 the notations that we will manipulate throughout the remaining of the present chapter.

In the following, we dwell on the case of view-independent curves, as we believe introducing them first is more intuitive since they solely account for shape features. They are therefore mostly defined in terms of abrupt variations of the surface's normals which are quantifiable using curvature differentials, or as the loci of zero-crossings of some curvature-related measurement.

7.2.1.1 Parabolic Curves

7.10 Parabolic Curves Over the Statue of Apollo of the Belvedere. Driven by the desire to unveil a mathematical framework that would provide rational explanations behind artistic beauty, Klein had some of the parabolic curves visualized over the famous Leochares' statue, but met quite a harsh disenchantment when it turned out that these lines neither assumed intuitive shapes, nor provided visually satisfying results.



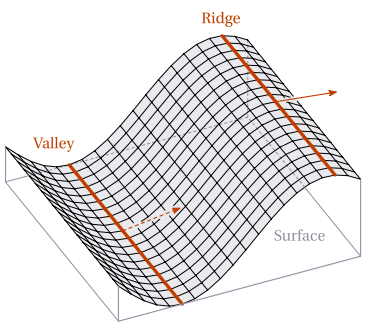
7.9 Parabolic Lines.

Considering the meaningfulness of the sign of Gaussian curvature κ_g as illustrated in Figure B.5, one could expect a given surface's parabolic curves to convey relevant information in terms of its shape. Also called flexional curves, the *parabolic curves* are defined as the loci of points where the Gaussian curvature vanishes: they therefore segregate elliptic regions from hyperbolic portions of the surface (*cf.* Figure 7.9). While such features indisputably constitute great local surface descriptors, the perceptual validity of the zero-crossings of the Gaussian curvature still cannot be taken for granted. This is factually illustrated by German mathematician Felix Klein's studies on parabolic curves [HCV52]. In a attempt to grasp a deeper understanding to the hidden links between mathematics and commonly accepted notions of beauty and aesthetics, Klein had the parabolic curves of the head of the Apollo of the Belvedere drawn directly onto a replica of the famous statue as shown in Figure 7.10. Still, no intuitive perceptual meaning of such lines' positions across the statue surface could unfortunately be found. Given this apparent inadequacy between the parabolic curves of a surface and the perceptual intuition we have from it, no line drawing generation methods actually rely on this kind of lines. Other inconveniences also hinder their use such as the instability and unpredictability of their shape when small or distant perturbations of the surface are introduced, as well as the lack of proper assessment of their "strength" which could enable the elimination of spurious lines.

7.2.1.2 Ridges and valleys

Perhaps the most efficient and commonly employed set of view-independent lines, ridges and valleys – alternatively referred as a whole to creases or crest lines depending on the nature of the geometry – actually propelled most research on the estimation of curvatures and their differentials quantities on discrete geometry.

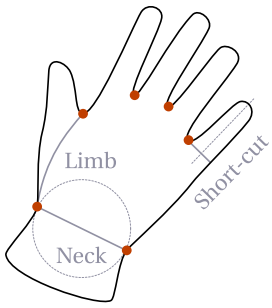
Perceptual Relevance of Ridges and Valleys



7.11 Ridge and Valley Lines.

Thanks to his deep understanding in differential geometry and visual perception, Koenderink first proposes the mathematical definitions for ridges and valleys that he ranks amongst the surface's singular surface features [Koe90]. Both kinds of lines embrace loci of sharp variation points of the surface normals, and as locations where the surface suddenly bends, represent relevant visual hints for shape communication (*cf.* Figure 7.11). Ridge lines are defined as the loci of points at which the normal curvature assumes a local maximum in the principal direction associated with the largest positive curvature. Complementarily, valleys coincide with loci of points at which the normal curvature reaches a local minimum in the direction associated with the largest, negative curvature:

$$\begin{array}{lll} \text{Ridge lines} & \kappa_1 > 0 \text{ and } D_{\mathbf{e}_1}\kappa_1 = 0 \text{ and } D_{\mathbf{e}_1}D_{\mathbf{e}_1}\kappa_1 < 0, \\ \text{Valley lines} & \kappa_1 < 0 \text{ and } D_{\mathbf{e}_1}\kappa_1 = 0 \text{ and } D_{\mathbf{e}_1}D_{\mathbf{e}_1}\kappa_1 > 0. \end{array} \tag{7.1}$$



7.12 The Minima Rule in Flatland. Several rules for shape decomposition have been appended to Hoffman and Richards's initial hypothesis, all involving local minima of curvature *(red dots)* as at least one extremity of the cuts *(grey lines)*: the limp, neck, and short-cut rules. The first kind of cuts links two curvature minima under a constraint of good continuation. The second rule draws cuts between curvature minima if both belong to a circle enclosed to the shape. Lastly, short-cuts correspond to cuts starting from a curvature minimum towards a boundary point and crossing a symmetry axis of the resulting subpart are severed if a symmetry axis of the resulting part has been crossed in the process.

The configuration where $\kappa_1 = 0$ corresponds to locations where the surface is comparable to a plane or reaches an umbilical points. Principal curvatures and associated directions cannot be defined in such cases. Fortunately, Belyaev *et al.* prove that crease lines do not pass by such locations [BAK97].

Chief among the lines directly computed over surfaces, creases amaze by the meaningfulness of their disposition across shapes 7.11. Ridges gather at the apex of locally convex shapes, and when superimposed over the shaded rendition of an homogeneous matte version of the surface coincide with its highlights. Valleys on the contrary stress sharply concave areas subject to ambient occlusion and self-shadowing and henceforth appearing darker than the average intensity of the object surface. This natural correspondence between valleys and thin dark areas makes them perfect candidates for lines composing a drawing made of ink strokes.

Additionally, studies demonstrate the remarkable perceptual relevance of valleys for complex shape decomposition. It has been experimentally proven that observers naturally partition shapes along curves following valley lines. This perceptual inclination is called Hoffman and Richards's *minima rule* which states that for the sake of shape recognition, the human visual system divides shapes into parts at loci of negative minima of the largest principal curvature along its associated principal direction [HRP+84]. Our instinctive shape decomposition scheme is indeed visually governed by hints of the traversability of the constitutive parts of the shape, and by contours of concave discontinuity that appear on surface made of convex intersecting subparts.

Other cognitive studies dwelling on the side-problem of the shape recognition from textural clues also highlight the prominent role of the principal curvature directions. Years before Hoffman and Richards's minima rules, Stevens first unveils the influence of principal curvature information on our perception of shape, and hypothesizes that observers when confronted to a surface only represented by a set of lines lying onto it, base their judgements on the assumption these roughly follow lines of principal curvature [Ste81]. Following the same rationale, Stevens and Brookes later prove that when made coincident with the surface's directions of principal curvature, lines showing through the surface also serve as compelling visual cues of its local slant, and henceforth help test subjects to infer pictorial reliefs from monocular images thanks to this natural bias [SB87]. The pertinence of principal curvature directions have furthermore be exploited in expressive rendering notably for guiding hatching and other filling patterns for pen-and-ink illustration [GIHL00, HZ00].

Extracting Ridges and Valleys for 3d Rendering

Because of the aforementioned evidences of their perceptual relevance, most efforts for object-space line drawing generation focus on the robust detection of ridges and valleys. Following their computation of differential surface quantities in the context of volumetric data, Monga *et al.* adorn their visualization renderings with crease lines although they do not explicitly manipulate lines but rather independently processed edge points. Interrante *et al.*'s approach mostly differs in its crease line detection tests [IFP95]: in place of Monga's approximation of the iso-surface's third-order differential (which potentially suffers from instability or over-smoothing), Interrante "walks" along the direction of maximal principal curvature and compares the current value of the maximal principal curvature with values of the normal curvature computed for neighboring points along that direction. Her resort to crease lines is motivated by her need to convey shape as concisely as possible in the context of medical imaging for the visualization of the radiation located within patients' bodies. Using a sparse set of carefully selected lines (rendered via a transparent texture) is here a non-invasive shape

revealing solution that lets way to the display of other information. Opacity can also be tuned in accordance with the curvature magnitude, if not simply thresholded.

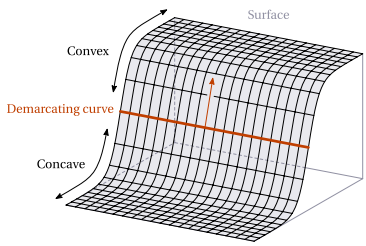
Puzzled by the poor performances of local surface-fitting methods for computing curvature derivatives on complex triangular meshes, Ohtake *et al.* propose to globally fit a surface consisting of the summation of spatially bounded radial-basis functions [OBS04]. They build a hierarchy of nested point sets scattered across the mesh's surface, and compute the interpolating function in an iterative fashion. The function interpolating the hierarchy's coarsest sets is progressively refined by adding the partition of unity approximation of the next finer point set and supplemental RBFs to it. Once the global fit has been performed, normal curvature and its derivatives can be analytically approximated using Monga and Benayoun's formulae for iso-surfaces. Ridge and valley points are finally found across edges of the mesh by checking the respective conditions of Equation 7.1, the precise location of the zero-crossing of the curvature derivative being predicted by linear interpolation along the edge. One of the appeals of Ohtake's method is how easily it lends itself to line abstraction: by computing the crease lines over one of the intermediary versions of the fitted surface, details are naturally and consistently omitted without endangering the closeness of the resulting lines.

Extending Goldfeather and Interrante's cubic approximation of differential quantities, Yoshizawa *et al.* consider back the case of local surface fitting [YBS05] and smooth the input mesh by replacing its original vertices by the arithmetic mean of their adjacent faces' centroids. Normals are re-estimated using Max's method [Max99] before the application of Goldfeather and Interrante's fitting procedure for each vertex of the new mesh (for the sake of stability, direct neighbors whose normal forms an obtuse angle with the central vertex's normal are ignored). They better guarantee the connectivity of the resulting lines thanks to an additional procedure aiming at the minimization of their fragmentation, and quantify their strength as the closeness of the local surface to a Duplin cyclide. They finally derive a quadric-based mesh simplification algorithm that smoothes the mesh's regions that are far from a crease line in terms of geodesic distance.

7.2.1.3 Demarcating Curves

It has been proved that crease lines turn out to be superbly efficient line features for shape communication, with a slight advantage for valley lines at the light of the minima rule and their natural coincidence with shadows. Still, in order to convey all shape information, both kinds of crease lines usually have to be displayed together (often with differing colors, lighter tones being dedicated to ridges, darker ones for valleys) and when used in the context of line drawing generation, may result in overloaded drawings.

This issue has been handled by Kolomenkin *et al.* who suggest a new type of lines meant to provide as much information as the union of ridges and valleys while limiting clutter [KST08]. Their *demarcating curves* are defined so as to highlight the loci of strongest inflexion of the surface (when the surface suddenly changes from convex to concave). As such, they run "in-between" ridge and valley lines, hence their name (*cf.* Figure 7.13). Mathematically speaking, they are defined as follows. Given a surface point $\mathbf{p}$, the tangential direction along with the curvature *derivative* reaches a local maximum is sought and then constitutes the direction of maximal curvature variation $\mathbf{g_p}$ at $\mathbf{p}$. $\mathbf{p}$ is a demarcating curve point if the normal curvature along $\mathbf{g_p}$ vanishes which means that the surface locally behaves like a plane along $\mathbf{g_p}$.



7.13 Demarcating Curves.

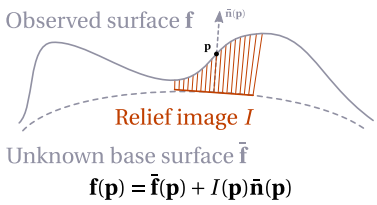
$$\text{Demarcating curves} \qquad \kappa_{\mathbf{n}}(\mathbf{g_p}) = 0 \;\text{ and }\; \mathbf{g_p} = \arg\max_{\mathbf{x}} \frac{D_{\mathbf{x}}\kappa_{\mathbf{n}}(\mathbf{p})}{||\mathbf{x}||} = \arg\max_{\mathbf{x}} \frac{C_{\mathbf{p}}(\mathbf{x},\mathbf{x},\mathbf{x})}{||\mathbf{x}||^3}. \tag{7.2}$$

Directions of maximal normal curvature variation are computed for each mesh vertex $\mathbf{p}$ by finding the roots of a third-order polynomial obtained by parameterizing the space of possible tangential directions by their angle with the maximal principal direction. The normal curvature at $\mathbf{p}$ along $\mathbf{g_p}$ is then evaluated using the Euler formulae B.2. Mesh edges are sequentially processed and once a zero-crossing has been found, the final position of the demarcating curve point is estimated by linear interpolation. Demarcating curves are interesting lines from their interplay with other kinds of lines: they run between ridges and valleys and as they isolate con-

vex regions from concave regions, coincide with the zero-crossing lines of the mean curvature. By definition, demarcating curves are a much more intelligible subset of the parabolic curves. By their ability to delineate local surface convexities from concavities, Kolomenkin *et al.* render them on top of the mean curvature-shaded rendition of the surface, their alliance revealing even more surface details. Kolomenkin's technique is mostly geared towards the automation of relics drawing in the context of archaeological illustration, and attention is therefore more attached to the lines' descriptive power than visual attractiveness. While legible, drawings made of demarcating curves exhibit a line density twice as important as for other object-space techniques. Moreover, demarcating curves do not highlight the surface's protruding and depressing features, and sometimes break when confronted to noisy surfaces.

7.2.1.4 Relief Edges

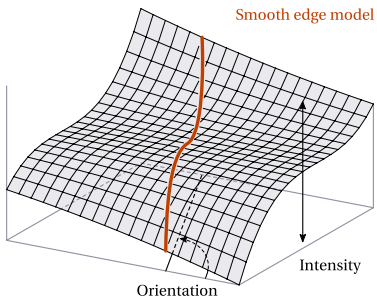
For their subsequent take on the issue of line drawing generation, Kolomenkin *et al.* drastically change their views on the question and re-cast it into a finer and more robust edge detection problem [KST09]. They represent the object surface as the layering of an unknown smooth manifold $\bar{\mathbf{f}}$ adorned with a local *relief image* I consisting of a height field storing the missing high-frequency details and defined on the smooth base surface's tangent plane (*cf.* Figure 7.14). Relief edges are naturally defined as edges in this relief image I, and instead of considering its partial derivatives in order to localize them, Kolomenkin *et al.* suggest to locally fit a smoothed edge model to the surface $\mathbf{f}$. In order to propose an analytic optimization procedure, they resort to the Monge's polynomial approximation $I_{\mathbf{p}}$ of the surface on the tangent plane at a given point $\mathbf{p}$ that similarly to a function's Taylor expansion, involves the second- and third-order structures of the surface $\mathbf{f}$:



7.14 Kolomenkin *et al.*'s Surface Representation in 2d.

$$I_{\mathbf{p}}(\mathbf{x}) = \frac{1}{2}\mathrm{II}_{\mathbf{p}}(\mathbf{x},\mathbf{x}) + \frac{1}{2}\mathrm{C}_{\mathbf{p}}(\mathbf{x},\mathbf{x},\mathbf{x}).$$

The model graphed in Figure 7.15 proposes two parameters by way of the edge's intensity and orientation, and is fitted to the local surface approximation $I_{\mathbf{p}}$ by minimizing its integrated Euclidean distance error estimated over a neighborhood of $\mathbf{p}$. Relief edges are finally defined as the loci of points for which the normal curvature of the surface at $\mathbf{p}$ in the tangential direction $\mathbf{g}_{\mathbf{p}}$ *perpendicular* to the best fitted edge vanishes:



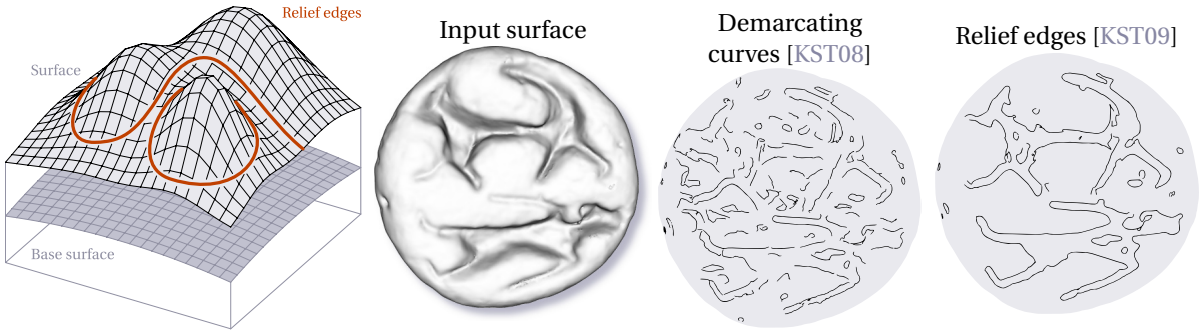
7.15 Smooth Edge Model.

$$\text{Relief edges} \qquad \kappa_{\mathbf{n}}\left(\mathbf{g}_{\mathbf{p}}^{\perp}\right) = 0 \;\text{ with }\; \mathbf{g}_{\mathbf{p}} = \arg\max_{\mathbf{x}} \int \left|\left| I_{\mathbf{p}}(\mathbf{x}) - \mathrm{edge}(\mathbf{x}) \right|\right|. \tag{7.3}$$

They are henceforth fairly reminiscent to the demarcating curves, stressing sharp inflections and segregating convex from concave regions. As a model-based approach, difficulties occur when the surface strongly deviates from the picked model, several orientations then becoming possible for a same location. Kolomenkin *et al.* handle such cases by limiting the range of acceptable edge orientations via an estimate of the smooth base surface's normals at the considered point. Gaussian smoothing can also be performed in order to yield high quality edges by smoothing the scalar normal curvature function $\kappa_{\mathbf{n}}(\mathbf{g}_{\mathbf{p}})$ across the surface. As for the demarcating curves, relief edges are dedicated to the line-based renditions of archaeological artifacts, and command a marginal advantage against Kolomenkin *et al.*'s former attempt in terms of quality and robustness to surface noise as evidenced in Figure 7.16.

7.2.2 Automatic Detection of View-Dependent Lines

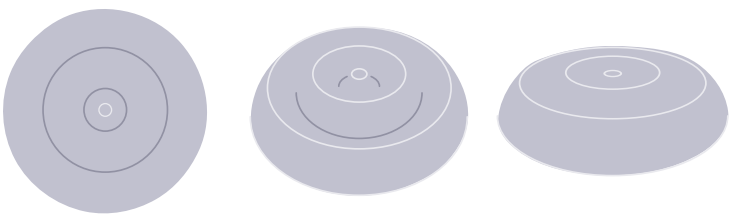
The previously reviewed view-independent lines represent compelling shape conveying enhancements, hence their intensive use for surface description, analysis, matching and registration where their sole reliance on shape make them robust and valuable tools. But in the context of rendering, this property is not as desirable.



7.16 Relief Edges and Demarcating Curves. Both defined as the loci of points corresponding to abrupt inflexions of the surface (zero-crossings of the normal curvature in direction of maximum surface variations), demarcating curves [KST08] and relief edges [KST09] often end up in visually similar results. The main difference actually stems from their fundamentally differing framework: while demarcating curves require the direct analysis of the surface using its local tensors of curvature and curvature derivatives, relief edges emanate from the transposition of model-based edge detection algorithms to the context of geometric feature detection.

The lines stick too faithfully to the highlighted surface features, and often exhibit an unsatisfactory "painted look". Against all odds, Girshick *et al.* advocate the sole resort to principal curvatures and their associated directions for the generation of 3d line drawings [GIHL00], and suggest to introduce a motion field defined over the principal direction field in order to diminish the "pasted" appearance of such lines. But their solution sounds cumbersome and somewhat unclear from the perspective of its practical implementation and artistic control. Without devaluating the geometric and perceptual relevance of the view-independent curves, it is now commonly admitted that the line-based rendition they produce, are not that compelling from an artistic standpoint, not only because of their lack of "liveliness", but also clutter.

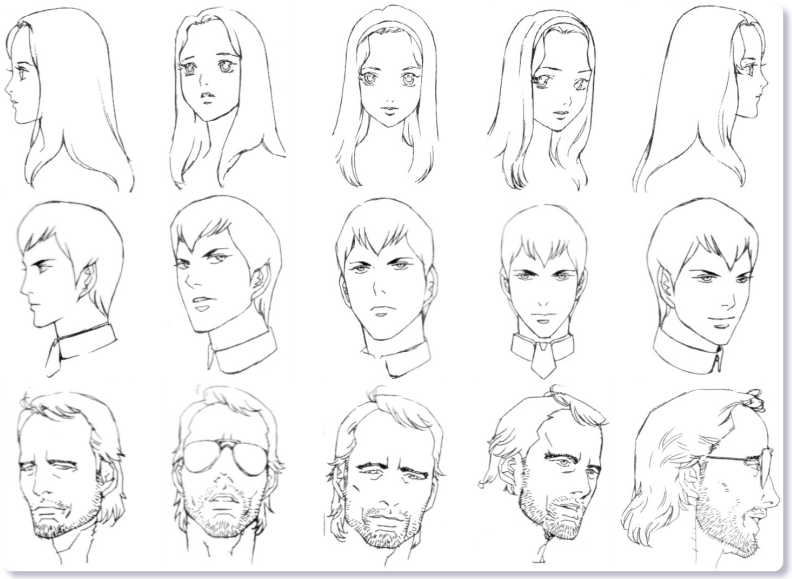
In order to mitigate the startling effect of the stillness of view-independent lines, some attempts at altering their rendition so that they would account for the viewing conditions have been briefly explored, mostly altering the local appearance attributes of the lines depending how adapted to the current view their portions are. DeCarlo succinctly illustrates such an approach in the case of the parabolic lines and loci of zero-mean curvature [DFRS03]. But, these examples were only to demonstrate the fact that local visibility of curves could be meaningfully controlled by curvature derivatives in the viewing direction. Dwelling longer on the question of making originally view-independent lines somehow related to the viewing conditions, Na *et al.* undertake to "redeem" crease lines [NJLS05]. Motivated by their perceptual relevance, they suggest to have their visibility, strength and brightness locally reflect the angle between their local surface normal and the view vector (*cf.* Figure 7.17). But while these improvements produce more satisfactory results than their entirely view-independent counterparts, one cannot think of these tricks as genuinely view-dependent lines, their location remaining as rigid as before.



7.17. Na *et al.*'s "Redeemed" Crease Lines [NJLS05]. Due to their outstanding perceptual relevance in terms of shape communication, Na *et al.* argue for sticking with ridge and valley lines, and only locally tune their appearance depending on the angles their normals form with the view vector. While ridge points are reinforced when their normals diverge from it, valleys are conversely strengthened when their normals aligns with it. This dual and complementary choice also happens to considerably limit clutter as illustrated above.

The desirability of extracting truly view-dependent lines is further strengthened when one looks at artists' work, where lines are carefully chosen for a particular view and aim at the conciliation between image understanding and visual appeal. This aspect of their lines is all the more conspicuous in their depiction of faces where pleasantness often takes precedence over accurate shape detailing (*cf.* Figure 7.18). While attractive for sole rendering, view-dependence can represent a inconvenience for other purposes. The visualization of com-

Akihiro Yamada, Concept artworks for *RahXephon*, 2002.



7.18 View Dependence of Hand-Drawn Lines. These different character designs clearly demonstrate the instinctively view-dependent nature of the lines composing real-life drawings. It seems indeed like a safe bet to assume that artists' primary goal is to find the best possible compromise between their personal aesthetical preferences and the unambiguous representation of the subject to be depicted from a given viewpoint. Following DeCarlo's remark [DFRS03], view dependence is especially manifest for the lines depicting the characters' noses (especially stylized in Japanese cartoons). In most cases, the line delineating the nose from the background on profile views progressively vanishes when the characters are observed in more frontal views.

plex data cannot afford such beautification of their lines for instance, as since there would be no more strict correspondences between line points and their associated surface points, line-based enhancement on stereo image pairs would be ruled out, and the sliding aspect of the lines could impede the understanding of the multi-layered rendition of the data. Victoria Interrante notably states that in the context of user experiments, observers often report such lines to evoke an unexpected "crawling over the surface" effect. This impression may actually be worsened by her will to have lines conveying shape while simultaneously displaying volumetric data inside of it.

7.2.2.1 Silhouette and Contour Curves

Maybe the strongest argument in favor of view-dependent lines is their incontestable necessity. Indeed, while view-independent lines provide valuable clues over the surface, therefore at the interior of its projection onto the screen plane, they do not delineate it from the background or other occluding objects. As such, most renderings always require the addition of another independent set of lines meant to highlight these objects' outlines.

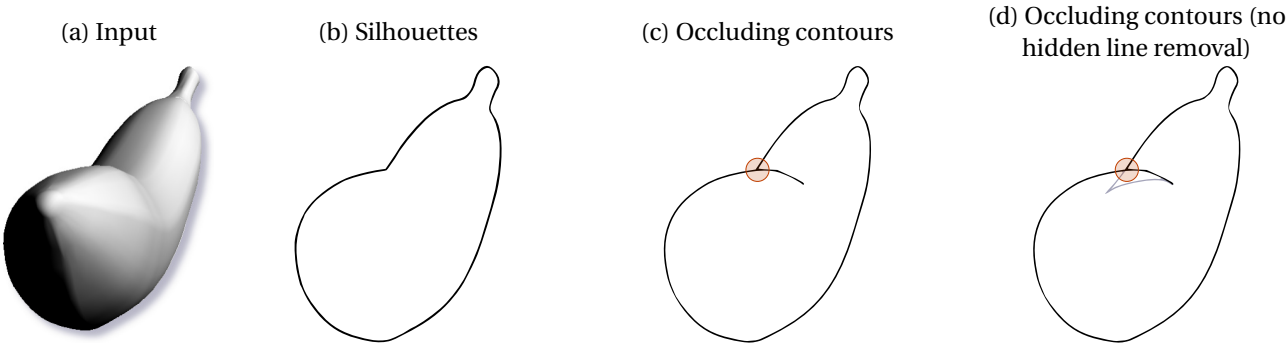
Perceptual Relevance of Silhouettes and Contours

Silhouette and *contour lines* are the 2d projections of the loci of points where the surface turns away from the eye. This notion can be mathematically captured by stating that they correspond to the points whose normals gets orthogonal with the view vector:

$$\text{Occluding contours} \qquad \mathbf{n}.\mathbf{v} = 0. \tag{7.4}$$

Silhouettes slightly differ from contours as they only form the object's strict outline, whereas the latter can also appear inside the shape of the projected object onto the screen (*cf.* Figures 7.19*(b)* and 7.19*(c)* respectively).

But more than their convenience, the appropriateness of the silhouettes and contours to the processes of our visual system is perhaps their uppermost remarkable feature. All the more remarkable as they are nowhere to be found in real-life scenes, and are subject to constant change since not even attached to specific shape features. Still, either when observed as a static set of curves, or as a dynamically evolving drawing, they grant us deep insights on the surfaces of objects. Mathematical proofs along with several visual experiments corrob-



7.19 Silhouette and Contour Curves. Shown above are line drawing renditions of a banana shaped surface *(a)*. While the silhouettes only highlight the object's outline and appear continuous *(b)*, the occluding contours make their way inside of its projected shape and do not show the same level of continuity *(c,d)*. These discontinuities give rise to specific line patterns such as the so-called *t-junctions* and *x-junctions*, circled in red on *(c)* and *(d)* respectively, that convey important visual information about the object's shape and material.

orate the intuition that even if their representation of objects is lacking, observers draw relevant and unambiguous inferences about the objects' shapes from them.

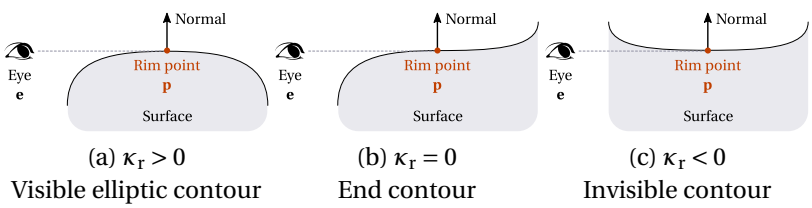
Research dedicated to shape-from-contour were thoroughly conducted by Marr, and later contradicted and extended by Koenderink. Of outmost fundamental interest, Koenderink's theorem relates the contour's *apparent curvature* with the surface's gaussian curvature, and constitutes mathematical explanation of the observers' ability to infer intrinsic differential properties of a shape by the interpretation of its contours on their retinal images. In his formulation, Koenderink distinguishes two important concepts: the contour and the rim as illustrated in Figure 7.20. The *contour* is the curve resulting from the intersection of the unit sphere centered around the center of projection and the conical surface made of all the rays emerging from it and locally tangent to the surface. The *rim*, or contour generator, designates the loci of surface points where these rays touch the surface. While the rim is a smooth space curve, the contour on the other hand may exhibit cups and self-intersections. Koenderink first considers at each rim point the *radial curvature* κ_r and the *traverse curvature* κ_t which correspond to the normal curvature computed in the view direction once projected onto the point's tangent plane and the orthogonal tangential direction respectively. He then derives the following fundamental equality:

$$\kappa_g = \kappa_r \kappa_t.$$

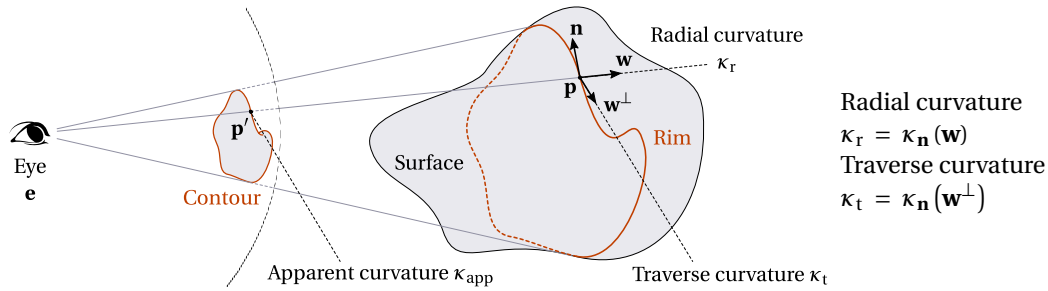
By accounting for the projection from rim points to contour points, he directly links the contour's *apparent curvature* with the surface's intrinsic gaussian curvature:

$$\kappa_{app} = ||\mathbf{e}-\mathbf{p}||\,\kappa_t = ||\mathbf{e}-\mathbf{p}||\,\frac{\kappa_g}{\kappa_r}. \tag{7.5}$$

Since the radial curvature κ_r cannot be negative at a contour point (otherwise the contour would be occluded as graphed in Figure 7.21), the sign of the contour's apparent curvature κ_{app} directly reflects the sign of the surface's gaussian curvature κ_g. Henceforth, from the local convexity or concavity of the contour ($\kappa_{app} > 0$ and $\kappa_{app} < 0$), the synclastic or anticlastic character of the corresponding local surface ($\kappa_g > 0$ and $\kappa_g < 0$, *cf.* Figure B.5) gets unambiguously assessable.



7.21. Radial Curvature at Occluding Contours. At the core of Koenderink's theorem is the observation that at a rim point $\mathbf{p}$ corresponding to a visible contour the radial curvature κ_r can either be positive *(a)* or null *(b)*, whereas rim points exhibiting a negative local radial curvature are necessarily occluded once projected onto the retinal image in order to yield the shape's contours *(c)*. Rim points of null radial curvature *(b)* mark the end of the occluding contour curves and may also coincide with locations where *suggestive contours* appear [DFRS03].



7.20 Koenderink's Rim and Contour [Koe84]. Koenderink explicitly distinguishes the rim curve from the contour curve in his analysis of how the perception of the latter could reveal shape information of the former. Koenderink's theorem states that at any rim point $\mathbf{p}$, the product of the radial and traverse curvatures, κ_r and κ_t respectively, equals the gaussian curvature of the surface κ_g at that location. He derives a direct relationship between the surface's gaussian curvature at $\mathbf{p}$ and the 2d contour's apparent curvature κ_{app} at $\mathbf{p}'$, and formally proves that the local curvature of the 2d occluding contours does reveal information related to the observed 3d shape.

7.22 Osvaldo Cavandoli's *La linea*. The fame enjoyed by the nameless character created by Cavandoli is yet another proof of the expressivity of line drawings, here restricted to their simplest form. Only thanks to the silhouettes that emerge from a single white line, Cavandoli managed to breathe life into his character for approximately 90 3 minute long episodes.

A last configuration of much theoretical interest is when κ_r vanishes on a contour. According to Equation 7.5, κ_{app} becomes infinite, and such locations correspond to the contour's cusps, or *ending contours*. Seamlessly handling such configurations motivated Koenderink's distinction between the continuous space rim and the discontinuous contour. All his findings constitute milestones in shape-from-contour theory which are all the more estimable as they invalidate part of Marr's prior results (notably his assumption that the contour's convexities and concavities depend on the relative distances) and lift previous constraints on the rim's planarity and the assumption of an infinite viewing distance.

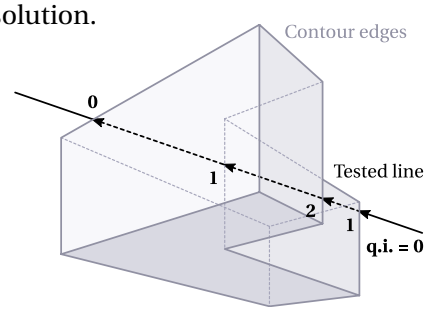
Despite the lack of any final mapping between silhouettes and contours and points of the surface, their dynamic nature also reveals much of the object's shape. Experiments show that observers, when presented to a series of renditions of a same object by their silhouettes from different vantage points, still manage to identify and mentally reconstruct its complete shape [WO53]. Similarly, other studies established that test subjects could also identify moving objects only by the dynamically changing drawing of their silhouettes and the verbal description of the rotational motions these objects were subject to [Tod85]. If we follow Cavanagh's interpretation of the aesthetic shortcuts found in artworks as an indirect revealer of our visual system's aptitudes, we see that artists such as Italian cartoonist Cavandoli instinctively play with our surprisingly ease of interpretation of the silhouettes (*cf.* Figure 7.22).

Either statically or dynamically, silhouettes and contours convey critical shape information, but much of the significance of the contour also lies in its discontinuities: indeed, *t-junctions* represent decisive visual hints for occlusion (*cf.* Figure 7.19*(c)*), sometimes to the point of creating impossible shapes (*cf.* Figure 6.61), whereas the perfect alignment of the lines constituting *x-junctions* is the condition for achieving the illusion of material transparency [Cav05] (*cf.* Figure 7.19*(d)*).

Silhouettes and Contours in Computer Graphics

When attempting to render objects with silhouette and contour curves, early research in Computer Graphics encountered two main hurdles to overcome: the need for fast procedures for their identification, and the issue of their local visibility. The former problem may appear less aggravating with the computational power and the programmable hardware architectures of nowadays' machines, but the fact contours need to be determined at each frame has long been a perplexing issue for their use. The second problem arises from the fact that before their projection onto the screen, contour points lie on the rim which is a closed 3d space curve. Since these lines coincide with the very locations where visibility is at stake, the simplistic handling of line visibility cannot be properly solved by common hidden surface removal procedures such as the z-buffer test, for it would be extremely sensitive to z-fighting and would not provide any fundamental solution.

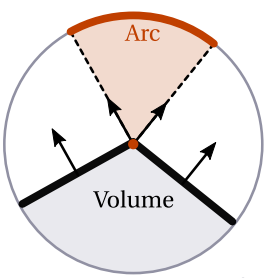
Visibility of Silhouettes and Contours Back in 1967, Arthur Appel tacked the specific visibility issue and introduced the concept of *quantitative invisibility* (or q.i.) which counts for a portion of a line, the number of *visible* surfaces obscuring it [App67]. Only lines whose q.i. equals zero are to be drawn. The rationale behind Appel's approach is to find a compromise between coarse hidden surface removal approaches and expensive hidden point removal techniques. For that aim, he proposes several fast heuristics in order to propagate the q.i. along lines and draws advantage from the fact that a line's q.i. actually changes only when it passes behind a contour curve (it increases when the lines moves towards the visible surface, and decreases when it comes from behind it as shown in Figure 7.23). But prior to its propagation, the q.i. of a contour line still needs to be initialized, often via ray-tracing which significantly slows down the process. Appel's algorithm also unfortunately suffers from instability, visibility error propagation and may get intricate to implement due to the extra care needed for the robust handling of corners.



7.23 3d Line With Different Quantitative Invisibility Values [App67].

Amongst the numerous extensions to Appel's algorithm, Markosian *et al.*'s system for "real-time non photo-realistic rendering" provides one of the most satisfactory solution [MKG+97]. Slightly altering Appel's original definition, Markosian *et al.* determine a line's q.i. by the number of surfaces hiding it, may they be visible or not. It turns out that this modification considerably simplifies the propagation of the q.i. values as for generic views since they now only changes at cusp vertices. Markosian *et al.* also manage to avoid most of the originally needed ray tests for q.i. initialization thanks to their walking procedure.

Accelerating Silhouette and Contour Extraction Along with their robust handling of line visibility, Markosian *et al.* devise a probabilistic approach for silhouette detection. Only a small amount of the mesh's edges is considered. If one of them proves to bind a front-facing facet with a back-facing one, the edge is flagged as a silhouette and the traversal of the sequence of its adjacent edges is triggered. This way, the probability of detection of a silhouette is made directly proportional to their length. Temporal coherence between frames is also exploited to favor the examination of earlier silhouette edges along with their neighbors.

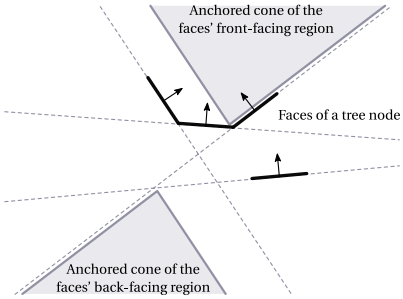


7.24 Gauss map of an Edge [GSG+99].

In their framework for automatic technical illustration rendition of geometry, Gooch *et al.* suggest an alternate solution for accelerating the identification of the mesh's silhouette edges [GSG+99]. They introduce the concept of an edge's *Gauss map* which corresponds to an arc on the sphere of directions defined by the normals of the edge's adjacent faces. Under parallel projection, a plane passing through this sphere's center completely defines the view and separates front-facing from back-facing faces on each of its side. If an edge's Gauss map happens to intersect this view-defining plane, it links a front-facing facets with a back-facing one and is a silhouette edge. Gooch *et al.* build a hierarchy of arcs defined over the sphere decomposed into a set of spherical triangles obtained by the progressive refinement of a icosahedron. Alas, in addition to the assumption of an orthographic projec-

tion, this method is fairly cumbersome to implement and its heavy pre-computation limits its suitability to changing meshes.

For their technique aiming at the real-time rendition of highly detailed geometry using coarser meshes and textures storing the finest variations of reflectance and normal, Sander *et al.* resort to the original model's silhouette edges for achieving convincing results [SGG+00]. While the textures make up for the visual impact of the geometry decimation between the two versions of the same mesh, the outlines of the rendered mesh often betrays its coarseness and irreversibly spoils the illusion. They present a tree-based approach to detect the detailed geometry's silhouettes. Edges are embedded as pairs of directed half-edges into a hierarchical search tree and form individual nodes. Prior rendering, the tree is built by a sequence of node joining operations (either parenting, adopting or merging) ordered in ways to decrease the presumed silhouette extraction cost at runtime. Once created, silhouette edges are effectively detected by a depth-first traversal of the tree. Most of the computations are effectively avoided by the early tests that assess whether or not all the faces associated with the edges of a node are front- or back-facing. These trials are expressed in a dual form that checks the inclusion of the vantage point **e** in one of the two *anchored cones* of the node (*cf.* Figure 7.25). Since the edge hierarchy along with the cones of its nodes are computed in a pre-processing phase, only simple point-cone inclusion tests are performed at runtime and if the two cones of a node are made aligned and of equal opening angle, most computations can be factorized. But as for Gooch *et al.*'s technique, the drawback of such an emphasis on pre-computation is the consequent unsuitability of the approach to changing meshes.



7.25 Sander *et al.*'s Early Test for Silhouette Rejection.

From Silhouette Edges to Silhouette Curves All the aforementioned techniques find their silhouette and contour lines among the mesh's geometric edges. While tolerable if the input model has been meticulously modeled, the direct dependency of the end drawings' quality to the tessellation is not that laudable. Hertzmann and Zorin study the extraction of silhouette and contour curves in the case of smooth surfaces [HZ00], notably by considering the local approximation by an analytic quadratic surface available thanks to the principal curvature-related information. From it, they derive from Equation 7.4 a first linearly-interpolated function whose zero set corresponds to the rim curve, now possibly made of straight line segments crossing the mesh's facets. They also note that for smooth surfaces, stable singularities only include smooth cusps characterized by the fact that the view vector locally become tangential to the silhouette curve. They hence express this second condition as the vanishing of another *cusp function* (corresponding to the radial curvature) expressed in the surface's principal coordinates thanks to the Euler formulae (*cf.* Equation B.2). This finding is in perfect accordance with Koenderink's precedent definition of the end contours. The precise localization of cusps which govern most of the projected contour curves' visibility changes is therefore expressed as the determination of the intersection of these two functions' zero sets. They finally introduce a fast silhouette detection scheme by casting the problem of finding these zero-crossings over the surface into the computation of the intersection points between the mesh's dual surface with the plane corresponding to the viewpoint.

7.2.2.2 Tweaking the Vantage Point

Silhouette and occluding contours sure do convey the most important shape cue such as objects' outlines, yet fail to account for the geometric salient features lying within the closed form of their projected shape. This is one of the reasons they are often used in conjunction with view-independent curves, typically crease lines, in order to alleviate this weakness. A successful trend of work attempts to overcome this deficiency by considering the silhouette and contour curves that could have arisen if the viewpoint slightly changed.

Formulated Silhouettes Driven by the goal to emulate terrain sketches made by professional cartographers, Whelan *et al.* note the visual limitation of the occluding contours, but nevertheless intend to use them as

decorations for their relief-dedicated *profile strokes*. These strokes highlight the abrupt changes in slope of regular sections of the terrain in along and across the line of sight, and succeed to mimic what cartographers would create fairly well [VW93]. But the terrain's silhouettes emerge from the accumulation of disconnected strokes more than from the drawing of explicit curves and would thus gain from being replaced by a cleaner version. Whelan *et al.* propose the *formulated silhouettes* that correspond to a subset of the contour curves that would appear from a slightly lower vantage point and stress surfaces with slope of negative gradient. These curves are subsequently thinned and chained into polygonal chains before their addition to the terrain's profile strokes.

Suggestive Contours

Further exploring Whelan and Visvalingam's intuition, DeCarlo *et al.* propose a new kind of curves meant to extent contour lines within the surface's projected shape so as to communicate shape-related information in the most visually pleasing way [DFRS03]. Explicitly drawing contours that would appear in nearby views, DeCarlo's suggestive contours enjoy at least two distinct albeit complementary definitions, and can be obtained either by way of an object-space algorithm presented hereafter, or an image-space approximation mentioned in Section 7.1.3. The canonical definition of the suggestive contours expresses them as the subset of the loci of surface points for which the radial curvature vanishes while increasing towards the viewer:

$$\text{Suggestive contours} \quad \kappa_r = \kappa_{\mathbf{n}}(\mathbf{w}) = 0 \;\text{ and }\; \begin{cases} D_{\mathbf{w}}\kappa_r > 0 & \text{if } \mathbf{n}.\mathbf{v} > 0 \text{ (front-facing)}, \\ D_{\mathbf{w}}\kappa_r < 0 & \text{otherwise.} \end{cases} \tag{7.6}$$

The constraint on the radial curvature derivative is crucial for breaking the closed loops made of the zero-crossings of the radial curvature and infuse a more natural look to the resulting drawing. Meeting genuine contour lines through the ending contours with G^1 continuity, suggestive contours substantially enhances them. Moreover, DeCarlo *et al.* prove that suggestive contours cannot meet contours at cusps as it would imply that the tangent to the curve would line up with the view direction and impose a null derivative of the radial curvature along $\mathbf{w}$ (which would violate their derivative test).

The second, and proved equivalent, definition of the suggestive contours makes even more explicit their concept of "almost contours" for a given vantage point, and states that they appear at locations where the silhouette function $\mathbf{n}.\mathbf{v}$ does not exactly cancel, but rather assumes a local positive minimum (or a local negative maximum when back-facing) in the tangential direction of sight:

$$\text{Suggestive contours} \quad D_{\mathbf{w}}\mathbf{n}.\mathbf{v} = 0 \;\text{ and }\; \begin{cases} D_{\mathbf{w}}D_{\mathbf{w}}\mathbf{n}.\mathbf{v} > 0 & \text{if } \mathbf{n}.\mathbf{v} > 0 \text{ (front-facing)}, \\ D_{\mathbf{w}}D_{\mathbf{w}}\mathbf{n}.\mathbf{v} < 0 & \text{otherwise.} \end{cases}$$

This alternate definition is particularly convenient for DeCarlo *et al.*'s conception of their image-based algorithm for suggestive contour extraction. While nonexistent for objects devoid on concavities (with a strictly positive gaussian curvature) and somewhat sensitive to viewpoint perturbations, suggestive contours doubtlessly greatly augment the perception of shapes by the extension of their occluding contours.

Time-Varying Suggestive Contours

Temporally-wise however, first examination shows that while the majority of the extracted suggestive contours exhibits a fair robustness during animations, others lines caused by geometric noise and negligible surface features appear much more volatile. Subsequent research by DeCarlo *et al.* thoroughly studies the suggestive contours' temporal behavior via the computation of the velocity of the zero-sets of the radial curvature surface function [DFR04], and derives adapted rules and criteria for line trimming and fading so that suggestive contours likely to be subject to abrupt motions are made less visible. Assumed to move along the gradient of the radial curvature, the speed of the suggestive contours' motions seem directly dependent to the rate of the variations along the surface of the asymptotic directions (the pair of tangential directions for which the

normal curvature vanishes in mono- and anticlastic regions). Their careful analysis reveals the stability of the suggestive contours when coinciding with parabolic lines, locations at which they do not exhibit any velocity. Along these additional tests for limiting temporal artifacts, the transposition of acceleration techniques such as Markosian's stochastic silhouette edge search [MKG+97] and Sander's hierarchical search tree [SGG+00], to the case of suggestive contours is also investigated and met with a varying levels of success: while Markosian's approach lends itself quite easily to the extraction of DeCarlo's suggestive contours, Sander's acceleration structure suffers from poor caching. Suggestive contours have also been used in the context of volumetric data visualization and extracted as the intersection lines between the data iso-surface's and the radial curvature function's zero sets [BKR+05]. The traversal of the 3d data volume is also highly reminiscent to Markosian's accelerated silhouette localization procedure.

Another aspect of dealing with issues arising from dynamically explored scenes is the apparition of line clutter when the object of interest is observed from a too distant vantage point. The question of the continuous handling of level-of-detail in the context of suggestive contour based drawings has been explored by Jeong *et al.* [JNLM05], and further deepened by Ni *et al.* [NJLM06]. Both approaches appropriately smooth the input geometry and build a hierarchy of meshes in such a way that the features described by the occluding and suggestive contours assume a desired screen-space size. Jeong *et al.* adapt Hoppe *et al.*'s view-dependent mesh refinement so that the triangles satisfy a projected size criterion. Edge collapse and vertex split events are triggered by tests considering the average area of the triangles adjacent to a given vertex, and smoothly carried out in time to minimize line popping. Curvature-related information must also be smoothed and updated as the scale of the mesh changes in order to limit temporal artifacts on the extracted suggestive contours. In order to accelerate Jeong's online mesh adaptation, Ni *et al.* instead chose to pre-compute the hierarchy of meshes prior rendering and only select its most appropriate at runtime. The input geometry is consequently view-independently preprocessed and simplified through a retained sequence of edge collapse operations, each of them causing a quantified and stored amount of detail loss. Levels of the hierarchy are explicitly described as fractions of the full sequence of simplification operations, and tensors of curvature and of curvature derivatives are pre-computed for each of them. With respect to Jeong's technique, Ni's exhibits improved performances, notably thanks to the resort to vertex and fragment shader programs, respectively in charge of the interpolation of the radial curvature after selection of the best-suited scale and the drawing of the lines. This last step however places the detection of lines at the scale of pixels and comes at the price of their stylization.

Suggestive and Principal Highlights

Whereas suggestive contours tend to gather in surface regions prone to occlusion and shadowing, suggestive contours are made coincident to the valleys of the front-lit rendition of the geometry –, lines dedicated to the line-based abstraction of more brightly-lit surface areas have also been examined. DeCarlo and Rusinkiewicz propose two new sets of linear features addressing the specific case of highlights: the *suggestive highlights* and the *principal highlights*. As its very name suggests, the former category of lines roughly acts as the symmetrical element of the suggestive contours:

$$\text{Suggestive highlights} \quad \kappa_r = \kappa_{\mathbf{n}}(\mathbf{w}) = 0 \;\text{ and }\; \begin{cases} D_{\mathbf{w}}\kappa_r < 0 & \text{if } \mathbf{n}.\mathbf{v} > 0 \text{ (front-facing)}, \\ D_{\mathbf{w}}\kappa_r > 0 & \text{otherwise.} \end{cases} \tag{7.7}$$

The only difference with Equation 7.6 is the sign of the derivative test, and when considered together, suggestive contours and suggestive highlights almost form back the closed loops of the zero-crossings of the radial curvature across the surface. Ensues from this change the alternate definition of that suggestive highlights as the loci of point where the silhouette function $\mathbf{n}.\mathbf{v}$ reaches either a local *positive* maximum (or a local *negative* minimum depending on their visibility) along the projection of the view direction $\mathbf{w}$.

Comparatively defined, principal highlights are defined so as to emphasize the surface points where the $\mathbf{n}.\mathbf{v}$ surface function assumes a significant local positive maximum (or again a negative minimum when back-

facing) in the tangential direction orthogonal to the projection of the view vector $\mathbf{w}^{\perp}$:

$$\text{Principal highlights} \quad D_{\mathbf{w}^{\perp}}\mathbf{n}.\mathbf{v} = 0 \;\text{ and }\; \begin{cases} D_{\mathbf{w}^{\perp}}D_{\mathbf{w}^{\perp}}\mathbf{n}.\mathbf{v} < 0 & \text{if } \mathbf{n}.\mathbf{v} > 0 \text{ (front-facing)}, \\ D_{\mathbf{w}^{\perp}}D_{\mathbf{w}^{\perp}}\mathbf{n}.\mathbf{v} > 0 & \text{otherwise.} \end{cases}$$

By infolding the directional derivation along $\mathbf{w}^{\perp}$, the first condition of the previous definition boils down to find surface locations where the *radial torsion* τ_g vanishes:

$$\text{Principal highlights} \quad \tau_g = 0 \;\text{ and }\; \begin{cases} D_{\mathbf{w}^{\perp}}\tau_g < 0 & \text{if } \mathbf{n}.\mathbf{v} > 0 \text{ (front-facing)}, \\ D_{\mathbf{w}^{\perp}}\tau_g > 0 & \text{otherwise.} \end{cases}$$

Yet it turns out that the zero-sets of the radial torsion brings out too many locations and as such, DeCarlo and Rusinkiewicz restrict principal highlights to the points for which the view-dependent vector $\mathbf{w}^{\perp}$ aligns with the direction of principal maximal curvature $\mathbf{e_1}$:

$$\text{Principal highlights} \quad \mathbf{w}.\mathbf{e_1} = 0 \;\text{ and }\; \begin{cases} D_{\mathbf{w}^{\perp}}\tau_g < 0 & \text{if } \mathbf{n}.\mathbf{v} > 0 \text{ (front-facing)}, \\ D_{\mathbf{w}^{\perp}}\tau_g > 0 & \text{otherwise.} \end{cases} \tag{7.8}$$

When working alongside the more classical occluding and suggestive contours, highlights lines (typically rendered in lighter shades) finish disambiguating shape by hinting specular occlusions at convexities and therefore often make up for the limited intervention of suggestive contours in these areas. Still since both emerging form the zero-set of the radial curvature, suggestive contours and highlights may line up too perfectly and evoke an embossing effect when displayed over a mid-tone colored rendition of the object. When this effect gets distracting, the authors advise to ignore the suggestive highlight lines and rely on the principal highlights instead.

7.2.2.3 Towards a View-Aware Version of Differential Geometry

Rather than considering lines appearing under slightly differing view conditions, Judd *et al.* instead consider the problem of desirable line finding from its roots [JDA07]. At the light of the perceptual insights recalled in Section III and the artists' intuitive skills for effective line rendition, they instead seek to place lines in ways they would best capture the *perceived* intensity variations. They therefore formulate their line extraction rule as highlighting the loci of surface points where the surface normal is *seen* to be exhibiting a locally maximal rate of variation. Their approach's strength comes from their incorporation of the considerations relative to the *perception* of these variations, and formulation of new view-dependent differential measurements to account for them. They consider the directional derivative of the surface normals along a direction $\mathbf{s}$ taken *on the screen* instead of along a specific tangential direction in the object-space, and include the effect of the parallel projection P in its estimation. Similarly as before, the notion of *view-dependent curvature* at a given screen position $\mathbf{p}'$ along a screen-plane direction $\mathbf{s}$ comes down to a screen-centric version of the shape operator (*cf.* Figure 7.26):

$$\mathcal{Q}_{\mathbf{p}'}(\mathbf{s}) = D_{\mathbf{s}}\mathbf{n}_{P^{-1}(\mathbf{p}')}. \tag{7.9}$$

As Koenderink defined his crease lines [Koe90], Judd *et al.* express their *apparent ridges* as the loci of points where the view-dependent curvature locally reaches a local maximum in its direction of maximal variation:

$$\text{Apparent ridges} \quad D_{\mathbf{t_1}}q_1 = 0 \;\text{ and }\; D_{\mathbf{t_1}}D_{\mathbf{t_1}}q_1 < 0, \;\text{ with }\; \begin{cases} q_1 &= \max_{\mathbf{s}} \left|\left|\mathcal{Q}_{\mathbf{p}'}(\mathbf{s})\right|\right|, \\ \mathbf{t_1} &= \arg\max_{\mathbf{s}} \left|\left|\mathcal{Q}_{\mathbf{p}'}(\mathbf{s})\right|\right|. \end{cases} \tag{7.10}$$

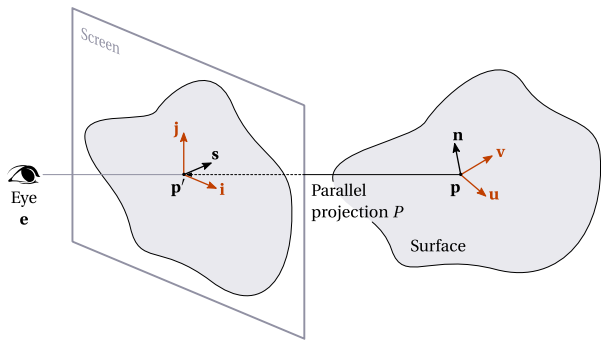
The maximal curvature q_1 and its associated screen-space direction $\mathbf{t_1}$ are found as the eigenvalue and eigenvector of the matrix of the view-dependent curvature operator $\mathcal{Q}_{\mathbf{p}'}$. Despite the fact that q_1 is always positive, the sign of the principal maximal curvature κ_1 at $\mathbf{p}$ enables the distinction between ridge- and valley-like features.

Defined as the composition of the object-space normal derivation operator with the projection's inverse, the view-dependent curvature $\mathcal{Q}_{\mathbf{p}'}$ remains a linear operator.

$$\mathcal{Q}_{\mathbf{p}'}(\mathbf{s}) = \underbrace{\begin{pmatrix} \mathbf{f}_{uu}.\mathbf{n} & \mathbf{f}_{uv}.\mathbf{n} \\ \mathbf{f}_{uv}.\mathbf{n} & \mathbf{f}_{vv}.\mathbf{n} \end{pmatrix}}_{\text{Shape operator}} \underbrace{\begin{pmatrix} \mathbf{u}.\mathbf{i} & \mathbf{u}.\mathbf{j} \\ \mathbf{v}.\mathbf{i} & \mathbf{v}.\mathbf{j} \end{pmatrix}}_{\text{Projection}} \begin{pmatrix} s_i \\ s_j \end{pmatrix}$$

where $\mathbf{s} = s_i\mathbf{i} + s_j\mathbf{j}$.

$\mathbf{f}$ denotes the surface's local parametrization of the tangent plane at $\mathbf{p}$ in the basis $(\mathbf{u},\mathbf{v})$. $(\mathbf{i},\mathbf{j})$ stands for the image plane's vector basis as illustrated in Figure 7.26.



7.26 View-Dependent Curvature as a Screen-Space Normal Differentiation Operator [JDA07].

Apparent ridges command a substantial premium over existing object-space line extraction algorithms for several reasons. Since they account for perspective foreshortening (to the point of exhibiting singularities at theoretical contours), they happen to naturally highlight contours and therefore can work as stand-alone lines. Since their expression roughly follows the crease lines' definition up to the intervention of projection, they also enjoy a comparable level of perceptual relevance. This aspect is attested by the similarity between apparent ridge drawings and "mean" line drawings obtained by the averaging of the response of a Canny edge detector run over several captures of differently-lit scenes. Lastly, since intrinsically view-dependent (contrary to Na *et al.*'s "redeemed" ridges and valleys [NJLS05]), apparent ridges are not plagued by the "paint-on" impression elicited by classical crease lines. To our knowledge and up until now, Judd *et al.*'s apparent ridges seem like the most successful automatic line extraction approach that manages to conciliate the competing goals of shape communication and visual pleasantness.

7.2.2.4 Capturing Illumination Variations

A recent trend of methods recasts object-space line localization into the transposition of edge detection on 3d surfaces. The rationale behind such approaches sounds similar to Judd *et al.*'s: relevant lines follow the surface features causing the most noticeable shading variations. The difference in interpretation concerns the emphasis put on the viewing conditions: while this statement fueled Judd's concern for incorporating the effects of projection in the computation of curvature [JDA07], it motivates here the direct analysis of the shading function itself. The following methods hence study the variation of the *illumination function* I which assumes the following general form: $I = \sum_i \mathbf{n}.\mathbf{l}_i$, where $\mathbf{l}_i$ denotes the light vector at $\mathbf{p}$ pointing towards the ith light source. Aching for the robust tracking of sharp intensity changes over small neighborhoods of pixels, image-based edge detectors often involve filters to estimate the isotropic intensity derivatives and reveal local maxima by tracking their zero-crossings. Thus, in order to translate similar concepts to the processing of 3d surfaces, one has to solve the issue of assessing the derivatives of the scalar *surface* function I. Such methods are often made indirectly view-dependent by having at least one the light sources coincident with the center of projection $\mathbf{e}$ of the camera. The reason behind the sole consideration of the local shading's *diffuse* component is the fact that although specular reflections reveal shape through the patterns of their distortions, they exhibit a higher dependency to the viewing and lighting conditions [FTA04].

Transposing the Canny filter to 3d surfaces The first family of illumination-driven lines are Xie *et al.*'s *Photic extremum lines* [XHT+07], that are defined as the surface locations where the illumination function reaches a local maximum in the direction of its gradient:

$$\text{Photic extremum lines} \quad D_{\nabla(\mathbf{n}.\mathbf{l})}\,||\nabla(\mathbf{n}.\mathbf{l})|| = 0 \;\text{ and }\; D_{\nabla(\mathbf{n}.\mathbf{l})} D_{\nabla(\mathbf{n}.\mathbf{l})}\,||\nabla(\mathbf{n}.\mathbf{l})|| < 0. \tag{7.11}$$

Inspired by the Canny image edge detector, Xie's lines do not handle the curvature surface information as explicitly as earlier methods did. Related concepts nevertheless intervene for the derivation of the surface il-

lumination scalar function $\mathbf{n}.\mathbf{l}$ which reveals the second-order structure of the local surface. Conversely, the third-order derivative necessary for assessing the nature of the detected extremum is computed using a two-pass scheme similar to Rusinkiewicz's for the estimation of the tensor of curvature derivatives [Rus04]. Photic extremum lines produce relatively satisfying results but sometimes fail to faithfully highlight all surface relevant features when the current illumination hides surface details in shadowed regions. This concern can still be alleviated by adding supplemental lights whose optimal layout for a set of specified regions of interest can be predicted by the optimization of the contrast of the global illumination function at these locations. Care must still be provided for choosing the different spot lights' cut-off angles in order to avoid spurious lines arising from a too discontinuous illumination. Another issue of this method is more severe though: when the surfaces are not locally sharp line discontinuities and multiple responses can appear.

Transposing the Laplacian-of-Gaussians filter to 3d surfaces The second category of illumination-based lines are the *Laplacian lines* suggested by Zhang *et al.* [ZHXC09] who define their lines as the loci of points at which the Laplacian of the illumination function vanishes while the magnitude of its gradient exceeds a specified threshold t for preventing response to negligible surface noise:

$$\text{Laplacian lines} \qquad \nabla^2 (\mathbf{n}.\mathbf{l}) = 0 \;\text{ and }\; ||\nabla (\mathbf{n}.\mathbf{l})|| \geq t. \tag{7.12}$$

Again, surface curvature filters through the computation of the Laplacian of the illumination surface function. A key feature of Zhang's technique comes from the following property: $\nabla^2 (\mathbf{n}.\mathbf{l}) = \left(\nabla^2 \mathbf{n}\right).\mathbf{l}$, which permits the confinement of most computations to a pre-processing step. For this phase, Zhang *et al.* employ a the mesh Laplace operator of Belkin *et al.* for its enhanced robustness to uneven surface tessellation. Contrary to Xie *et al.*, the gradients of the illumination function are either estimated using Rusinkiewicz's approach for differentiation on the 3d surface if allowed by the tessellation's quality, or approximated using an optimization problem close to the discrete solving of the Poisson equation. Generally, Zhang's drawings are much similar to Xie's (to the point of sharing its weakness in the presence of corners) which was to be expected at the light of their close definition, but exhibit a smoother and crisper aspect. The major difference comes from the real-time performances of Zhang's algorithm thanks to its pre-processing.

7.2.3 Object-Space Approaches: The Pros and Cons

As for image-space techniques, we now enumerate the strengths and weaknesses of object-space approaches for automatic line rendering. Here follow their main distinguishing features.

Accurate and diverse. Contrary to approaches that adapt to the limited information stored in 2d discrete buffers, object-space techniques profit from the knowledge of the full geometry, its invisible parts included. Their accuracy is therefore naturally better than their image-based counterparts which are plagued with aliasing artifacts caused by the limited resolution and precision of their buffers. In addition, directly considering the 3d surface commands a substantial advantage of allowing the analysis of intricate differential geometry quantities and concepts hardly processable and encodable through 2d scalar arrays. This improved scope of analysis notably explains the much more considerable variety of object-space lines.

Amenable to stylization. One of the key features of object-space lines is how easily they lend themselves to stylization as opposed to drawings obtained by 2d image processing. Whereas the latter consist of a set of independent pixels, the former is usually made of segments running along the mesh edges or across its facets. Chaining these into polygonal chains and further fitting parametric curves to them hence do not require the same level of intricacy as for raster curves. Earliest attempts at line stylization were conducted by Markosian *et al.* in the context of their real-time non photo-realistic renderer [MKG+97]: by introducing width and color variations, by perturbing lines from their initial path via noise function, or by simulating artistic brush strokes using textured quads. Northrup and Markosian further explore the possibilities offered by

object-space lines' propensity to vectorization [NM00] and notably highlight the need to propose stylization procedures that would ensure a constant image-space size of its features. Kalnins *et al.* notably make most of their line stylization alternatives accessible to non-expert users via their interactive *Wysiwyg NPR* interface.

Computationally expensive. In terms of performances, image-based approaches whose complexity depends on the screen's resolution command an undisputable advantage against object-based techniques which in spite of witty early rejection tests and acceleration procedures, require the traversal of the geometry and clearly more complex computations than simple image filtering. Another issue also arises: contrary to image-based methods that naturally handle the issue of line visibility, object-space techniques usually necessitate additional processing in order to assess the effective visibility of the 3d curve portions they produce. Solution dedicated to silhouettes have already been presented [App67, MKG+97] but these proved to be fairly complicated to implement, brittle in the presence of specific vertex layouts and restricted to a limited set of generic scene-enclosing views. The need to handle stylized strokes further complicates the issue as they do not have an infinitesimal width anymore and therefore require "wider" depth tests to avoid inconsistent or flickering overlaps. Nowadays' retained solution to the line visibility problem seems to be Northrup and Markosian's *Id reference image*-based solution that handles line visibility, adjacency and chaining in screen-space [NM00]. By testing the depth of lines against the scene's z-buffer and rendering them in different unique colors used as their *ids* in an off-screen buffer, they determine which curve is visible at any given pixel of the rendered image, as well as its immediate neighbors for facilitating line segment formation. Kaplan proposed heuristics in order to infer the lines' local quantitative invisibility from such item buffer images. Cole and Finkelstein enhance Northrup's initial method in order to prevent aliasing artifacts and introduce the concept of partial visibility estimated using super-sampling of the *Id* reference image [CF08].

Hard to animate. Presented below are several considerations regarding the temporal behavior of object-space lines.

- A first natural question pertains to the temporal behavior of view-dependent curves' shape when the viewpoint happens to change position.
- A second concern emanates from the fact that the majority of these lines either rely on curvature-related measurements or more derivatives evaluated across the 3d surface. These quantities cannot be estimated for each frame. Most methods hence often compute them during a pre-processing stage whose result is not updated during rendering. Consequently, freely deformable objects are left out and cannot have their lines interactively extracted and displayed. Only rigid transformations are supported as the tensors of curvature and curvature derivatives are left untouched by such metric-preserving transformations. This issue of the impossibility of the online computation of the surfaces' differential measurements has recently been tackled by Kalogerakis *et al.* who propose to *predict* their temporal evolution during the course of the animation [KNS+09]. They resort to machine learning techniques such as independent component analysis and regression in order to learn a non-linear mapping from the time-varying shape parameters of the animation to the curvature data. They thoroughly study the cases of skeleton-based and blend-shape animations, as well as cloth simulations. Conversely, image-space lines generated by linear filtering of continuously varying geometric buffers naturally enjoy a relatively satisfactory temporal coherence.
- Finally, a last animation related concern does not involve directly the lines but the temporal behavior of their stylization attributes. This matter becomes all the more severe for view-dependent lines whose shape continuously changes, possibly breaks or merges during animation. Solutions for devising as continuous as possible curve parameterizations, and proposing adapted stroke textures for such unpredictable settings have been proposed by Kalnins *et al.* [KDMF03], and recently extended by Bénard *et al.* [BCGF10].

CHAPTER 8

Machine Learning for Interactive Line Rendering

This last contributive chapter details original research in the context of line rendering of 3d geometry. At the light of existing research presented in Chapter 7, several questions come to mind. The first one naturally concerns the relevance of the lines extracted by previously mentioned methods either in terms of shape comprehension or aesthetics, and consequently their validation. Object-space approaches (*cf.* Section 7.2) especially give rise to such doubts as most of them introduce new differential measurements to assess geometric saliency, and are hence not theoretically directly comparable with earlier attempts. This lack of confrontation with existing techniques, as well as the necessary evaluation of their efficacy for shape communication found their answers with recent user studies presented in Section 8.1. Or maybe lack thereof as no clear conclusion could be met, and no technique deemed universal.

This hence begs a second question directly related to the concerns tackled in this thesis: has a good balance between automation, prior knowledge, and user intervention been reached thus far? Considering the fact earlier methods do not account for users' wishes for line extraction and their perplexing results on certain models, we claim this is not the case yet, and propose an *interactive* framework for line rendering by example. Very recently submitted to EUROGRAPHICS'11, our research is still in its infancy and most of Section 8.2 revolves around the validation of our formalization of line extraction as a binary classification problem.

8.1 Remaining Challenges in Line Rendering

The sheer amount of research dedicated to the automatic generation of line drawings is impressive, and therefore so is the disparity in the proposed solutions, some of them taking drastically different paths in order to reach a same destination. More than the problems of the line identification or of the quantification of their perceptual of geometric relevance, it is the theoretical approaches themselves that become less and less comparable: from the transposition of image processing routines to the analysis of the meshes' differential geometry, singling out any method from the sole consideration of its fundamentals is now unclear.

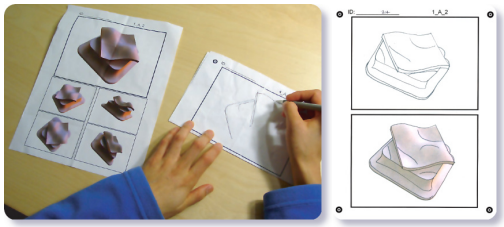
Moreover, the subjectivity behind the assessment of the quality of their outcomes demanded the establishment of proper comparison procedures. Indeed, the gamble with proposing such a variety in distinct and partial solutions is the risk of hindering their actual use from the lack of clear winner. This interrogation recently surfaced in user studies conducted by Cole *et al.* [CGL+08, CSD+09]. In addition to this legitimate inquiry, we also question the place of the artist in the line drawing creation process. In accordance with the present thesis' main concern, the resort to prior models and user intervention, we raise doubts about the rightfulness of only proposing fully-automated solutions for a task as aesthetically-driven as the creation of a drawing.

8.1.1 Automatic Line Drawings: Could Less be More?

Evaluating automatic line drawings is no easy task. But the side-effect of the lack of means of evaluation either in terms of perceptual efficacy or aesthetics results in a *"yet another method"* kind of research where newer techniques outshine previous ones either by their increased performances, the novelty of their shape analysis, or the belief of their authors. It should be admitted however, that some of the most recent takes on the matter foresee and partially fulfill this need of actual comparisons with earlier techniques. Zhang *et al.*'s argumentation in favor of their Laplacian lines [ZHXC09], as well as Kolomenkin *et al.*'s analysis of the behavior

of their demarcating curves [KST08] attest this fact, but mostly revolve around the interplay and the inclusion relationships of their lines with existing ones.

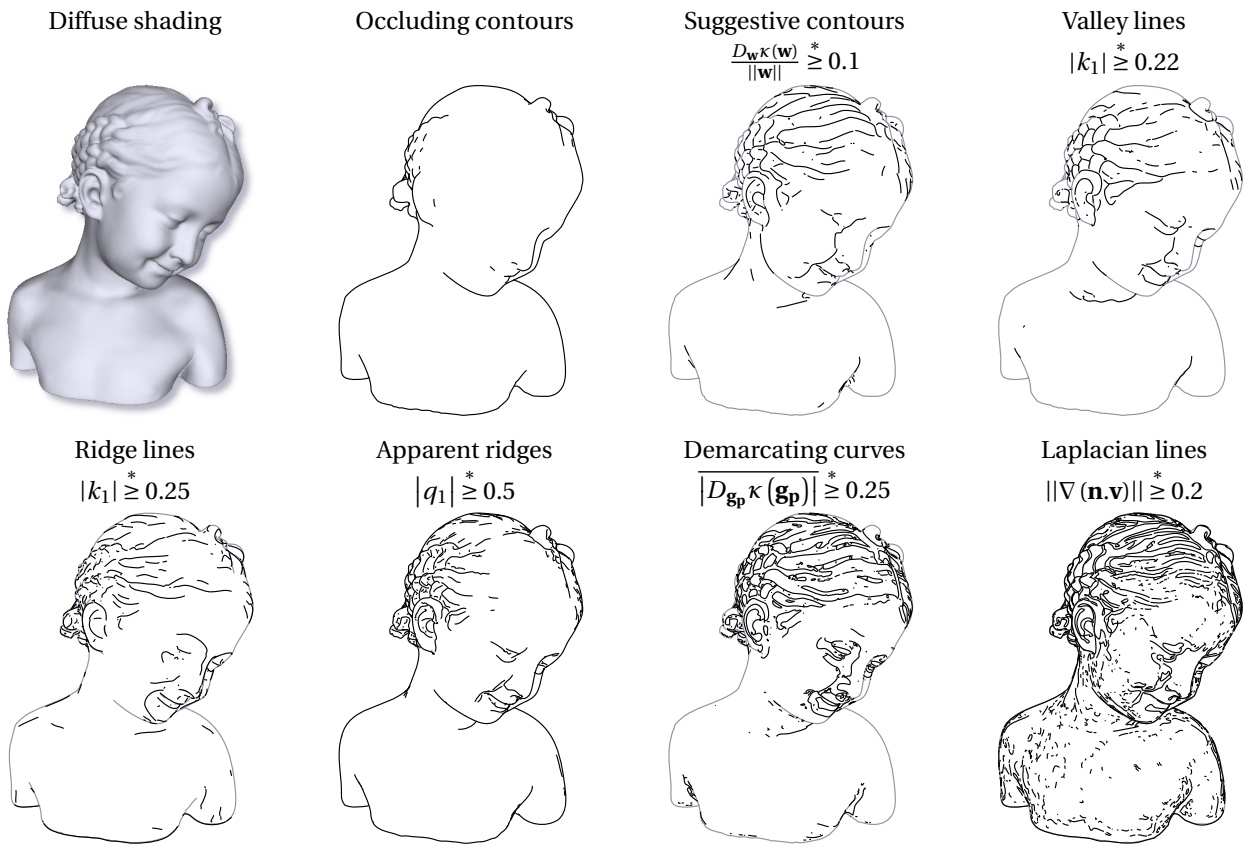
In their seminal study asking the question *"Where do people draw lines"*, Cole *et al.* first lead the way for the validation of computer-generated line drawings, and explicitly tackle the question of whether or not available algorithms keep in touch with what actual artists would do if they wanted to unambiguously describe a shape by way of a line drawing [CGL+08]. Their experiment involves around 29 individuals, all familiar with drawings albeit with different skill level, and consists in having them represent a photo-realistically rendered shape for a given lighting and vantage point via a line drawing (*cf.* Figure 8.1 *(left)*).



8.1 Experimental Set-Up for Cole *et al.*'s *"Where Do People Draw Lines?"* Study [CGL+08].

They gather more than two hundreds of different shapes such as mechanical parts, clothes or bones. The main hurdle of such an experiment if the fulfilment of having manually traced lines registered onto the rendered surface's geometric details. If this condition is unsatisfied, the validity of the reached conclusions would be in peril. Cole *et al.*'s parry resides in having test subjects draw their lines twice, the first time over a blank sheet of paper, the second time over a faint version of the rendered object to enable registration (*cf.* Figure 8.1 *(right)*). The set of retained CG lines used for analysis comprises intensity discontinuities of the diffusely shaded rendition of the shape, view-independent crease lines, Judd's apparent ridges and DeCarlo's suggestive contours (all of them being decorated by the shape's silhouettes and occluding contours). For their statistical analysis, Cole *et al.* draw most of their conclusions regarding the appropriateness of these CG lines from the study of their *precision & recall curves*. Here, *precision* denotes the fraction of the human drawing's pixels lying sufficiently close (below 1mm in screen-space) to the CG drawing's pixels, whereas *recall* designates the amount of pixels from the CG drawing lying as close to the human drawing's pixels. It turns out that ridge and valleys lines (apparent ridges included) do a superb job in most cases, while suggestive contours command a slight advantage for smooth objects such as folded clothes. Gleaning ground-breaking conclusions is here difficult, but one can only praise the attempt at such a careful estimation of most line drawing algorithms at the light of real artists' work.

Hovering over the success of their previous study, Cole *et al.* propose the following year a second even more general questioning *How well do line drawings depict shape* [CSD+09]. They use Koenderink's gauge figure protocol in order to unveil the pictorial relief elicited by the line drawings gathered on the occasion of their past experiment. But Cole's study outperforms any similar earlier attempts by the unmatched number of participants which raises up to approximatively 500 test subjects scattered across the Earth and carrying out the experiment through the Web. Conclusions are here much more thorough than from their previous study. Nuancing Koenderink *et al.*'s conclusions that line depictions could easily prove to be as articulate representations as fully shaded renditions of the object [KDCL96], Cole *et al.* note from the statistical analysis of the angular error distribution that some shape related information is irremediably lost in the ditching of the shading cues as attested by the participants' lower performances in terms of shape accuracy. While the interpretation of lines differ for distinct shapes, a same drawing nonetheless gives rise to consistent interpretations among subjects, even if inaccurate. Certain perceptual biases are also worth mentioning: among these, the positive curvature bias which requires the presence of lines so that to indicate a change of the curvature sign; or the fact that suggestive contours are often perceived as surface inflexion markers. More worryingly, drawings of blobby shapes can be completely misinterpreted, regardless of the employed algorithms, and for the most unfamiliar shapes, even the addition of shading cannot disambiguate the erroneous perception of shape. It also appears that errors are not scattered across the model's surface but are localized in specific areas where algorithms conflict. In any case, line drawings based on the highlighting of specific geometric properties seem to be understood as such, and show comparable performances with human drawings. Consequently, even if none of them can pretend to represent a universal solution to the problem, they still constitute a fairly effective basis for the creation of line drawings.

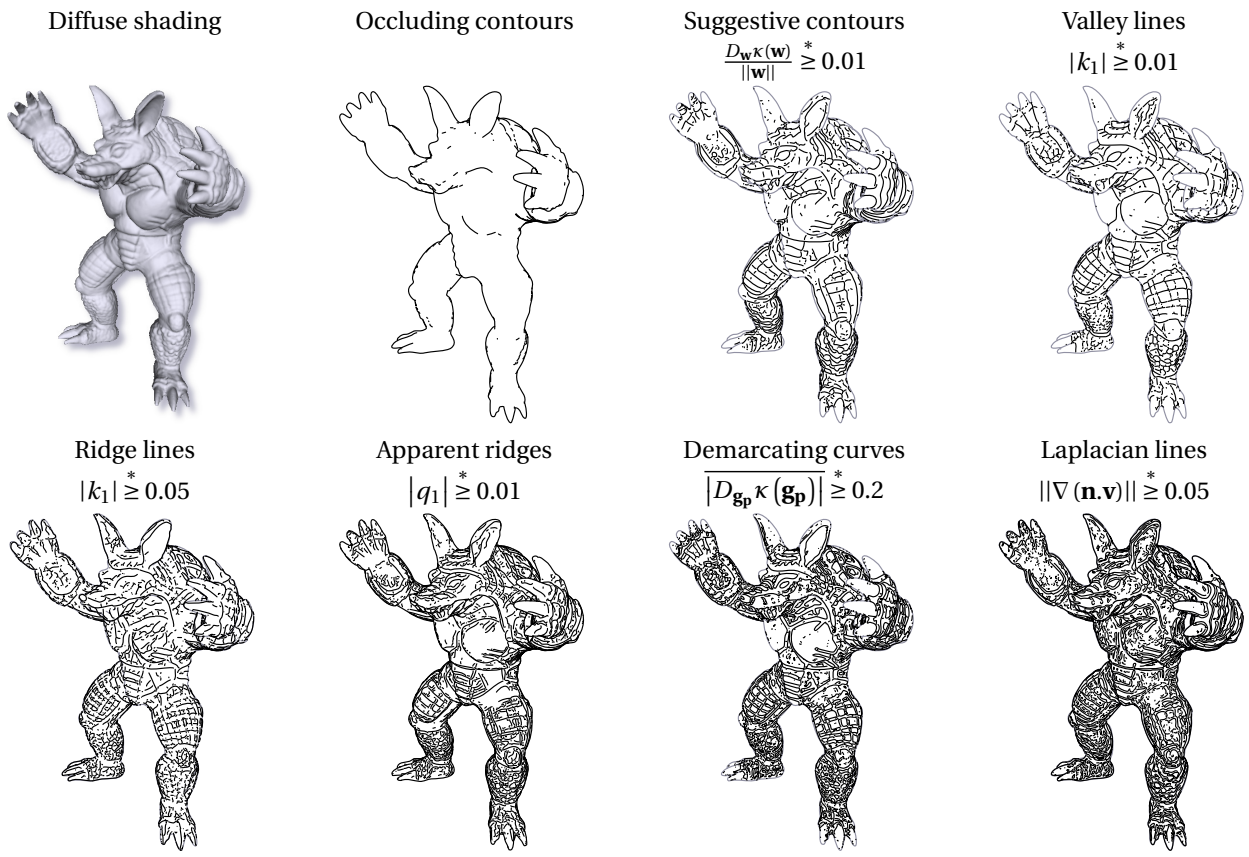


8.2 Line Drawings Generated by Existing Object-Space Methods for the *Bimba con Nastrino* 3d Model. All accompanying thresholds must be normalized with respect to a feature size specific to the surface. Please refer to the main text for more details.

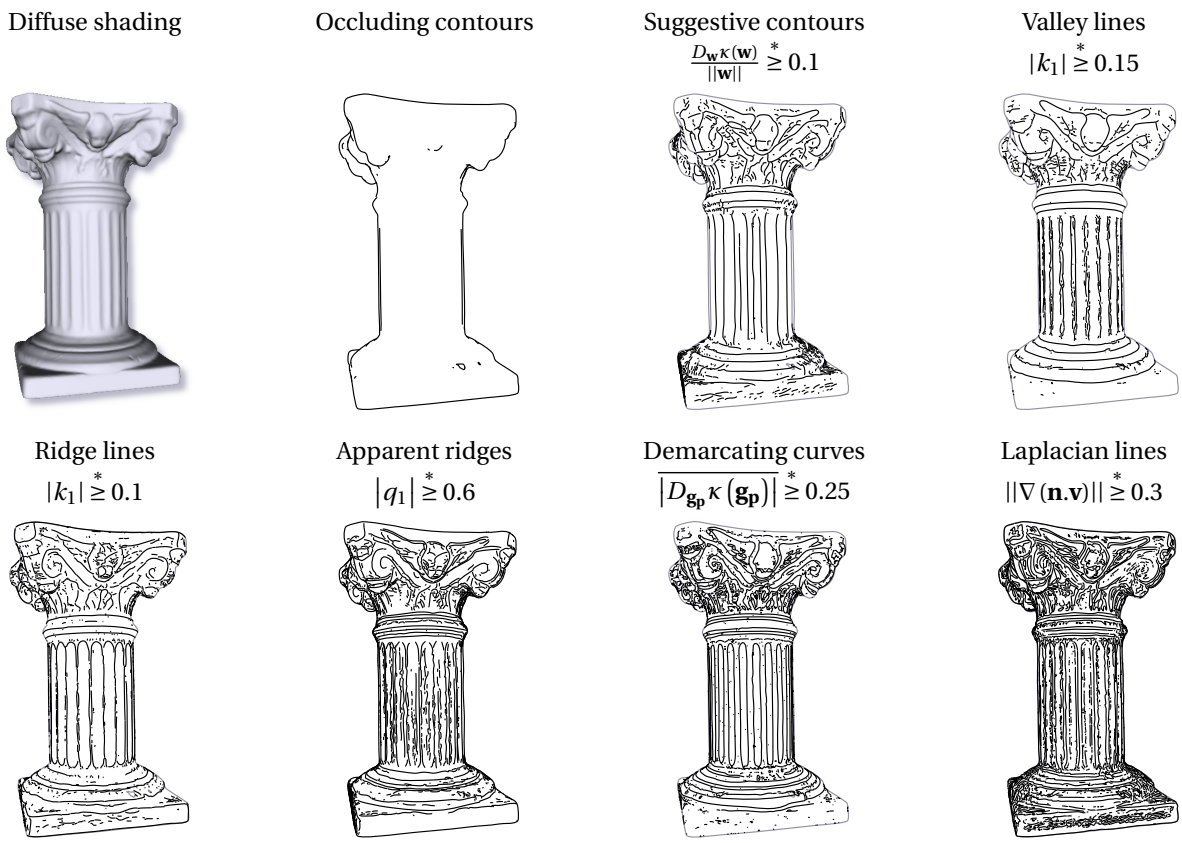
8.1.2 Automatic Line Drawings: a Visual Comparison

Since a comparison on a pure fundamental basis is impossible as success criteria such as perceptual relevance for shape communication and visual pleasantness cannot be computationally characterized, we propose a small gallery of images from Figure 8.2 to 8.7, showing the results of a majority of object-space line drawing algorithms over a as various as possible set of 3d models. The selection of the selected meshes are meant to evaluate the quality of the currently achievable line-based renderings over a wide spectrum of shapes, both in terms of geometrical features and complexity and in terms of familiarity. Henceforth, we present drawings computed over strongly semantically-connoted shapes such as a human figure (*cf.* Figure 8.2) or an animal (*cf.* Figure 8.3), as well as a series of objects. These were picked in order to best illustrate the differences and complementarity of the different methods (*cf.* Figure 8.4) and consider their outcomes over different kinds of geometric behavior, over a smooth (*cf.* Figure 8.5), rectilinear (*cf.* Figure 8.6) or entirely convex (*cf.* Figure 8.7) objects[1].

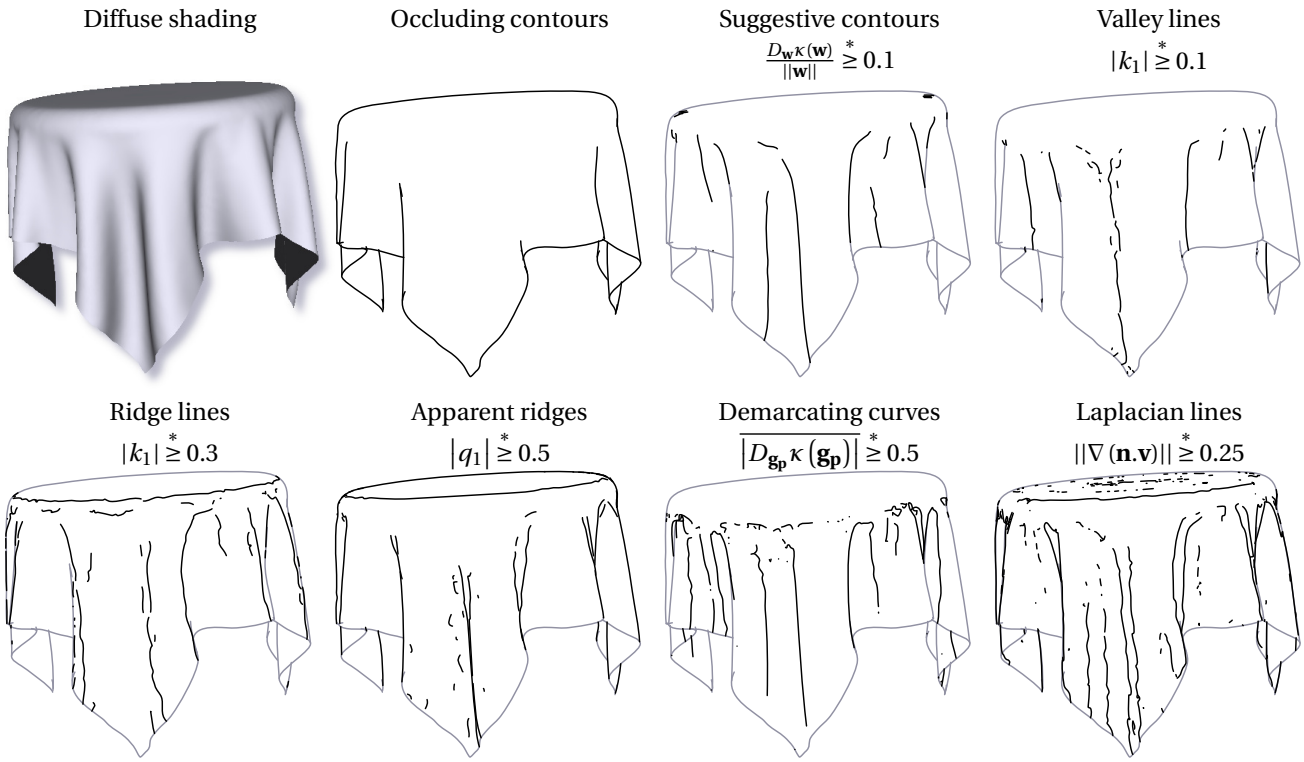
[1] For the majority of the presented techniques, we tried to resort as far as it was possible to their respective authors' original implementations, though most of them required slight code re-writing which naturally may have in spite of all of our attention, introduced technical flaws. We notably thank Szymon Rusinkiewicz and Doug DeCarlo for having made public their curvature estimation and suggestive contour implementation through their *trimesh2* library and *RTSC* software, along with Tilke Judd and Michael Kolomenkin for sharing their code for the computation of the apparent ridges and demarcating curves over triangular meshes. We propose our own implementation of Zhang's Laplacian lines according to the method detailed in their recent article [ZHXC09]. In order to guarantee a fair comparison with all the other methods, we did not incorporate any additional smoothing pre-processes apart from the Gaussian smoothing naturally arising from Belkin *et al.*'s discrete operator when pre-computing the values of the Laplacian of the mesh normals [BSW08]. We opted for an adaptive version of Belkin's formula so that to only consider the mesh vertices' 3-ring neighborhoods. Lastly, we compute the gradients of the illumination function at the zero-crossings of the Laplacian via the optimization scheme proposed by Zhang *et al.*. This step only considers the union of the 1-ring neighborhoods of the pair of vertices from the mesh edge over which the Laplacian of the illumination has been found to vanish.



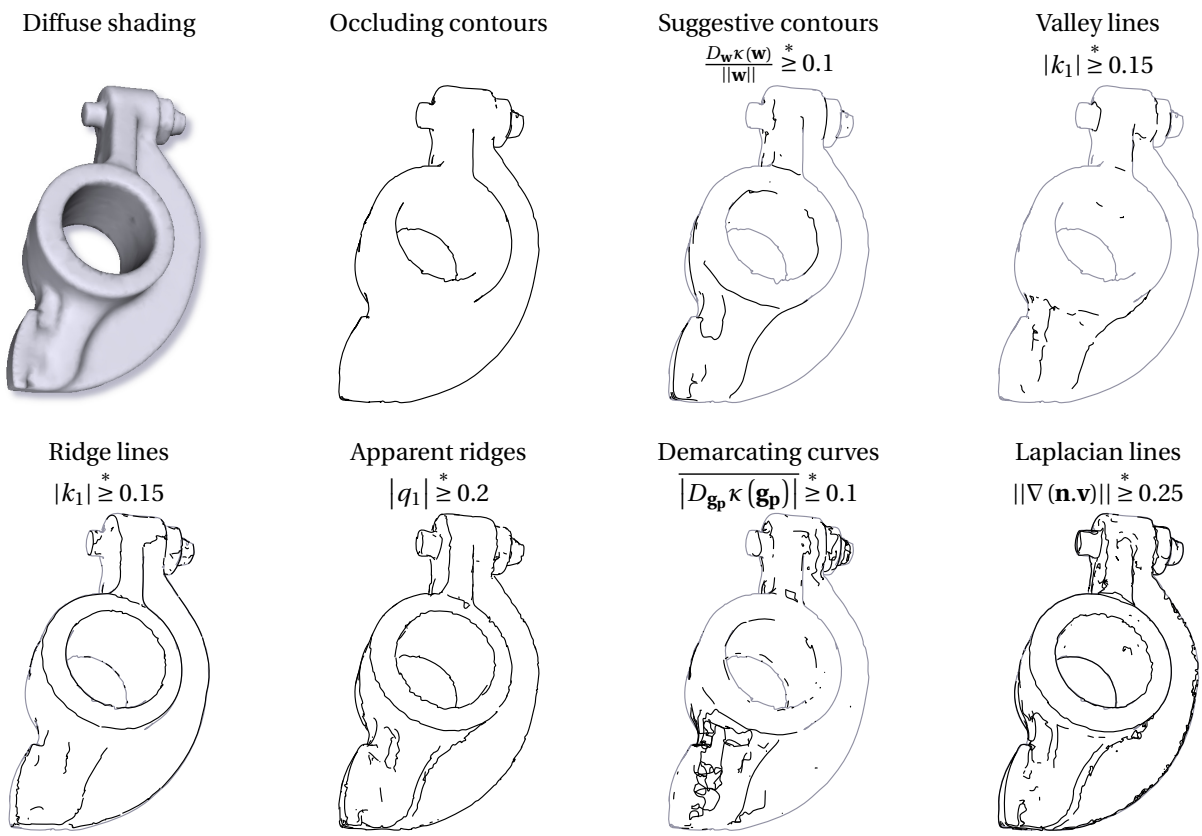
8.3 Line Drawings Generated by Existing Object-Space Methods for the *Armadillo* 3d Model.



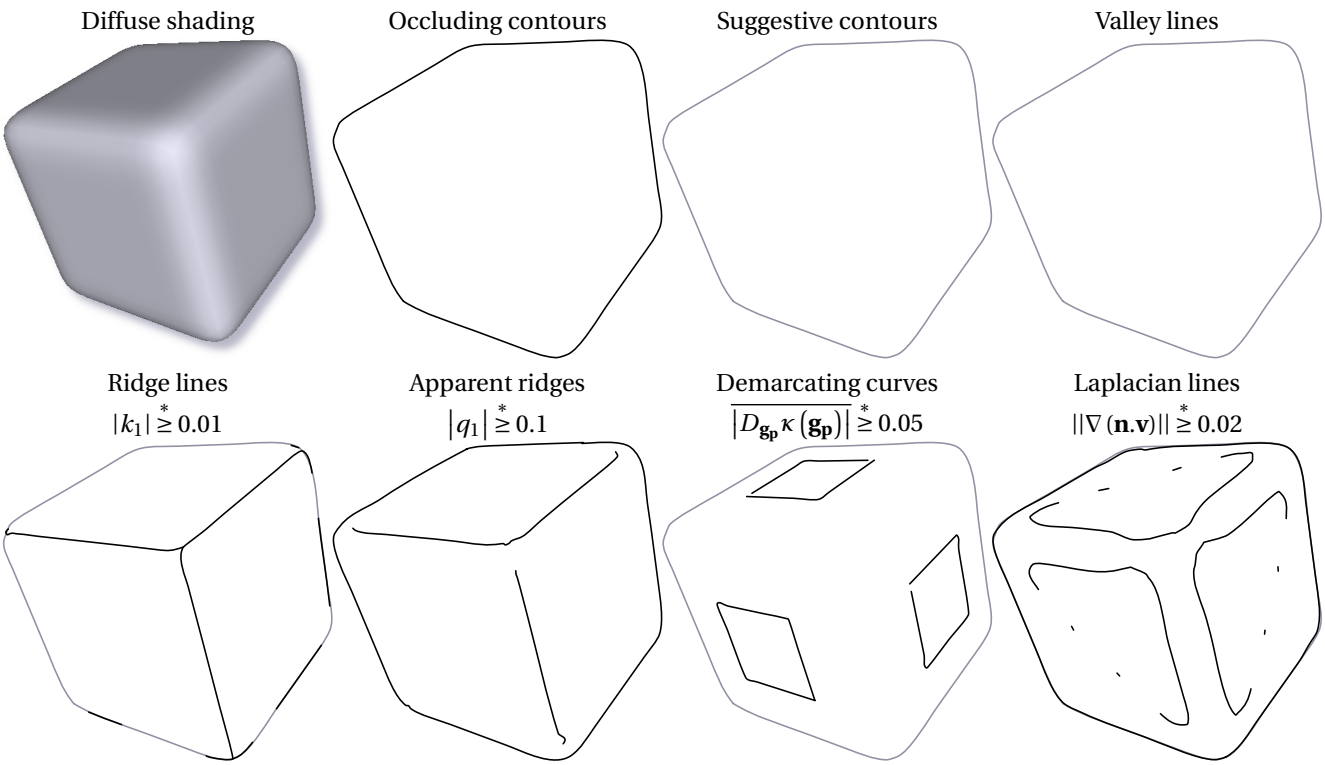
8.4 Line Drawings Generated by Existing Object-Space Methods for the *Column* 3d Model.



8.5 Line Drawings Generated by Existing Object-Space Methods for the *Tablecloth* 3d Model.



8.6 Line Drawings Generated by Existing Object-Space Methods for the *Rocker arm* 3d Model.



8.7 Line Drawings Generated by Existing Object-Space Methods for the *Rounded cube* 3d Model.

Implementation Related Concerns

A small difference in overall quality between our results and the ones found in the publications dedicated to their respective technique finds part of its explanation by the fact we explicitly fit 2d Bézier splines on the list of 3d segments extracted after their perspective projection onto the image plane. Even though tiniest bumps are ruled out by the optimization process used for fitting, the fact we record and display lines in vector form stresses out their small irregularities in a much more pronounced fashion than when viewed as anti-aliased raster curves. Hidden line portions are determined by comparing the depth of each segment once screen-projected against the z-buffer of the scene. Similar in spirit to Cole *et al.*'s assessment of partial line visibility [CF08], we use a higher-resolution off-screen rendered version of the scene's depth buffer. Tests are performed by walking along each segment, each step being separated by a distance of two pixels on the screen. A last reason behind the unflattering aspect of our drawings with respect to the their creators' is that we do not fade then. This voluntary choice has a dramatic impact on the final aspect of the rendering as the most spurious lines which typically have a weaker strengths are not here colored so as it to be unnoticed. They stand out even more as they usually have shorter lengths. We nevertheless decided not to fade these weaker lines as we want to observe the entirely of the detected lines in order to evaluate the relevance of each line detection rule and notice their differences, even at the expense of the immediate appeal of the drawings.

All line renderings are presented along with the threshold values used for their computation. The $\overset{*}{\geq}$ sign indicates the fact that all methods still require the normalization of the user-specified parameters before the direct comparison with the quantities they are supposed to be applied to. Ways of doing so differ between algorithms but have in common the need of finding the processed geometry's "feature size": DeCarlo *et al.* use the hundredth of the 10-quantile of the principal curvature distribution, whereas Kolomenkin *et al.* retain the 80-quantile of the distribution of the magnitudes of the curvature derivatives in the curvature gradient directions. While the curvature thresholds for the crease lines are directly normalized by DeCarlo's feature size, the normalization of the threshold on the radial curvature derivatives for the suggestive contours requires its squaring due to the higher level of derivation. Surprisingly, Judd repeats this last rule in spite of the fact that her line filtering is applied on the magnitude of her view-dependent curvature (which as such would have make it

more comparable to the case of the ridges and valleys). In our implementation of Zhang's Laplacian lines, we use the same normalization approach as for the crease lines since quantities from the surface's second-order structure are considered.

Diving into the Subject

With all these technical precisions made, we now compare the efficiency of the different displayed drawings and draw conclusions on the suitability of the different flavors of lines. Naturally all of them exhibit strengths and weaknesses. **Occluding contours** indisputably convey the most critical shape cue, for all presented examples, as since they highlight occlusions and depth discontinuities are indispensable for figure-ground separation. They are so important that most of the subsequent line detection methods require the composition with these lines (drawn as greyish lines). However, silhouette and occluding contours fail at highlighting most of the features at the interior of the surface's projected shape.

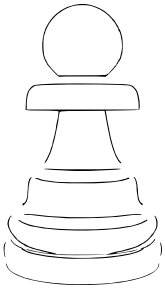
Suggestive contours' main appeal is the seamlessness with which they extend occluding contours, and the substantial improvement they naturally bring to them, especially tangible in Figures 8.2, 8.3 and 8.4. They additionally perform better than other techniques for smoother surfaces (*cf.* Figure 8.5) but can also prove misleading in terms of shape communication [CGL+08] whilst most of the time pleasing to the eye. Their only insurmountable limitation is the requirement of local surface concavities which is a strict condition to their very existence as demonstrated in Figure 8.7. This is explicitly visible when one considers the ending contours' theoretical shape (*cf.* Figure 7.21*(b)*). DeCarlo later addressed this specific shortcoming with the devising of his suggestive highlights.

Valley and ridge lines command a marginal advantage with respect to aforementioned methods in terms of perceptual meaningfulness: valley lines are in perfect accordance with the "minima rule" for shape understanding by visual decomposition into parts henceforth proposing legible drawings even for complex shapes (*cf.* Figures 8.2, 8.3 and 8.4), whereas ridge lines become useful for depicting crest lines whose sharpness should be accounted for in the final drawings (cd Figures 8.6 and 8.7). While compelling, they alas often fail in the aesthetics department which is of primary importance of in the case of their use as expressive renderings: when viewed in motion, they appear as stuck over the object's features and their combined display often result in cluttered results.

Possibly the most elaborated method both in terms of theoretical background and results, **apparent ridges** attempt to keep the perceptual strength of crease lines while alleviating their flaws. As such, most of their associated drawings mostly appear as clever combinations of both kinds of crease lines, and slide along the surface as the vantage point moves. It should be noted that apparent ridges are the only object-space line category which can afford the removal of the occluding contours (the grey lines in Figure 8.5 are boundary lines as the mesh is not 2-manifold). As apparent ridges can encompass linear features assimilable to both valleys and ridges, their outcome often exhibit a higher line density than other line drawings as observed in Figures 8.3 and 8.4 and in the latter illustration, the coexistence of both ridge- and valley-like lines again generate clutter. While the line detection procedure by Judd *et al.* allow apparent ridges to incorporate curves directly comparable to silhouettes, it comes at the price of their smoothness and robustness with respect to the surface' tessellation as Judd's view-dependent curvature is theoretically undefined at true contours. This visual inconvenience stand even more out once splines are fit to the extracted polygonal chains (*cf.* Figures 8.6 and 8.7) and when the viewpoint changes.

Demarcating curves are somewhat harder to directly compare to the other line sets, due to their diverging primary objective of shape depiction in the context of archaeological artifacts' description. View-independent, they present the same disadvantage as crease lines when examined under a moving viewpoint and since they run in-between convex and concave regions, sometimes produce unexpected results, similar to the ones obtained using the surface's parabolic curves (*cf.* Figure 8.2). The displayed drawings do not do much justice to the strength of the demarcating curves as they are meant to work in pair with the mean curvature shading of the surface to reveal their true potential. Yet most of them still are fairly satisfactory especially when the processed surface exhibits bumps and folds (*cf.* Figure 8.3, 8.4 and 8.5). The *Column* model is especially inter-

esting here as while definitely related to ridges and valleys, demarcating curves propose the most compelling result with clearly defined closed curves and without any line redundancy. By definition, they still cannot highlight extruding and protruding features and theoretically share the same weakness as the suggestive contours as the surface should offer concave regions so the demarcating curves to be detected (*cf.* Figure 8.7, note the extremely low threshold used in order to have lines beings retained).



Laplacian lines are presented as achieving drawings of the same caliber as the apparent ridges by their authors. It unfortunately turns that their method is very sensitive to the quality of the mesh tessellation and may require more surface smoothing passes in order to remove all the spurious lines our implementation of their work ends up extracting. Since most of the letdowns of their corresponding results may be caused by our flawed implementation, we are to nuance our observations and judgements, and that the management of their strength as the integral of the illumination gradient's magnitude may greatly improve their appearance by making the strongest lines stand out. It also appears from our experiment that Laplacian lines sometimes seem to line up with the silhouettes but only under a limited number of situations (when the mean curvature stays roughly constant along the line of sight).

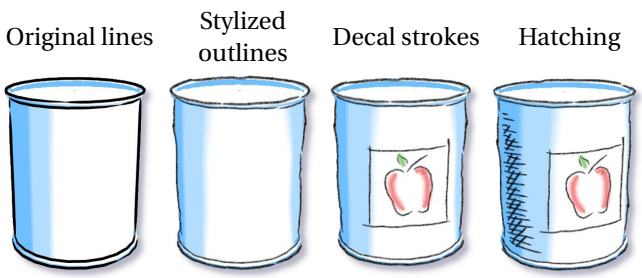
All these illustrations clearly attest of the non-existence of an universal solution, the *Column* examples presented in Figure 8.4 being in that sense the most edifying: not only there is no straightforwardly perfect way of choosing among the different sets of lines, but also different portions of the model may require distinct treatments (between the body and the capital of the column). Are missing from this short comparison Kolomenkin's **relief edges** and Xie's **photic extremum lines**, but since they are theoretically much related to the demarcating curves and Laplacian lines respectively, we believe their omission does not compromise the previous statement.

8.1.3 Putting the Artist Back into the Loop

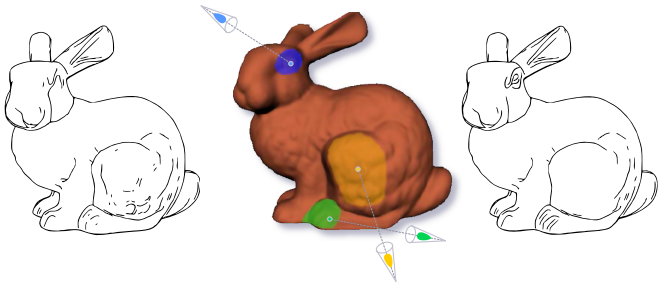
At the light of the insights gathered from Cole *et al.*'s studies, we can wonder whether the commitment to the complete automation of the generation of line drawings is the best path to follow. While the seeking of a specific differential geometric behavior makes sense, the fact errors tend to gather in localized areas [CSD+09] hints the possibility of a fairly manageable correction by the user to dissipate ambiguous and spurious lines while adding missing ones. When considering the literature detailed in Chapter 7, the nonexistent place reserved to users is more than perplexing given the impenetrability of their conception of a satisfactory drawing.

At the best of our knowledge, apart from the hardly predictable tweaking of global threshold parameters, only a couple of techniques give limited controls users in order to mold their own drawing and add their fine touch to the final results. Kalnins *et al.*'s *Wysiwyg NPR* system sure offers several user-dedicated tools in order to annotate models and decorate them with crease, decal and hatching strokes [KMM+02]. But these inputs, while adapted to new vantage points if needed, are meant to be stand-alone additions to the rendering and not to modify or correct its execution by any way (*cf.* Figure 8.8). In contrast, Xue *et al.*'s illumination-driven photic extremum lines gift the artist with a more albeit indirect freedom [XHT+07]: indeed, by allowing the manual placement of supplemental auxiliary lights, the system can process this user feedback and draw additional lines out of it (*cf.* Figure 8.9). However, Xue's controls while more involving to the system, are also more meditated than Kalnins's who invite artists to "draw strokes directly on 3d models". Ideally we would like to devise a system able to truly process user-feedback and adjusting its course accordingly while staying as intuitive as possible, notably by proposing control handles manipulating strokes.

The last and closest attempt at interactive line drawing creation is Lum and Ma's *"expressive line selection by example"* that infers from the user's feedback the visibility of a set of candidate lines extracted from the considered geometry [LM05]. Casting the problem of line selection as a binary classification issue (the visible *versus* invisible lines), Lum and Ma resort to two well-tested machine learning methods in order to reach their goal: neural networks, and support vector machines. The candidate lines whose visibility is to be learned from the user's inputs include depth and surface discontinuities, along with suggestive contours and silhouette

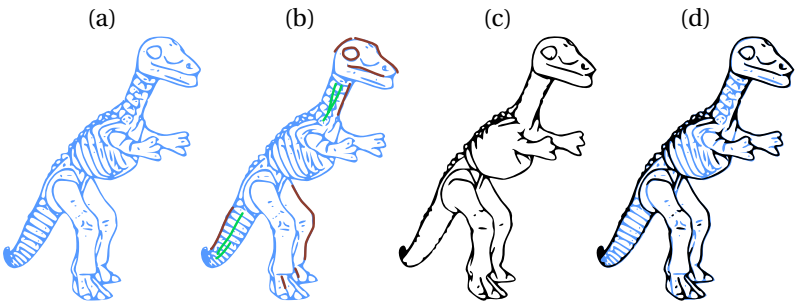


8.8 User Controls Provided by Kalnins *et al.*'s *Wysiwyg NPR* System [KMM+02].



8.9 Xie *et al.*'s Line Drawing Edition Through Light Placement [XHT+07].

edges. Their corresponding geometric properties are henceforth gathered in order to form feature vectors to be subsequently discriminated. Two versions of their algorithm are proposed: an image-based approach relying on feature vectors made of the concatenation of the values lying at the vicinity of a screen pixel, and an object-space algorithm which conversely examines the geometric information available at each mesh edge. Their results are promising but their framework suffers from several limitations: their object-space approach only restricts output lines to the mesh's edges; their style capture only resolves around the notion of line visibility and consists in the removal of unwanted strokes without the proposal of new ones; finally, user's feedback directly has access to the candidate lines in order to strengthen or discard them. We would like to instead have artists start their drawing and have our system anticipate it. Lastly, in both versions of their techniques, identifiable lines are either broken down in pixels, or in mesh edges that are independently classified, we believe that the loss of the continuity of the actual lines is undesirable as more robust classification rules could be inferred if these were treated as a whole instead.



8.10. Lum and Ma's Line Selection by Example. All images are redrawn from [LM05] and show the interaction between the user and Lum's line selection system. From a set of automatically extracted lines detected using earlier automatic line drawing techniques *(a)*, their system learns the line visibility function from strokes drawn by the user (green strokes flag lines to hide, whereas brown strokes stress lines to keep *(b)*). Strokes deemed relevant by the system can either be visualized alone *(c)*, or superimposed with the entirety of candidate lines for further correction *(d)*.

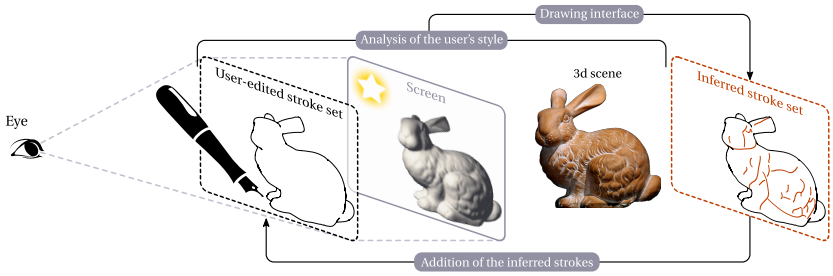
A similar classification based perspective for line rendering has also been explored by Cole *et al.* for processing their collections of drawings [CGL+08]. They indeed use decision trees to unveil the geometric quantities that would best predict the "mean" drawings obtained by the averaging of the artists' work for a same shape. Our approach is definitely related to theirs although we strive to propose a fully interactive alternative.

8.1.4 Our Proposal

The underlying claim behind this last part of my research is the following: in order to obtain a *satisfactory* computer-generated line drawing, one cannot overlook users and should instead place artists' feedbacks and interactions with the generative system at the core of all considerations. Therefore, the research detailed herein exhibits a much more pronounced reliance to user interactions than any of the other techniques presented so far. Our ultimate goal is to provide an interactive, example-based line rendering method that behaves like a line-based, "non photo-realistic rendering filter" directly applicable to any specific geometry.

As we saw with the analysis of most existing line rendering techniques presented in Chapter 7, in the absence of any semantic knowledge or fixed aesthetic conventions, most line drawing generation methods usually favor the analysis of the geometry as the primary source of information for the discovery of their lines. Our approach would be no exception, at the only difference we would have its rendering decisions

inferred rather than predetermined. We thus venture even further on Lum *et al.*'s path by casting the whole issue of line *extraction* as a binary classification problem, and go beyond the question of their sole selection. Figure 8.11 presents an overview of our method's work flow concept, with users directly annotating the geometry projected onto the screen and our technique proposing new strokes in accordance with the drawing style it would have captured so far.



8.11. Overview of the Work Flow of our Line Drawing Technique. We aim at proposing an interactive system working on a trial-and-error principle. Users would trace example lines over a shaded rendition of the geometry and then let the system infer and anticipate their wishes. In case of lacking inferences or false positive responses, corrections should also be made possible and be appropriately processed in order to further polish the system's computational representation of its users' styles.

8.2 Line Drawing Creation as a Predictive Process

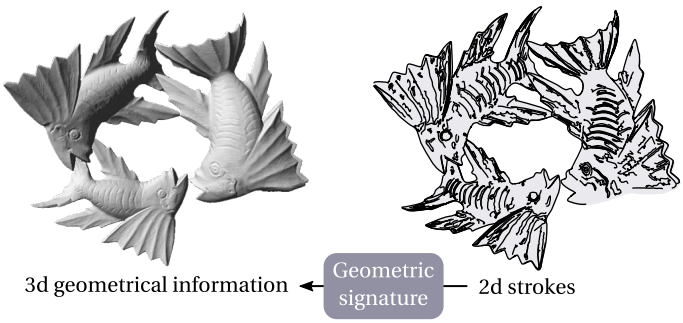
As alluded in Section 8.1, our ultimate goal of providing an interactive framework for line drawing creation is closely related to Lum *et al.*'s take on the issue of allying the machine's computational strengths with the user's aesthetical control of its user [LM05]. They manage to propose a first successful step towards this goal by resorting to classical machine learning tools and cast the issue of interactive line *selection* as a binary classification problem. Our approach is strongly related to their gait as we also resort to classification-dedicated methods in order to embed the user's progressive feedback into our line drawing creation algorithm. Our major contribution lies in the fact that we intend to employ machine learning, not only for the assisted selection of a strokes among a set of existing candidates, but for the *creation* of brand new strokes respecting the inferred user's style.

We henceforth propose a final, more user-centric example-based content generation technique than the ones we proposed so far in this thesis. Follow in the remaining of this Section details concerning the theoretical choices and implementation concerns behind our approach: Section 8.2.1 is mostly dedicated to the explanation of our representation of the information enabling the analysis of a drawing's strokes from a geometrical perspective. Section 8.2.2 then dwells on the classification method we retained for the fulfillment of our classification objectives. An intensive collection of results aiming at the validation of our overall framework is later presented in Section 8.2.3 in order to assess the suitability and relevance of our suggested work to its end purpose. Finally, once thoroughly validated, our analysis stage guides the new stroke generation process that is summarized in Section 8.2.4. Due to lack of time, this last step is not as polished as it could be and further experiment would be required for its completion. Therefore the complete integration of our technique in a real-world interactive application is not achieved, making the therein presented material more a proof of concept than finished work. Its limitations, shortcomings and possible extensions are all mentioned in the next Section 8.3.

8.2.1 Describing Strokes by their Geometrical Footprint

From a higher perspective, our objective is to recast the problem of the users'style non-parametrical capture through classification by having our method be able to discriminate relevant *versus* irrelevant strokes with respect to their ongoing drawing. As users draw over and correct our technique's automatic proposals of new strokes, classification can be updated and further refined so that to better represent users' decisions.

The very concept of style is highly misleading, some clarifications are therefore necessary. By users' style, we designate their choices behind their drawn strokes' locations across the 3d surface, not their appearance



8.12 Stroke Characterization by their Geometric Footprint. The *geometric footprint* of a given stroke designates the set of measurements lying under the 2d positions of its paths on the screen, taken from the 3d geometry after perspective projection. The careful selection of the said measurements is paramount for the success of our approach, for they will represent the stroke's geometrical "signature" and directly condition our method's learning capacities.

attributes *per se* such as their local width, jitter or path patterns such as dashes. Bearing Strothotte's distinction between a stroke's path and style in mind [SS02], we are mostly in the automatic guessing of the users' decisions concerning to former and leave the capture of the later as future work.

In our proposal, we assume the stroke placement by the users is more motivated by efficient shape communication and geometry-related concerns than true aesthetic licence. Our issue then comes down to manage to unveil the hidden relationships that relate the stroke and specific unknown characteristics of the behavior of the geometry lying underneath (*cf.* Figure 8.12). In order to guess of what consists this "specific behavior", we first need to isolate a set of several surface-related measurements to monitor and devise robust ways to associate these with the drawing strokes. Indeed, the combination of these quantities is to constitute each stroke's "geometric footprint", a summary of the evolution of the geometric measurements beneath its path. These geometry-infused stroke descriptors will constitute the data set upon which regression is to be performed later on. As such, their computation is of primary importance as they are at the core of the connection between the drawing and its geometrical interpretation by the method, and hence have a dramatic impact on the generative performances of the obtained inferences. If not chosen carefully, the inferred rendering rules would indeed be likely to fail to identify the users' intent and prove useless when the drawing is to be extended.

8.2.1.1 Selection of the Stroke Features

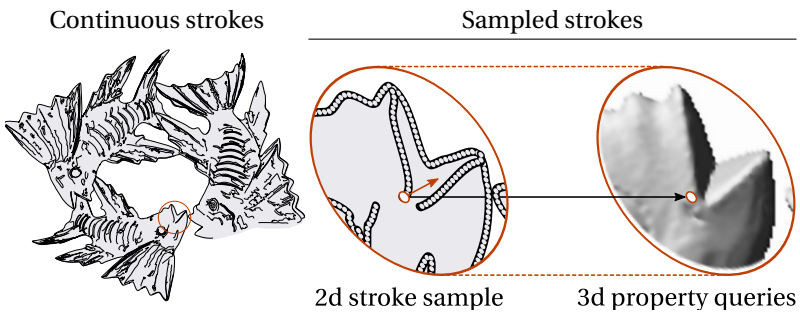
At the light of existing successful techniques for automatic line drawing extraction detailed in Section 7.2, we can focus our attention to a limited set of potentially useful scene features involving not only the surfaces of the models, but also the viewing and lighting conditions. Below is an itemized list of the measurements we consider for evaluating a specific stroke's feature vector. As we consider the scene from the perspective of the stroke to be described, the performed measurements can account not only for spatial positions alone, but also for the tangential directions of the stroke at those locations. We examine the following quantities:

- **Surface slant**: Defined as the dot product between the view vector and the surface normal, we expect it being useful for the detection of contour-like user strokes, may they be true occluding contours or even suggestive contours.
- **Shading diffuse component**: We compute it as the sum over all the lights present in the scene of the light vectors and the surface normal. Its direct consideration may help us to isolate curves whose path roughly follows iso-photic curves' paths.
- **Screen-space distance to the closest object boundary or occlusion**: Complementary to the surface slant feature, we expect this distance measure to further enhance our detection of silhouette-following user inputs and overcome the difficulties arising from the extremely localized nature of such curves – namely their dangerous closeness to the empty background –.
- **Maximal and minimal principal curvatures**: Chief among our curvature-related measurements, the surface's principal curvatures are indispensable differential geometric quantities for the detection of crease lines which, considering the perceptual relevance of valley lines, are to play a crucial role for our method to provide compelling predictions.

- **Normal curvature in the direction of the stroke's tangent**: Complementary to the stroke direction-independent principal curvatures, the normal curvature along the stroke's path will also contribute to the detection of many curvature-driven lines.
- **Gaussian curvature**: By the tacking of the zero-crossings of the principal curvature product, we could identify parabolic curve-resembling strokes.
- **Radial curvature**: This valuable view-dependent quantity is to prove useful when our methods get confronted to strokes by the user coinciding with the surface's suggestive contours or suggestive highlights [DFR04, DR07].
- **Judd *et al.*'s view-dependent curvature** [JDA07]: Perhaps harder to robustly handle than the previously mentioned curvatures due to the singularities arising during their computations, view-dependent curvature is still one of our only screen-space measurements and is likely to participate in the finding of all the kinds of lines apparent ridges encompass, from crease lines to contours.
- **Normal curvature in its gradient direction**: Independent from both viewing conditions and directions of the strokes, this measurement is directly inspired from Kolomenkin's demarcating curves extraction technique [KST08] and its addition to our set of all other curvatures further increases to chances to detect lines that do not follow the outline of all previously mentioned curves.
- **Angular difference between the stroke's tangent and the screen-projected directions of principal curvature**: Girshick *et al.*, as well as Hertzmann *et al.* praised both the perceptual relevance and naturalness of lines whose paths followed the depicted geometry's directions of principal curvatures [GIHL00, HZ00]. By having a look at the evolution of the angles formed by the strokes' local tangents and the surface's underlying directions of principal curvature, we could therefore detect such lines.
- **Magnitude of the perspective depth derivative across the curve**: The monitoring of the strengths of the depth discontinuities along the stroke's local normals facilitates the detection of silhouettes and occluding contours. Inspired by Saito and Takahashi's image-based technique for the non photo-realistic depictions of 3d models [ST90], we favor their perspective depth over the eye-space z-coordinate for our analysis for it accounts best for it better ensures the linearity of the depth estimate on the screen.
- **Derivatives of the principal curvature in their respective principal direction**: In the context of automatic crease line extraction, studying the sign of the maximal curvature derivatives enables the distinguishing of valley lines from ridge lines [Koe90]. As such, we believe the addition of these quantities may bring insightful and complementary information to our stroke feature vectors.
- **Derivative of the radial curvature along the line of sight**: Following the same rationale as for the principal curvature derivatives, incorporating values accounting for the derivative may enhance our classifier's ability to handle suggestive contours and highlights [DFR04, DR07].
- **Derivative of the view-dependent curvature in its direction of maximal variation**: Same as before, only transposed to the case of Judd *et al.*'s apparent ridges.
- **Surface slant Laplacian**: Exploiting Belkin's approximation of the Laplacian operator on discrete surfaces [BSW08], the second derivatives of the surface slant is likely to facilitate the identification of lines highlight preeminent surface features while taking the viewing conditions into account. We notably expect this supplemental feature to prove particularly efficient against the surface's Laplacian lines [ZHXC09].
- **Shading diffuse component Laplacian**: Computed similarly as the Laplacian approximation of the surface slant, this second-order quantity is added to the stroke's bag of features in order to effectively characterize the lines following the shading discontinuities such as the photic extremum lines [XHT+07]. As such lines have been experimentally proved to also constitute a remarkably efficient while economic way of communicating shape [KDCL96], we are to expect that users will favor such locations for placing their strokes. Enabling our classification tools to detect these is therefore of capital importance.

8.2.1.2 Retrieval of the Stroke Features

As mentioned above, we intend to devise a concise albeit discriminative description of the users' strokes by way of measurements performed on the parts of the 3d surface they lie upon. We will therefore sample quantities from the visible parts of the 3d models' geometry according to the 2d path described by the strokes on the imaging screen. For that aim, we associate to each considered stroke a *sequence* of sampling positions running along its course that we obtain by walking along their path at constant screen-size step of 2 pixels (*cf.* Figure 8.13). Either obtained from the direct hand gestures of the user or from object-space line extraction algorithm, all strokes are being fitted splines using Schneider's method [Sch90]. As a chain of Bézier curves, we compute the 2d positions of the geometric samples their to their arc length parametrization and can also estimate with precision the strokes' local directions thanks to their tangents at the considered locations.



8.13. Stroke-Guided Sampling of the Projected Geometry. For characterizing each stroke of a drawing and take decisions regarding its description, we need to consider and concisely encode the behavior of our differential geometry-related measurements along its course. For that aim, we evenly sample each curve after it has been parameterized by its arc length (each step corresponding to 2 pixel-wide spaces in all our examples). Along with the resulting sample positions, we store the stroke's tangents at the corresponding locations. These directions are useful for several directional queries (such as the normal curvature in the stroke direction) as well as for the assessment of stroke-consistent sets of differential quantities.

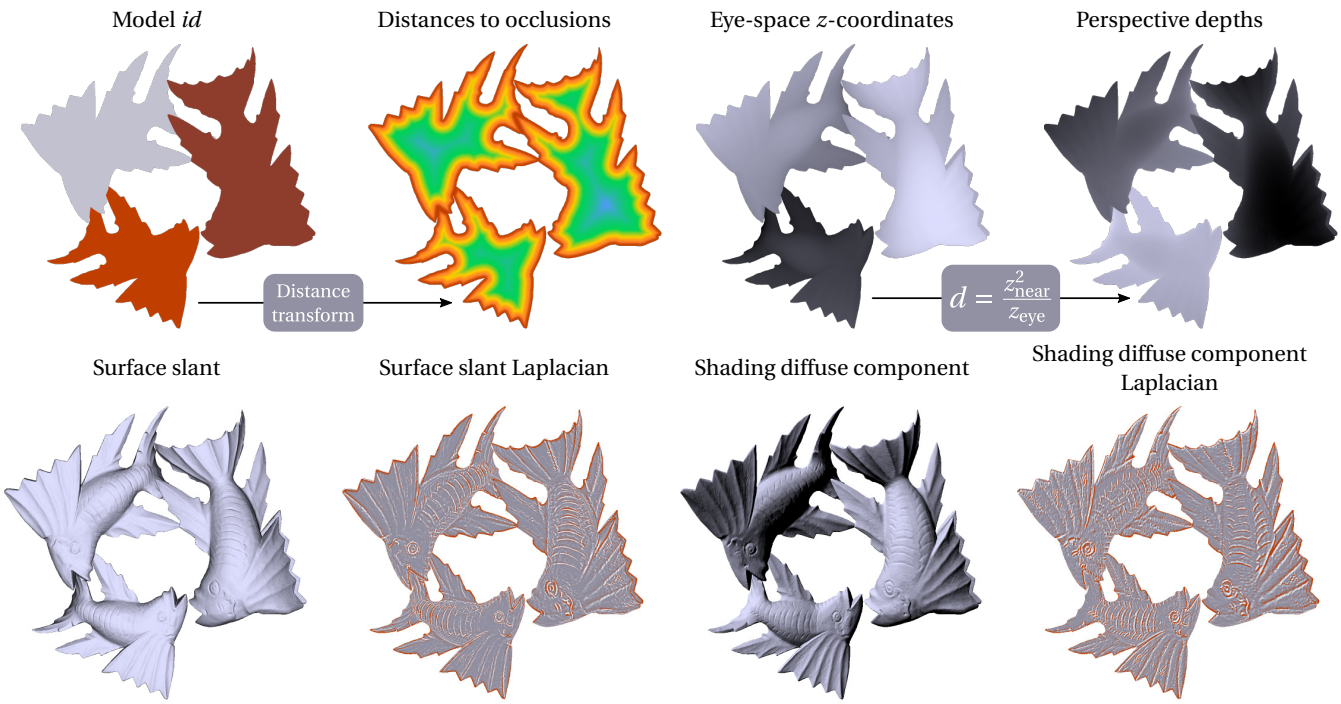
Querying the Geometry

We thus now need to consider parts of the 3d models' geometry that project through perspective projection onto the previously obtained 2d positions. While the option of casting a ray passing through the camera's center of projection and each screen sampling position, tracking its intersections with the geometry, and considering the surface's local surface properties at the visible intersection point would lead to the most accurate set of measurements possible, we believe that the resort to such a computationally intensive solution would compromise our perspectives of interactivity with the user. We therefore favored an image-based approach directly inspired by Saito and Takahashi's approach [ST90], and instead store all the information necessary for the computation of our stroke features into screen-aligned G-buffers – G as coined by Saito and Takahashi, standing for geometric.

The stroke-guided sampling of the surface properties then only boils down to the direct access to a two-dimensional array and greatly alleviates the computation costs the first alternative would have required. All buffers are established through off-screen rendering and brought back to the CPU for analysis so as to process them and build our strokes' feature vectors out of them. Data repatriation from the graphical hardware is known for dreadfully plaguing performances in the context of rendering applications, but in our case it is only performed once as we require the user to fine tune the viewing and lighting conditions before starting drawing.

In order to prevent the undesirable effects of a possible direct dependence of the buffers' resolution to the imaging screen size, all our buffers are of a different – ideally larger – fixed resolution. Quantization-related artifacts are mitigated by the internal representation of these G-buffers as floating textures. Similarly, in order to limit aliasing when these buffers are being queried at sub-pixel precision locations, bilinear interpolation is resorted to.

Figures 8.14 and 8.15 displays the collections of buffers directly obtainable from off-screen rendering. These quantities are by default interpolated across the triangulation faces of the models through trilinear interpolation from the data stored at their vertices. Special care has nevertheless been provided to several measurements. Of most importance, the principal directions of maximal curvature for which such an interpolation scheme would have made no sense. We hence trade in this case, trilinear interpolation between scalar values for the symmetrized two-way spherical interpolation between a set of three quaternions. The princi-



8.14 Extracting Geometrical Information from the Scene as 2d G-buffers. Retrieved as floating textures from the graphics hardware ensuring their fast computation as an off-screen rendering process, the G-buffers store all the scene's relevant geometrical information. Normals and the principal directions of maximal curvature are stored as complete 3d eye-space directions. Conversely, all other directions are only expressed as the tangential angles they form with the directions of maximal curvature. Similarly, we keep the eye-space z-coordinates in order to allow the recovery of a given buffer pixel's 3d position if necessary. Most of the curvature-related measurements presented here are computed for each fragment of the 3d scene in order to achieve the best accuracy possible.

pal directions of maximal curvature, computed using Rusinkiewicz's two-pass method [Rus04], indeed define along with the vertices' normals a local frame at each vertex which can be described as a quaternion. Prior to their processing by the graphics hardware, the principal maximal directions of a same face are "harmonized" with one another so as to avoid obtuse angles between adjacent vertices in the tangent plane.

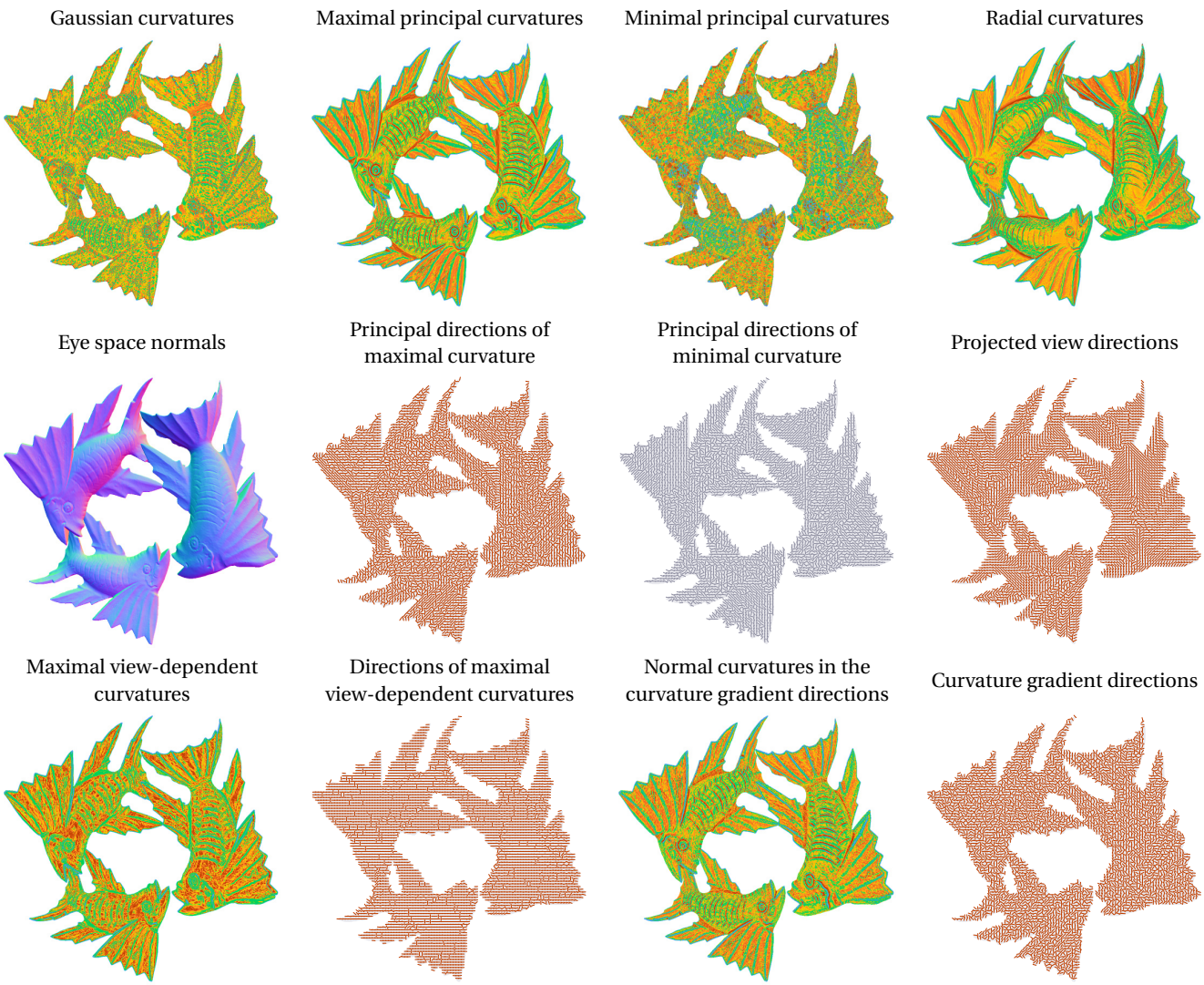
Here is recalled the quaternion interpolation scheme we use for the computation of the principal curvature directions across the facets of the mesh. Assuming (α, β, γ) designate the barycentric coordinates of the processed fragment across the face and $(\mathbf{q}_A, \mathbf{q}_B, \mathbf{q}_C)$ the triple of quaternions lying at its vertices, we compute intermediary quaternions as:

$$\begin{aligned}\mathbf{q}(\alpha,\beta,\gamma;\mathbf{q}_A,\mathbf{q}_B,\mathbf{q}_C) = {} & \alpha\text{slerp}\left(\gamma;\text{slerp}\left(\frac{\beta}{\alpha+\beta};\mathbf{q}_A,\mathbf{q}_B\right),\mathbf{q}_C\right) + \beta\text{slerp}\left(\alpha;\text{slerp}\left(\frac{\gamma}{\beta+\gamma};\mathbf{q}_B,\mathbf{q}_C\right),\mathbf{q}_A\right) \\ & + \gamma\text{slerp}\left(\beta;\text{slerp}\left(\frac{\alpha}{\gamma+\alpha};\mathbf{q}_C,\mathbf{q}_A\right),\mathbf{q}_B\right).\end{aligned}$$

With each rendered fragment's interpolated principal curvatures and principal directions at hand, we can also recompute the radial and view-dependent curvatures following the techniques of [DFRS03] and [JDA07] instead of directly taking their automatic, bilinear interpolated counterparts. The estimation of the curvature gradient direction as well as the normal curvature along it [KST08], are similarly evaluated on a per-fragment basis which allows us to enjoy as accurate as possible results.

Approximating Stroke-Dependent Derivatives

Although most of the stroke features detailed in Section 8.2.1.1 are directly stored in the aforementioned G-buffers and can therefore be directly accessed at any screen position, some of them are still left missing due to their direct dependence to the stroke path's shape itself. Examples include the normal curvature in the



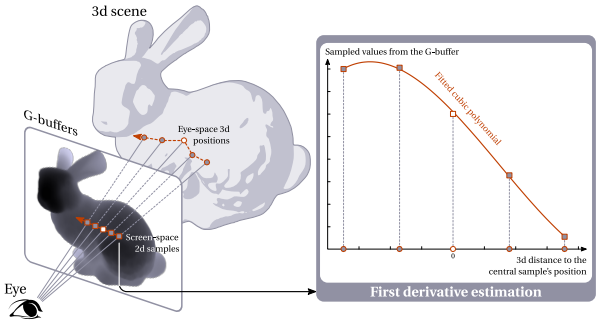
8.15 Extracting Geometrical Information from the Scene as 2d G-buffers (Continued).

stroke's tangential direction – which can be estimated using the principal curvatures and their related directions according to the Euler's formula –, but most importantly all the features involving first derivatives whose direction is subordinate to the stroke's. The latter measurements encompass all curvature derivatives, as well as the perspective depth discontinuities' strengths across the stroke's path.

A first hurdle towards their computation lies in the very nature of the handled principal curvature directions, obtained as the eigenvectors of each vertex' averaged second fundamental form matrix [Rus04]. Indeed, while their *direction* is always relevant, their representative *vector* cannot be directly incorporated into our computations as 180 degree flips between adjacent vertices often occur. This phenomenon hinders the direct interpolation and straightforward access of derivatives in these directions, as their sign therefore become incomparable from one sample to another – only their magnitude remaining spatially consistent.

In order to cope with this issue, we evaluate all derivative-based feature values in the imaging screen space and take the sampled stroke's tangents as local references for consistently orienting the differentiation directions along the path of the stroke. Our approach, illustrated in Figure 8.16, draws inspiration from the finite-differences approximation of derivatives and considers the sequences of values from both sides of the stroke, taken at constant screen-space steps along the direction of derivation. These sets of values are then robustly approximated by fitting cubic polynomials for which the first- and second-derivatives can be analytically computed. In order to best account for the 3d information stored in the G-buffers and get as close as possible to the real world space derivatives, we parameterize our polynomials by the Euclidean distances between the samples' back-projected eye-space 3d positions as an attempt to instead account for the geodesic distances across

8.16 Computation of Screen-Space Derivatives. In order to alleviate the limitations in terms of precision of the derivatives we can estimate from our G-buffers, we trade finite difference approximations for the fitting of cubic polynomials to the sequences of the values sampled along the directions of derivation. The resort to screen space evaluated derivatives is motivated by the need of making all the derivatives belonging to a same stroke consistent. This is done by parameterizing the successive directions of derivation with respect to the underlying stroke's local tangents.



the 3d surfaces.

Computing the Stroke Features' Final Footprint

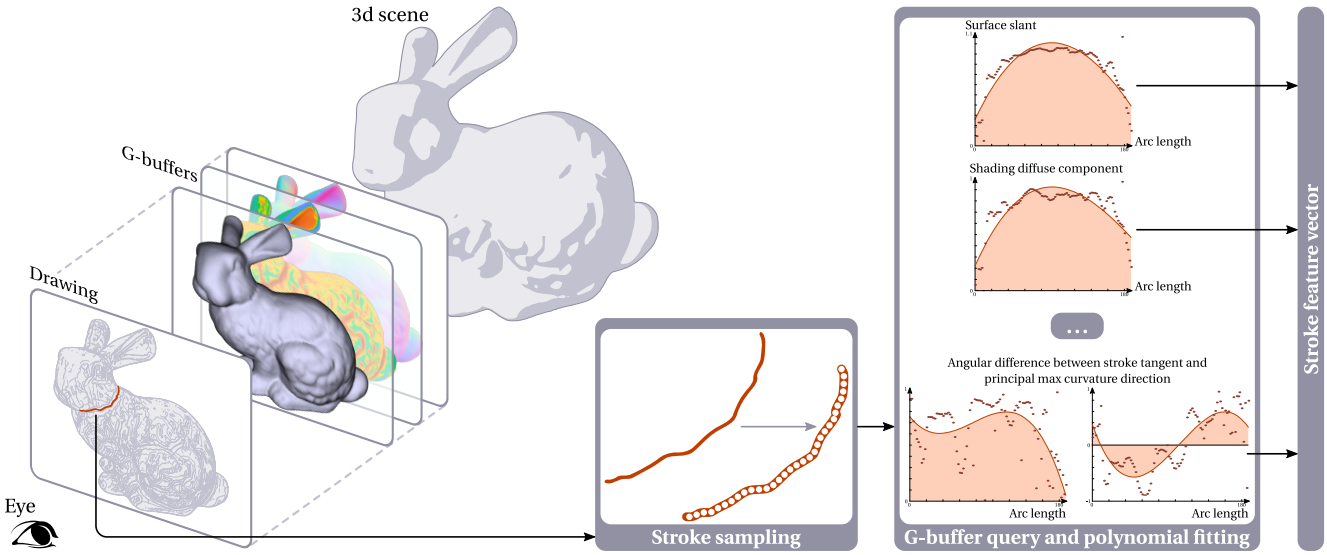
Considering back our initial problem of stroke-driven sampling of our sets of G-buffers, we now have to handle the ordered sequences of the sampled values for each geometric feature mentioned in Section 8.2.1.1, and mold our stroke's signature vectors out of them. In order to exploit the information conveyed by the sequence order itself, limit aliasing-related artifacts – the G-buffers being stored as 1024×1024 arrays of floats – and achieve an improved robustness to outliers, we find these sequences' best approximating cubic polynomials in the least-square sense. We then consider their integral normalized by the length of their associated stroke. The resulting value then becomes the sequence's representative value and as such, one of the components of the stroke's final signature. Once the different sequences corresponding to each of our geometric features are processed, each stroke is finally simply represented by a N_{feat}-dimensional feature vector, $N_{\text{feat}} = 19$ corresponding to the total number of the aforementioned geometric features. All steps leading to the estimation of the strokes' feature vectors are recapitulated in Figure 8.17.

Preparing the G-buffers

Before proceeding to the stroke sampling, G-buffers storing theoretically unbounded geometric quantities are clamped to their associated distribution's 2- and 98-quantiles in order to limit the detrimental influence of their extremal values. This is especially beneficial for measurements such as Judd's view-dependent curvature – which gets infinitely high at occluding contours' locations – or quantities computed using Belkin's Laplacian discrete operator. The effect of this clamping step is easily noticeable in the probability density functions graphed in Figure 8.18, each of them being terminated by two peaks of probability roughly equal to 0.02.

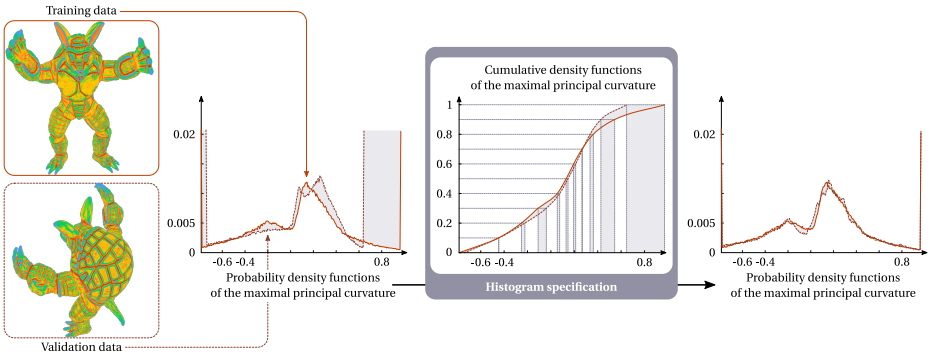
Now follow secondary details concerning the computations invoked by our preprocessing of the G-buffers in order to allow the generalization of the style learned by our classifier. Indeed, while in the context of the extension of a primary drawing by the user to the whole scene a single set of G-buffers is consulted for the description of the strokes, one could also be interested in exploiting the inferred rendering rules over new 3d settings obtained either by changing the viewpoint, lighting set-up or in applying them to entirely different geometric models. Hence are needed some additional treatments of the G-buffers in order to ensure the sound comparison of the geometric quantities arising from such different scenes.

Once trained, the strong classifier can easily be run over strokes corresponding to features from different vantage points or even distinct models. In such "style transfer" cases however, additional processes must nevertheless be performed in order to ensure the relevance of the comparisons between samples drawn from the training and the validation data sets. For that aim, we undertake histogram specification between the training and validation distributions of all the geometric features considered for training our classifier. Their probability density functions are approximated by fitting a monotonic cubic polynomial to the sequence of their 10 percentiles. This way, the distributions of the features composing the validation data set can be



8.17 Computing the Strokes' Representative Feature Vectors. The above figure summarizes the different steps necessary for the evaluation of a given stroke's descriptive features. Our subsequent classification stage will resort to these values in order to decide whether or not the stroke is in agreement with the user's ongoing drawing. Let us consider the stroke displayed in red within the drawing at the bottom left corner. First, we find sampling positions along its path by walking at constant arc length steps. Geometry-related values stored by the G-buffers at these locations are then gathered and processed in order to compute the classification query values. Once sorted by their associated arc length parameter *(dark red dots)*, we fit a cubic polynomial to them *(red curves)*. Finally, the stroke's feature vector components are computed as these approximative functions' integral normalized by the stroke's length. Note that in the case of angular queries, we compute the normalized integral of both the angles' cosines and sinuses, and find the resulting representative angle thanks to their arc tangent at the very last moment. This approach is directly inspired by what is classically done in directional statistics.

brought to the domains of the distributions on which the sequence of weak classifiers has been adjusted.



8.18. Data Processing for "Style Transfer". If we apply our learned drawing style to scenes other than the one used for training, we need to process the new geometric data to make them comparable with the training scene's. Some measurements are indeed not naturally bounded such as curvatures or quantities involving the Laplacian approximation of the surface's Gauss map. We thus need to "embed" the new geometric information *(validation data)* to the domain for which the classifier parameters remain valid *(training data)*. For that aim, we perform histogram specification for each geometric feature taken independently. Their cumulative distribution functions are robustly computed by fitting cubic polynomials to their 10 observed percentiles.

8.2.2 Learning the User's Style with AdaBoost

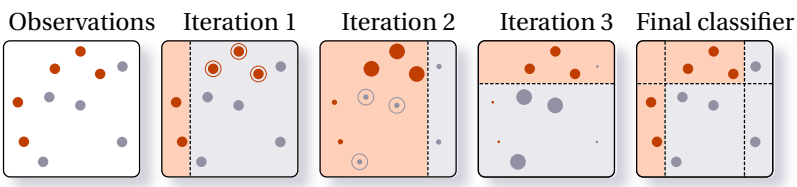
Similarly to Lum *et al.*'s intuition of recasting the issue of assisted line selection as a classification problem, we extend it to the case of example-driven drawing creation. With our stroke descriptors at hand, we can now consider their geometry-based feature vectors in order to find the best computational ways to discriminate between relevant and irrelevant strokes with respect to the users' roughly sketches and this way draw closer to the capture of their drawing styles. In order to reach that objective and overcome the obvious difficulties of capturing the possibly non-linear correlations between the stroke's relevance and associated geometrical features, we resort to the AdaBoost learning technique introduced back in 1995 by Yoav Freund and Robert

Schapire [FS95]. Their classification performance enhancing has known a tremendous success since then due to the alliance of its algorithmic simplicity with its empirical prowess, and has been incorporated to many experiments beside ours. Its theoretical properties, as well as its possible improvements have also been intensively studied.

Standing for *adaptive boosting*, ADABOOST 's main strength comes from the simplicity and efficiency of its underlying principle. Starting from the observation that while directly finding accurate classification rules is very difficult in most cases, devising general rules of thumb, may they be not that accurate, is conversely much easier, Freund and Schapire conceived a meta-algorithm which greatly increases the performances of any existing learning algorithms. By cleverly combining them and constantly adapting the observation set on which they are trained, ADABOOST can mold a *strong classifier* – strongly correlated with the distributions of the observations' actual category – from any set of *weak classifiers* created by even the simplest of learning procedures. The key insight is not only to combine the rules of thumb obtained by applying the weak learning procedures to different subsets of examples, but also to attach a weight to each observation. By controlling each training example's importance and therefore influence over the learning procedures themselves, ADABOOST can have them focus on a specific set of examples and enjoy an infinity of weak classifiers from a finite set of learning procedures.

Working as an iterative process, ADABOOST intuitively updates the weight of a training observation according to the correct recognition of its label by the strong classifier being constructed. If left misclassified at the end of a boosting iteration, any observation must have its weight increased in ways such that it would subsequently play a greater role on the learning procedures, and hopefully be better recognized once the weak classifier they would produce during the following iteration would be incorporated to the strong classifier. ADABOOST therefore builds a sequence of weak classifiers trained over the set of training observations from the perspective of their weight distribution that evolves as it can more or less accurately predicts their actual categories. The weight updating and weak classifier training processes are repeated until a predefined number of steps has been fulfilled – which directly affects the resulting classifier's complexity – or the classification error went below a given threshold. The strong classifier's decisions are finally taken according to the sign of the weighted average of the votes of the rules of thumb gathered at each step. Figure 8.19 provides a graphical explanation of whole procedure.

Empirical evidences, as well as more formal proofs attest the remarkable performances of the ADABOOST algorithm in terms of its achievements in example-driven learning. Numerous and various illustrations of its application to disparate learning tasks definitely show the rapid and steady decrease of the classification error it generates alongside the course of its iterations – the ADABOOST 's training error has notably been proven to be bounded –, often overshadowing other famous techniques such as support-vector machines or neural networks. Additionally, the fact it naturally creates linear strong classifiers of increasing complexity which remain outstandingly robust to over-fitting, makes it a perfect candidate for our objectives as our classifier will have to be general enough to be applied to unexplored parts, or even brand new feature spaces.



8.19. ADABOOST Toy Example. Redrawn from Robert Schapire's lectures. Shown above is a toy example giving the intuition behind the ADABOOST learning procedure. The studied observations are represented by dots whose color reflects the label, and size accounts for the associated weights. Their only distinguishing features are their position coordinates. The two learning procedures chosen here thus consist in finding of the best segregating x-, or y-plane. The boosting procedure comprises three successive steps. At each iteration, the observations misclassified by the rule of thumb found in the previous iteration are being granted an increased importance so as to be better classified on the next turn. The final strong classifier is finally obtained by taking the weighted average of the votes of all the different rules of thumb selected during the execution of the ADABOOST procedure.

From the perspective of our line drawing anticipation purposes, we briefly present the layout of the ADABOOST algorithm for the sake of completeness in the following paragraphs. In addiction, a graphical interpretation of it, along with all the employed notations is also exposed in Figure 8.21 and may prove to be of a

complementary aid for a novice reader. First, let us state the context of our problem again, and make more explicit the role we intent to have AdaBoost play for its solving. In a nutshell, we want our method to non-perimetrically capture the users' drawing styles and to do so by resorting to a binary classification framework. In our approach, this "captured style" assumes the form of a *classifier* which once fitted to the available users' inputs, can decide given a new stroke, whether or not it should be added to their drawings, for it appears consistent with them. For that aim, we assume stroke placement by the users is mostly driven by geometry-related concerns, and therefore we proposed in Section 8.2.1 a specific selection of geometric measurements as the stroke features – denoted $\mathbf{x} \in \mathbb{R}^{N_{\text{feat}}}$ – we will consider for taking decisions. We therefore look for a decision-making mapping

$$\begin{array}{rccc} H: & \mathbb{R}^{N_{\text{feat}}} & \longrightarrow & \{-1,1\} \\ & \mathbf{x} & \longmapsto & y = H(\mathbf{x}) \end{array},$$

where y designates the predicted category – 1 if deemed relevant, -1 otherwise – of the stroke of geometric signature $\mathbf{x}$. As mentioned earlier, AdaBoost computes H as the combination of weak classifiers noted

$$\begin{array}{rccc} h: & \mathbb{R}^{N_{\text{feat}}} & \longrightarrow & \{-1,1\} \\ & \mathbf{x} & \longmapsto & y = h(\mathbf{x}) \end{array},$$

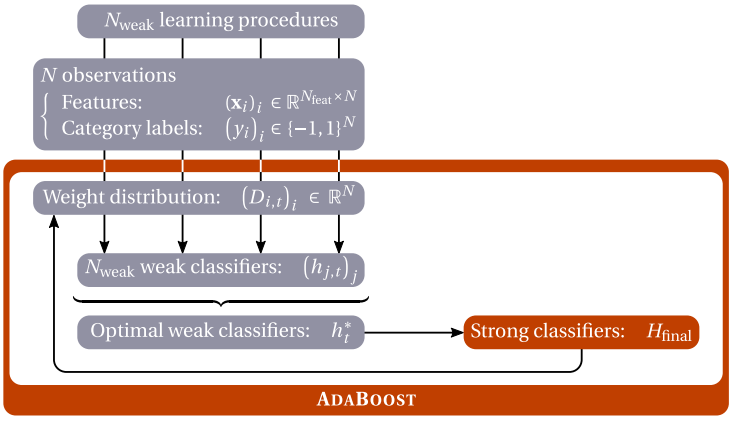
created by a set of learning procedures. In order to obtain these rules of thumb and train its strong classifier, AdaBoost naturally requires a training set of already-labeled observations $(\mathbf{x}_i, y_i)_i$. In the context of our application, positive observations naturally includes the users' manually traced strokes as well as the validated predictions of our method. Conversely, negative observations correspond to already gathered unsatisfactory predictions, flagged as so by the users, or even a random selection of automatically computable lines for starting up the classification procedure. A unit-normalized weight distribution $(D_{i,t})_i$ is associated with the complete example set, and is updated at each round t according to the trained classifier's current prediction abilities. The detailed pseudo-code of the whole procedure is given in Figure 8.20.

given training set $(\mathbf{x}_i, y_i)_{i=1\ldots N}$ with for all i, and $y_i \in \{1,-1\}$
initialize weight distribution $(D_{i,1})_i$
for all boosting round t
  get rules of thumb $(h_{j,t})_{j=1\ldots N_{\text{weak}}}$ from weak learning procedures trained on $(D_{i,t})_i$
  select h_t^* such that $\arg\min_{h_{j,t}} \{\varepsilon_{j,t}\} = \arg\min_{h_{j,t}} \left\{\sum_{i=1}^N D_{i,t} \mathbb{1}_{[h_{j,t}(\mathbf{x}_i) \neq y_i]}\right\}$
  update weight distribution with for all i, $D_{i,t+1} = \frac{1}{Z_t} D_{i,t} \begin{cases} \exp -\alpha_t & \text{if } h_t^*(\mathbf{x}_i) = y_i \\ \exp \alpha_t & \text{otherwise} \end{cases}$
    where $\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t^*}{\varepsilon_t^*}$ and Z_t is a normalizing constant
output $H_{\text{final}}(\mathbf{x}) = \operatorname{sign} \sum_t \alpha_t h_t^*(\mathbf{x})$.

8.20 Pseudo-Code of the AdaBoost algorithm [FS95].

Now we need to expose the set of weak learning procedures we chose for our technique. As their sole requirements are their consistency across distributions, and the fact they have to perform only slightly better than random guesses, we decide to resort to fairly straightforward tests. All these tests consider one of the $N^{\text{feat}} = 19$ features of the stroke signatures and therefore only deals with the partition of one-dimensional domains. They are of three kinds – all illustrated in Figure 8.22 – and study the weight distributions of the positively-labeled and negatively-labeled training example distributions separately:

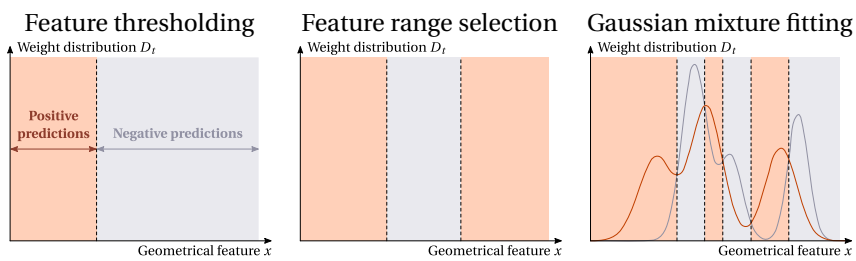
- **Thresholding:** Obviously the simplest partition test possible, we seek, given the current observation weight distribution $(D_{i,t})_i$, for the best possible segregating value that would minimize the weighted classification error $\varepsilon_{j,t}$.
- **Range:** It only consists in the extension of the former test to the two-threshold case and constitute a generalization of it.



8.21 Work Flow of the AdaBoost Algorithm. In order to construct the strong classifiers H_{final} from a set of inaccurate rules of thumb, AdaBoost has the weight distribution associated with the training observations $(D_{i,t})_i$ updated at each round t. The weak classifier training procedures therefore produce a theoretically unbounded sequence of imperfect classification rules $(h_{j,t})_t$, adaptively favor their training towards the most often misclassified examples, and finally combine their votes according to their individual performances.

- **Mixture-of-gaussian modeling**: Probably the most computationally intensive trial of our tests, the Gaussian-Mixture Modeling (GMM) classification test works by approximating the weight distributions of the positive and negative observations separately. When confronted to a new feature value, the ensuing weak classifier makes its final decision by comparing the feature value's occurrence likelihoods returned to the two fitted models and assigning it the category corresponding to the maximal probability. This test therefore leads to more complex domain partitions than the ones generated by the aforementioned kinds of classifiers. We create classifiers for mixtures made up of 1, 2, and 3 gaussian functions and we let the AdaBoost procedure to naturally deal with this obvious redundancy.

At each iteration, these three straightforward learning procedures are run over the components of all the stroke feature vectors constituting the classifier's training set. The only exceptions are the angular difference features which are only treated by an altered version of range detection strategy that accounts for the circular nature of the feature domain in these specific cases. At any rate, provided the fact our stroke signatures hold $N_{\text{feat}} = 19$ features – two of which being of angular nature –, the total number of rules of thumb we create at each step is equal to $N_{\text{weak}} = 5(N_{\text{feat}} - 2) + 2 = 87$ in all the examples presented herein.



8.22. Rules of Thumb Used by our Method. The background's color represents the domains of the feature value x for which the weak classifier returns either a positive *(red)*, or a negative response *(grey)*. It should be noted that its training studies the distributions of the weights currently attached to each observation $(D_{i,t})_i$ and not their sole labels $(y_i)_i$, for the former evolves alongside the execution of the algorithm and confers it its adaptive character.

The addition of the ever-evolving weight distribution $(D_{i,t})_i$ obviously plays a capital role in the success of whole learning process. And so does its initialization that should by no way be taken lightly or for granted. In their general presentation of the AdaBoost algorithm, Freund and Schapire advise to initially attach a same, equal weight to all training examples – which therefore equals $D_{i,1} = \frac{1}{N}$, $\forall i \in \{1 \dots N\}$. This intuitive solution may not provide us with the most satisfactory classification results possible for several reasons. The first one is directly related to our interactive application scenario that assumes the most relevant part of the positive training set stems from the users' inputs and feedbacks. Therefore, the set of positively-labeled observations is to be built incrementally by the users for the intervention of our method to be as seamless as possible, and is likely to be small at the beginning of a new drawing. The size of the positive and negative subsets is hence to often strongly differ, yet must have an equal importance for the training of our classifier. The second reason for a different initialization scheme comes from the real nature of the observations we handle, namely strokes. More than geometric signatures, they are two-dimensional paths whose combination is meant to compose a drawing. Therefore, we want long curves to be scrutinized with more attention than more discrete, shorter lines during training. Bearing these two objectives in mind, we propose the following initializing weights in-

stead:

- for the observations whose labels are positive: $D_{i,1} = 0.5\frac{l_i}{\sum_{i:\ y_i=1} l_i}$,
- for the observations whose labels are negative: $D_{i,1} = 0.5\frac{l_i}{\sum_{i:\ y_i=-1} l_i}$,

where l_i refers to the total arc length of the stroke corresponding to the ith observation $(\mathbf{x}_i, y_i)$. It happens our different experiments showed great improvement after the integration of this alternate initializing step.

As a more realistic example than the one presented in Figure 8.19, Figure 8.23 exposes all the steps of the boosting procedure behind the construction of a strong valley line-detecting classifier as the careful combination of stroke features monitoring weak classifiers. At every iteration of the algorithm, the so-called stroke features are processed independently. The two distributions made of the features corresponding to the selected and rejected stroke sets are jointly examined by the different weak classifiers so that they could tune their decision parameters and come up with the best possible predictions. Finally, the weak classifier achieving the minimal classification error with respect to the user's manual selection is retained. Observations left misclassified after the addition of the last optimal weak classifier to the strong classifier are weighted so they are being granted a stronger importance in the subsequent steps. Indeed, the weights attached to the observations directly alter the feature distributions. They henceforth also influence the training of the next weak classifiers picked in order to enhance the quality of the strong classifier's decisions. This process is repeated until a fixed number of iterations has been reached – 20 in all our examples.

8.2.3 Classification Results and Validation of our Approach

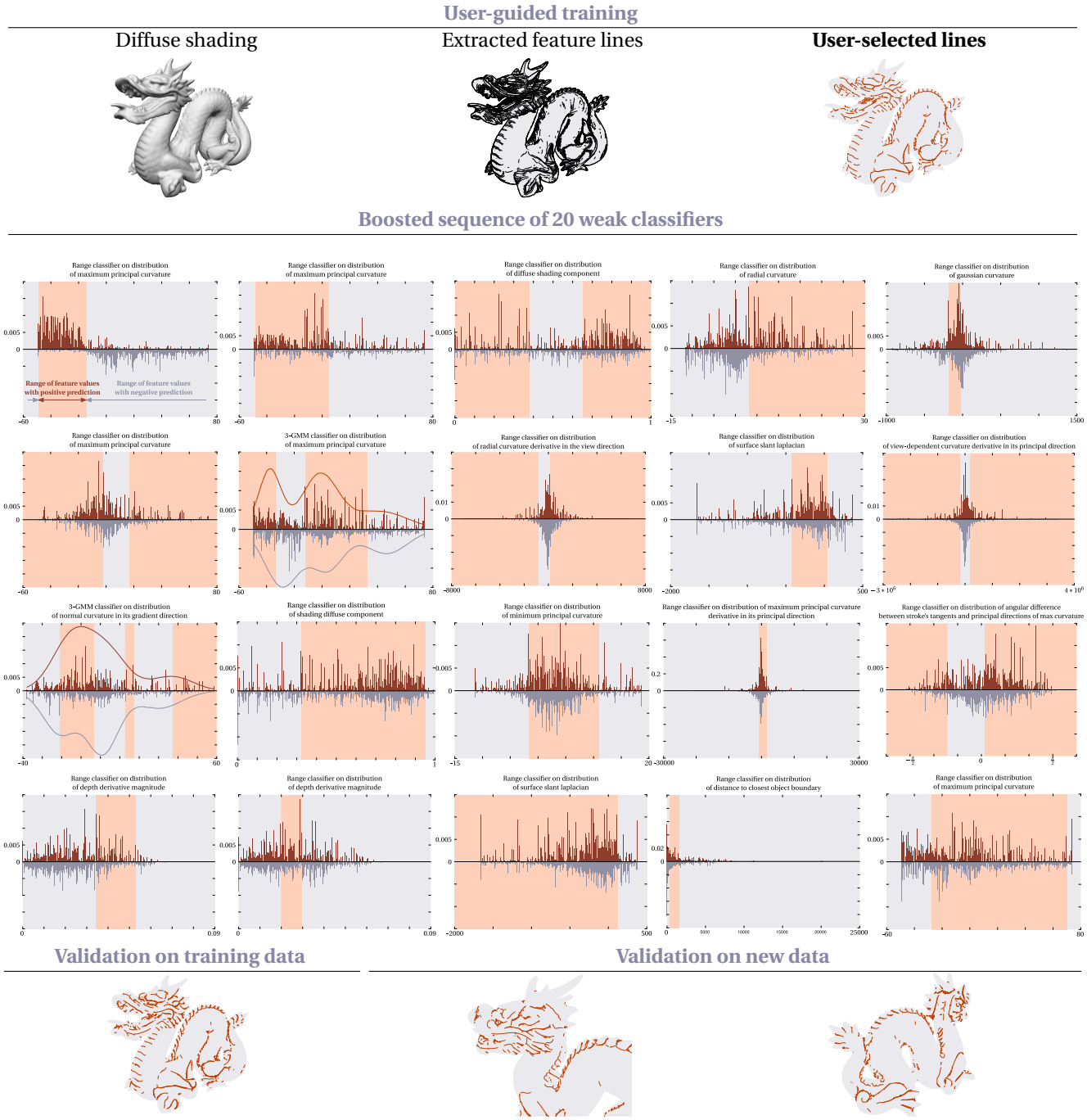
Follow automatic line selections performed by our boosted strong classifier that illustrate its suitability to isolate a specific categories of feature lines among a wide selection of curves.

Silhouettes and occluding contours [HZ00], view-independent crease lines – encompassing ridges and valleys –, suggestive contours and highlights [DFRS03, DR07], apparent ridges [JDA07], demarcating curves [KST08] and principal highlights [DR07] take part in the tests and for a given vantage point, form the total set of strokes used for training. Among these, a specific flavor of lines is explicitly flagged as relevant by the user and constitutes the subset of positive responses to which the strong classifier is fitted.

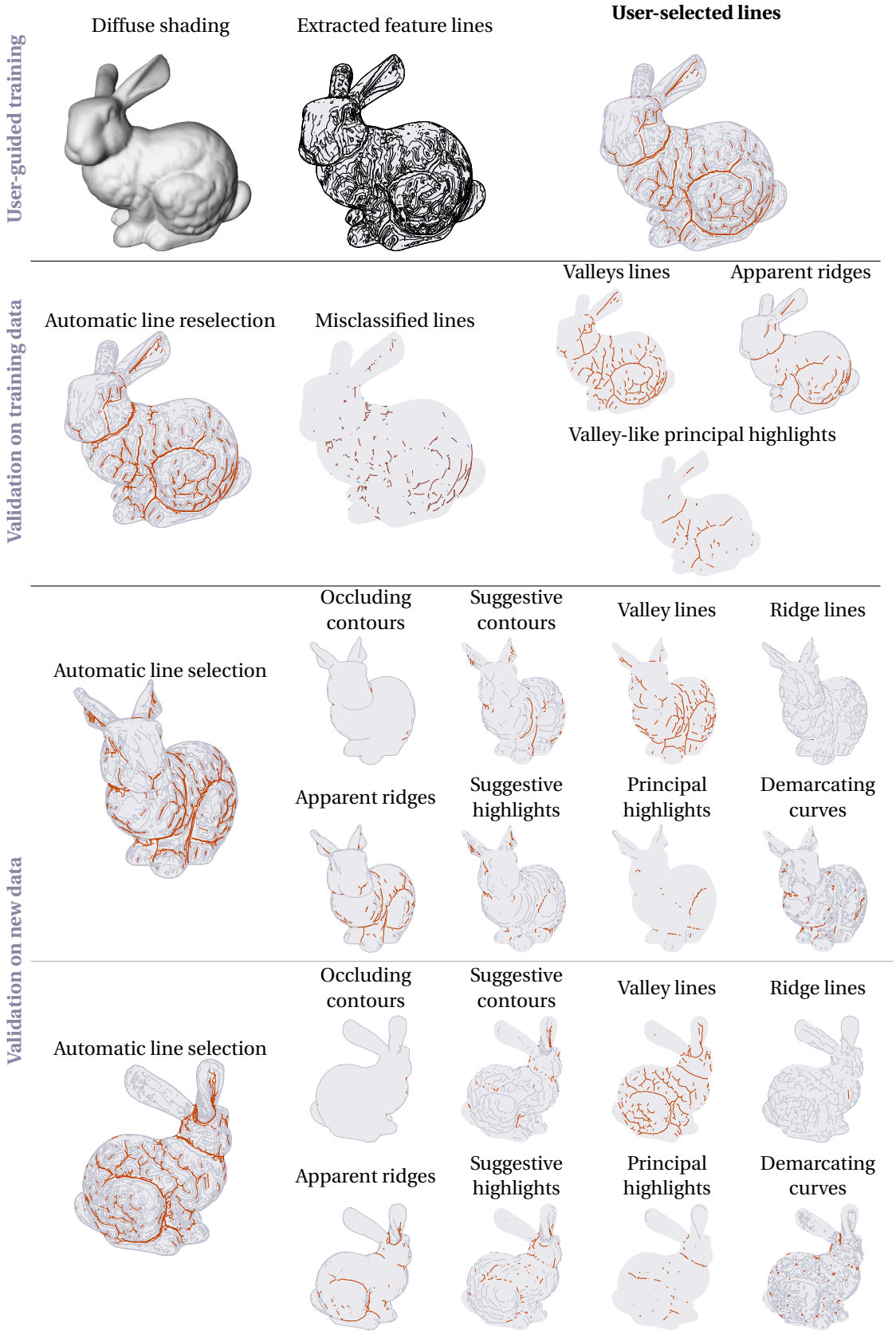
As a first step for assessing its success in "guessing" the user's intent, we run the resulting classifier over the whole set of strokes used for its training and observe whether or not it can properly reproduce the user's initial selection. A second, more advanced and critical test consists in changing the viewpoint, computing entirely different feature line sets using the classical object space line extraction techniques, and considering the predictions of the classifier on these brand new sets of curves.

The following results are introduced in order to increasing misclassification, starting with highly faithful reselections by the classifier of lines taken from the training set itself, to less accurate predictions. Among the classification trials we conducted, we present here tests aiming at the implicit identification by the classifier of a specific type of automatically extractable lines, such as valley lines (*cf.* Figures 8.24 and 8.28), ridge lines (*cf.* Figure 8.25), demarcating curves (*cf.* Figure 8.26) as well as occluding contours extended by their suggestive counterparts (*cf.* Figure 8.27). In all classification illustrations, selected lines are displayed in **red** while rejected lines are drawn in thinner **grey** curves. In the specific case of misclassified lines, color conventions slightly differ, with false positives – curves that should not have been selected – appearing in an alternate **darker red** and false negatives – lines that were wrongly rejected – in **blue**. Misclassification rates established by the direct comparison between the user indications with the line selection performed by the trained classifier on the training data set sure yield valuable information on the quality of its decisions. But we also stress the fact that its ability to isolate – or at least confine its selection to – the subset of the lines guiding the test is an as, if not more, important suitability indicator for our end purposes.

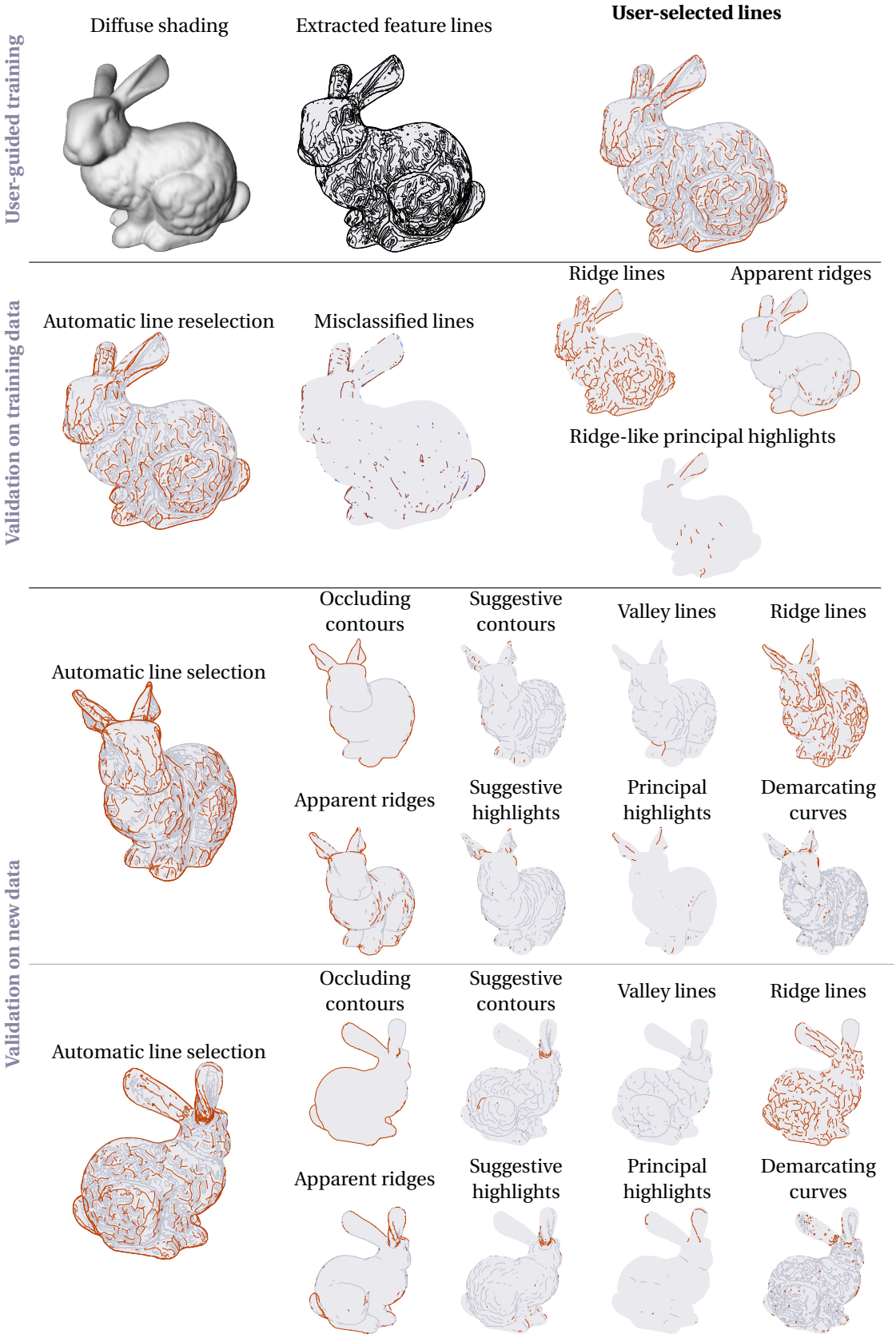
In addition to the line drawings made up of the strokes the different strong classifiers retain, we expose supplemental measurements in order to more objectively assess their quality and more importantly, evaluate



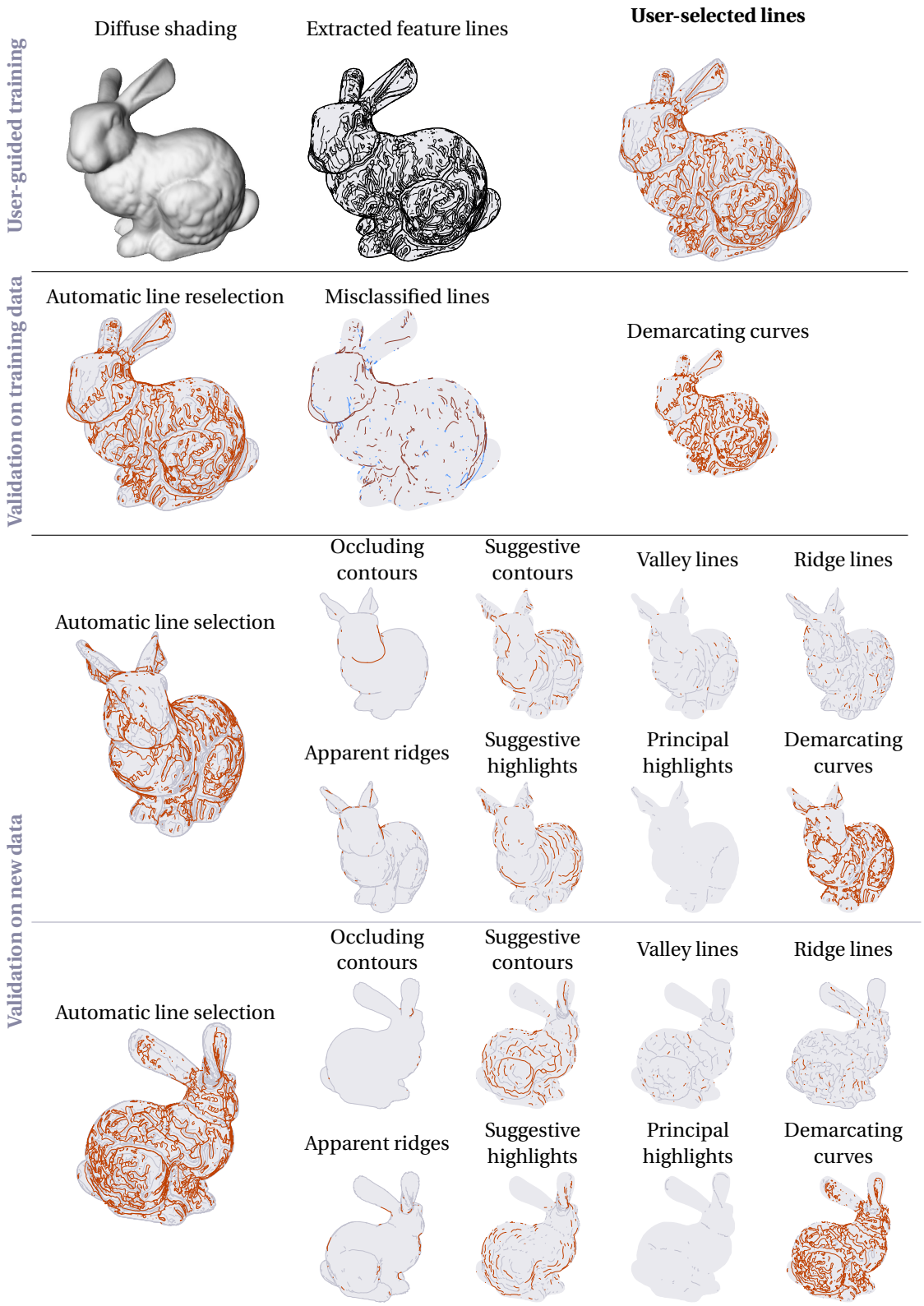
8.23 Detailing the Execution of the ADABOOST Algorithm. One the main appeal of the ADABOOST learning procedure is its ability to create strong classifiers from the optimally-weighted sequence of a set of weak classifiers whose only requirement is to return slightly better predictions than plain random guesses. Above is the complete weak classifier collection created along the execution of the algorithm for the detection of valley lines on the *Dragon* model. Each of these classifiers only considers a single component from the strokes' feature vectors, and adjusts its decision parameters in order to best capture the distributions corresponding to the selected and rejected stroke sets *(red and – negative – grey histogram respectively)*. At each boosting step, the two distributions are updated in ways they would give more importance to observations that are misclassified by the current strong classifier. This key observation explains the possible presence of multiple occurrences of a same "kind of" weak classifier within the sequence, as well as the ever-changing aspect of the distributions in question. Note that the decisions taken by distinct occurrences of a "same" weak classifier are to differ as their parameters have been fitted to differently-weighted distributions.



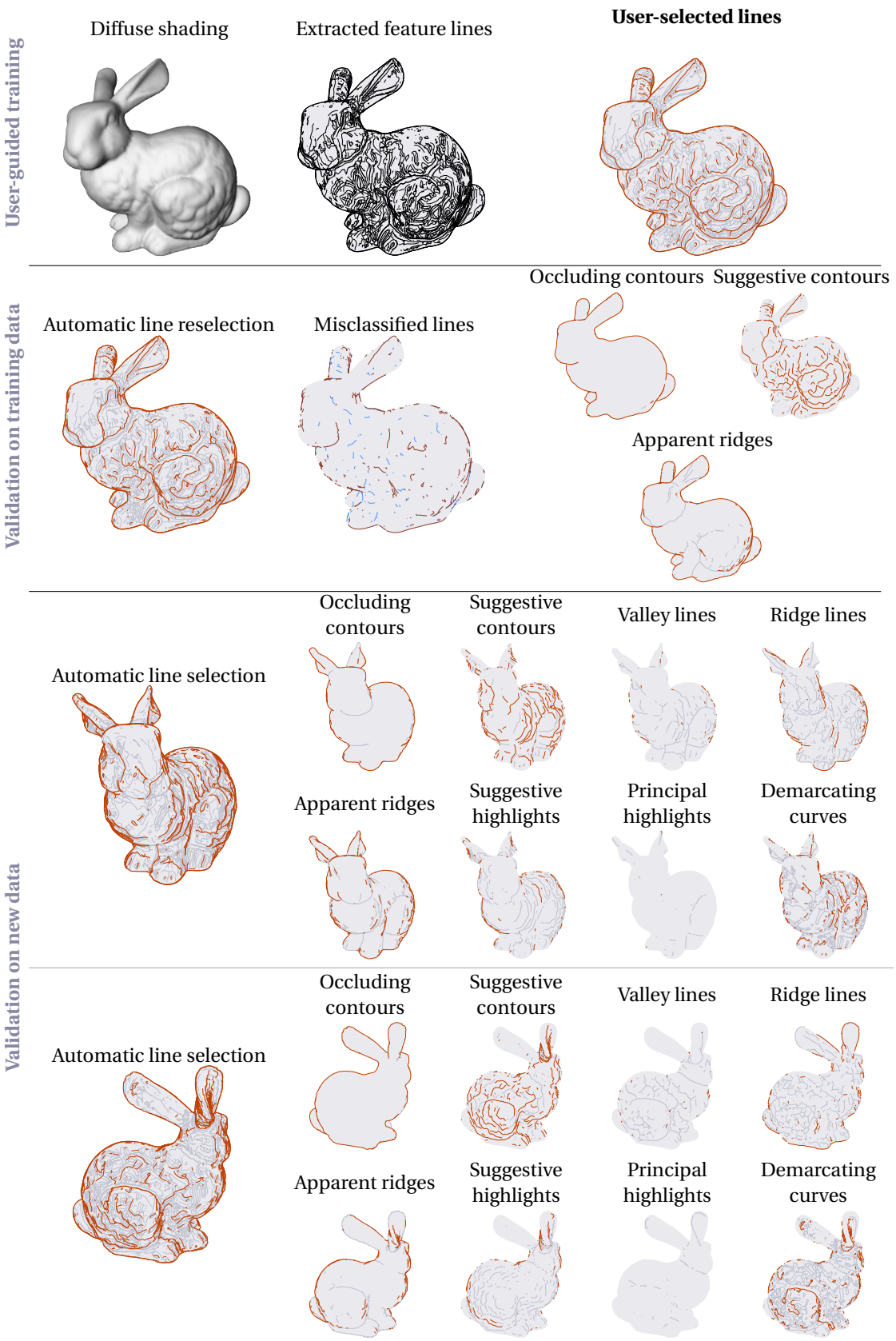
8.24 Valley Line Detection by our Boosted Classifier. The above selection of drawings shows an example of classification training tests we performed in order to assess the quality of the predictions yielded by the ADABOOST algorithm. Valleys lines, along with the compatible subsets of apparent ridges and principal highlights were manually selected while leaving all the remaining lines flagged as irrelevant *(top row)*. We observe that once trained, the classifier considering the optimal combination of the strokes' geometric features works quite well in this case, either when applied to the training stroke set *(middle row)* or to drawings established under various 3d scene settings *(bottom rows)*.



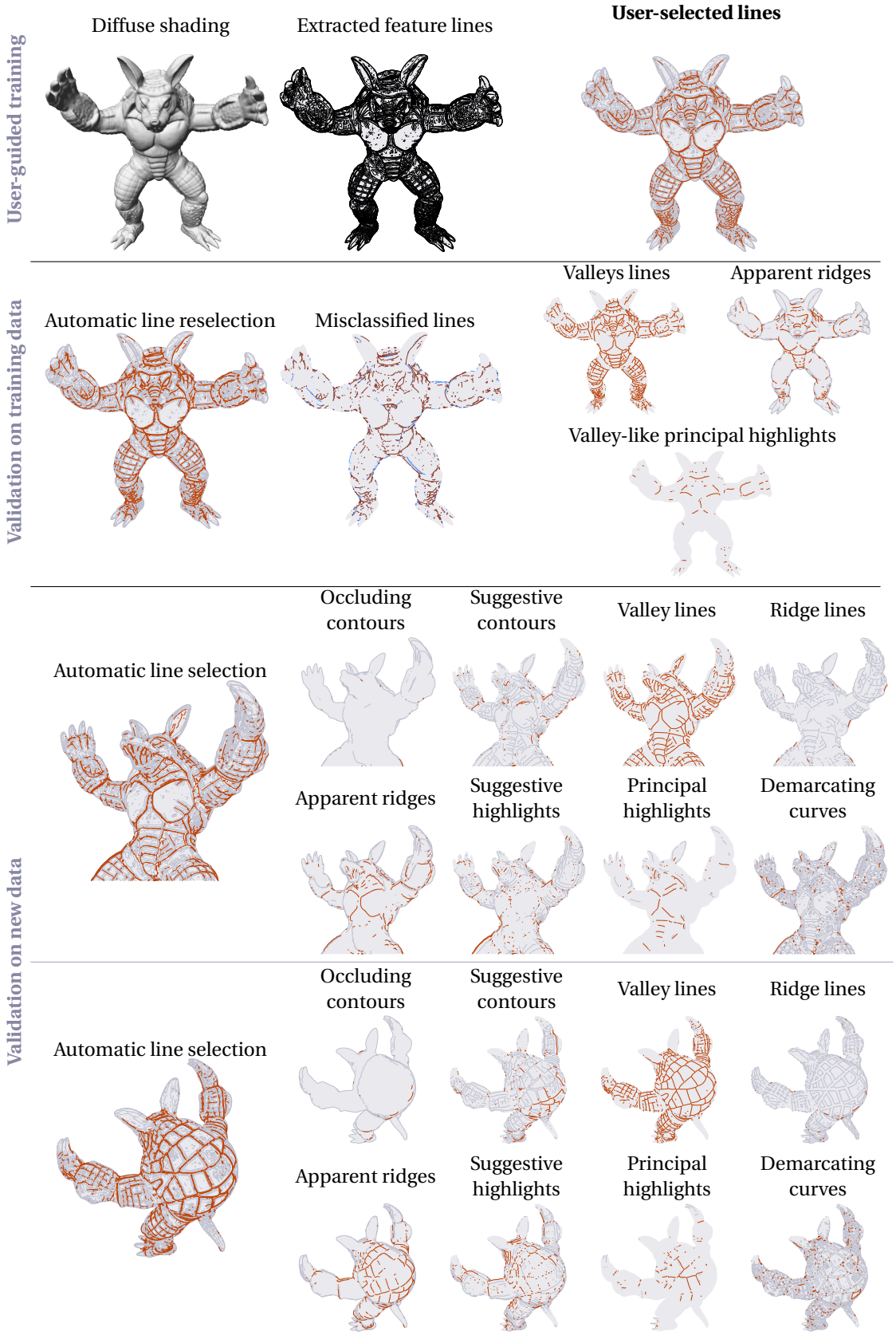
8.25 Ridge Line Detection by our Boosted Classifier. Above precedes another classification trial similar in spirits to the one presented in Figure 8.24, the major and sole difference being its focus on ridge-like feature lines in place of valley lines. Results are again more than satisfactory, with theoretical ridge lines constituting in all cases, the preeminent part of the strokes retained by the classifier.



8.26 Demarcating Curve Detection by our Boosted Classifier. Here is presented yet another classification test for our classifier, dedicated here to the selection of the demarcating curves. This example is more difficult than the ones illustrated in Figures 8.24 and 8.25 because of the more expanded and dispersed repartition of these lines throughout the drawings. The ensuing intensive amount of overlap between retained and irrelevant strokes naturally induces a more pronounced misclassification, nevertheless the classifier's output for new vantage points remain remarkably good as the lines it flags as relevant mostly belongs to the new line subsets corresponding to the demarcating curves.



8.27 Occluding and Suggestive Contour Detection by our Boosted Classifier. Same as in Figure 8.26 except here the classifier is trained in ways such that it would detect occluding and suggestive contours. Due to their dispersed placement across the drawing, suggestive contours cause the classifier to accept too many lines and lead to an higher misclassification rate *(middle row)*. Nevertheless, for new viewpoints, occluding and suggestive contours are still specifically chosen among all other categories of lines *(bottom rows)*.



8.28 Valley Line Detection by our Boosted Classifier. This example shows the suitability of our approach for more complex and intricate geometric models. Although the direct reselection of the training set's lines may appear slightly too permissive albeit relevant *(middle row)*, the predictions of our strong classifier for new vantage points are impressively accurate and show a clear tendency to favor theoretical valley lines *(bottom rows)*.

their suitability to the actual task of these tests: the automatic detection of a specific kind of feature lines. All these additional information are displayed in Figure 8.29.

In the specific case of validation procedures directly involving the training sets themselves, we have an exact notion of misclassification. Conversely, for lines corresponding to drawings established for new vantage points, this concept is not as precise as strokes from different line categories can exhibit strong overlap. This is especially true with apparent ridges that often appear highly redundant with view-independent crease lines and even occluding contours. Therefore, the notion of misclassification cannot possibly be made directly coincident with the belonging of a line to the line category targeted by the test. For lack of measurements that would account for our classifiers' bad decisions, we instead inspect the relative proportions of the different line categories of the strokes that end up being *selected* by the strong classifier. Ideally, we expect these to be comparable with the relative amounts constituting the user's manual selections on which the different boosting procedures have be run. We propose two distinct values for all the aforementioned measurements: one unweighed version that directly counts observations, and an alternate version for which all observations contribute according to their associated stroke's length. We are actually mostly interested in the later quantities as longer strokes are likely to have a more dramatic impact on the resulting drawings. Following the same rationale, we also specifically initiated the boosting procedure with weight distributions granting an increased emphasis to long curves.

We can draw several positive conclusions from the consideration of the classification results from Figure 8.29. Of uttermost importance, the unexpectedly strong recall values exhibited by our strong classifiers in all cases. More than precision that encodes the probability that a selected stroke was actually to be selected, recall is here of critical importance. Indeed, it instead quantifies the probability of identifying all relevant observations and therefore directly reflects the completeness of the classification. The accuracy values of our strong classifiers are consequently fairly high and their classification errors relatively low. It is interesting to remark that the length-weighted counterparts of all these measurements always indicate improved performances and show that, as expected, longer curves tend to be better classified than shorter, possibly spurious ones. This trend is similarly visible in the line category ratios of the classifiers' selections. In all cases, we observe that the aimed line categories take the lion's share, a phenomenon even more pronounced when the stroke lengths are taken into consideration. This last observation is a more than promising sign regarding the ability of our chosen learning procedure and set-up to provide accurate line selections and predictions.

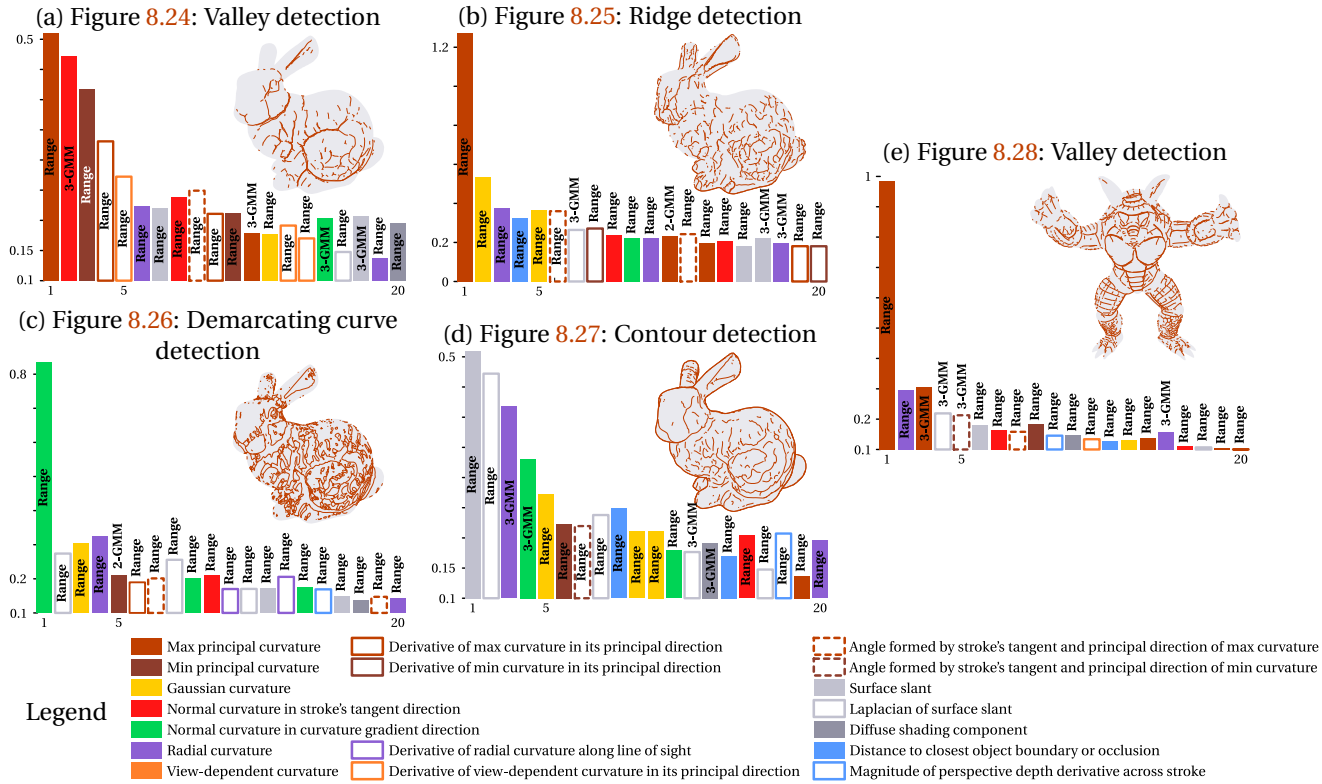
As a final inspection of the AdaBoost algorithm in order to validate our set-up, we lastly consider the sequence of the weak classifiers whose weighted combinations yield the strong classifiers studied so far (*cf.* Figure 8.30). Through their analysis, we aim at unveiling whether or not, not only the learning procedure but our whole stroke description framework actually manages to identify the user-selected strokes' true nature. Observations of these sequences hint that it is actually the case as the weak classifiers of preeminent weights are definitely related with the theoretical definitions of the curves to be recognized. Crease lines, may they correspond to ridges or valleys, are mostly detected thanks to the distinguishing distribution of their maximum principal curvatures and normal curvature in the stroke's local directions (*cf.* Figure 8.30*(a,b,e)*). Similarly, demarcating curves, in spite of the more complicated patterns they trace along the surface, are clearly revealed by the values of the curvature in the curvature gradient direction (*cf.* Figure 8.30*(c)*), which is perfectly consistent with their characterization by Kolomenkin *et al.* [KST08]. Finally, in the thorny case of the dual identification of the occluding contours along with their suggestive extensions, we observe that the surface slant along with its second derivative play a capital role in their recognition (*cf.* Figure 8.30*(d)*). This indeed agrees with Koenderink's and DeCarlo's definitions [Koe84, DFRS03].

Once put into perspective, all the previously detailed observations attest the efficiency of our learning approach and the suitability of our geometry-infused stroke description. They also tend to show that despite the storage of all the geometric information as sampled buffers, aliasing-related issues impede neither the performance, neither the quality of our strong classifiers' predictions.

Figure 8.31 finally displays an illustration of the "transfer" of the style captured by the classifier from one training geometric model to a brand new one whose fineness of tessellation greatly differs. Many geometric properties – especially all curvature-related ones – therefore evolves in different domains and the domain par-

		User-guided selection		Validation on training data				Validation on new data	
		N_{total}	$N_{selection}$	Precision	Recall	Accuracy	Error	Validation 1	Validation 2
Figure 8.24		2,121	399	0.4787	0.9139	0.8934	0.109		
	*	2,121	399	0.7223	0.9821	0.9544	0.0408		
Figure 8.25		1,997	599	0.616	0.8744	0.8583	0.124		
	*	1,997	599	0.7926	0.9706	0.9493	0.0363		
Figure 8.26		2,081	714	0.5672	0.7219	0.7765	0.1729		
	*	2,081	714	0.7627	0.9112	0.88	0.0684		
Figure 8.27		1,884	494	0.4393	0.6401	0.7882	0.1418		
	*	1,884	494	0.7555	0.9044	0.9039	0.055		
Figure 8.28		8,102	2,004	0.3982	0.7607	0.8202	0.1804		
	*	8,102	2,004	0.5774	0.8583	0.9121	0.0727		

8.29 Performance Evaluation of the Presented Boosted Classifiers. Shown above are several statistical classification quality measures intended to quantify the reliability of the predictions of the classifiers trained in the context of the different application scenarios detailed in Figures 8.24 through 8.28. They are also to provide a complementary, more objective assessment of their achievements than the sole visual inspection of the drawings made of the lines they retain. Misclassification related measures such as precision and recall can only be proposed when the validation and training data sets coincide. In other cases, we instead inspect the relative proportions of the different line categories among the set of the *selected* curves. We notably expect this balance to be identified by the classifier through its training and remain comparable between the user's choice *(left column)* and the automatic selection by the classifier *(right columns)*. The measurements of the rows marked with red asterisks *(*)* have been computed such that the strokes are weighted in accordance to their length. Therefore, the longer lines show a more pronounced impact on the computations than shorter ones. Chief among the possible observations we can draw from these results, we note that our classifiers usually exhibit fairly strong recall values in all configurations. This fact is especially encouraging as recall can be thought of as the probability of a relevant curve being picked by the classifier, its high value attesting the completeness of the classification.



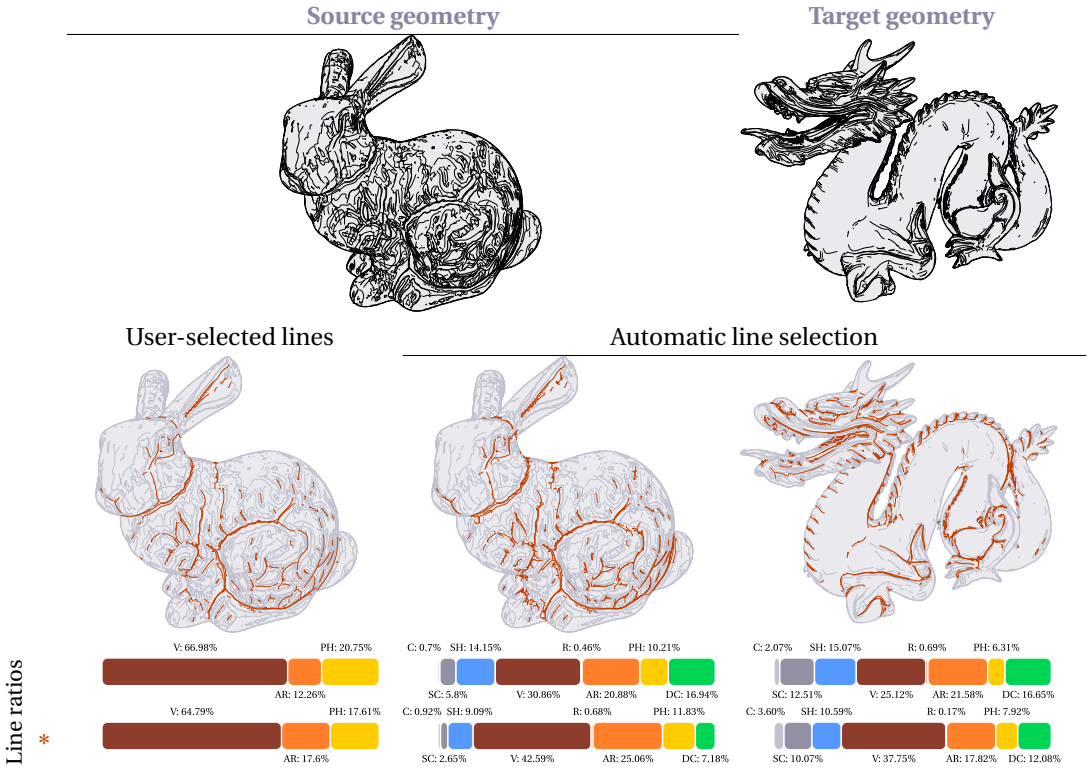
8.30 Breakdown of the Presented Strong Classifiers. The above charts detail the subset of weak classifiers, along with their established contributions, picked during the course of the ADABOOST algorithm in order to mold the strong classifiers whose results have been presented so far. We observe that in all of the five presented cases, only a couple of weak classifiers – three at best over twenty – ends up playing a prevalent role in the resulting strong classifier's decisions. We also note that they always seem in accordance with the task at hand: crease line detection *(a,b,e)* mostly resorts to principal curvature-related classifiers, whereas demarcating curves *(c)* are identified thanks to the scrutiny of the normal curvature in its gradient direction, and the occluding and suggestive contours *(d)* thanks to the inspection of the surface slant and its second derivative.

titions dictated by the gathered rules of thumb may not be valid anymore. Still, thanks to the preprocess of our G-buffers (*cf.* Figure 8.18) along with the enhanced generalization properties of the ADABOOST 's predictions, the application of our strong classifier to this new geometric context end up in remarkably consistent line selections.

8.2.4 Guiding the Creation of New Strokes by the Inferred User's Style

Last section detailed the user's style learning process and thoroughly analyzed the performance and how appropriate our learning framework was. From the set of conducted experiments and the breakdown of the established classifiers, we reached more than positive conclusions regarding our system's ability to properly predict the relevance of the lines used for validation. But foretelling the agreement of *existing* lines with the user's drawing is not sufficient though. What we need now is devising ways that exploit our classifiers' predicting abilities in order to generate *from scratch* lines that would successfully pass the classification test. With these new curves at hand, we could then easily extend the user's drawing and anticipate his/her choices.

A first step to reach that goal is to visualize in more detail the "search space" we have to explore. By search space, we mean the set of all positions acceptable by our classifiers, which are positions through which our new lines are likely to pass through if they also want to be deemed relevant by our system. Figure 8.32 presents such locations for all the validation examples shown in Section 8.2.3. On that occasion, we can also appreciate the impressive 3d consistency of the classifiers' responses across vantage points. This positive observation remains valid even in cases when the strong classifier has to detect widespread, or on the contrary very localized lines, as attested by the visualizations dedicated to the detection of the *Bunny* model's demarcating curves (*cf.*



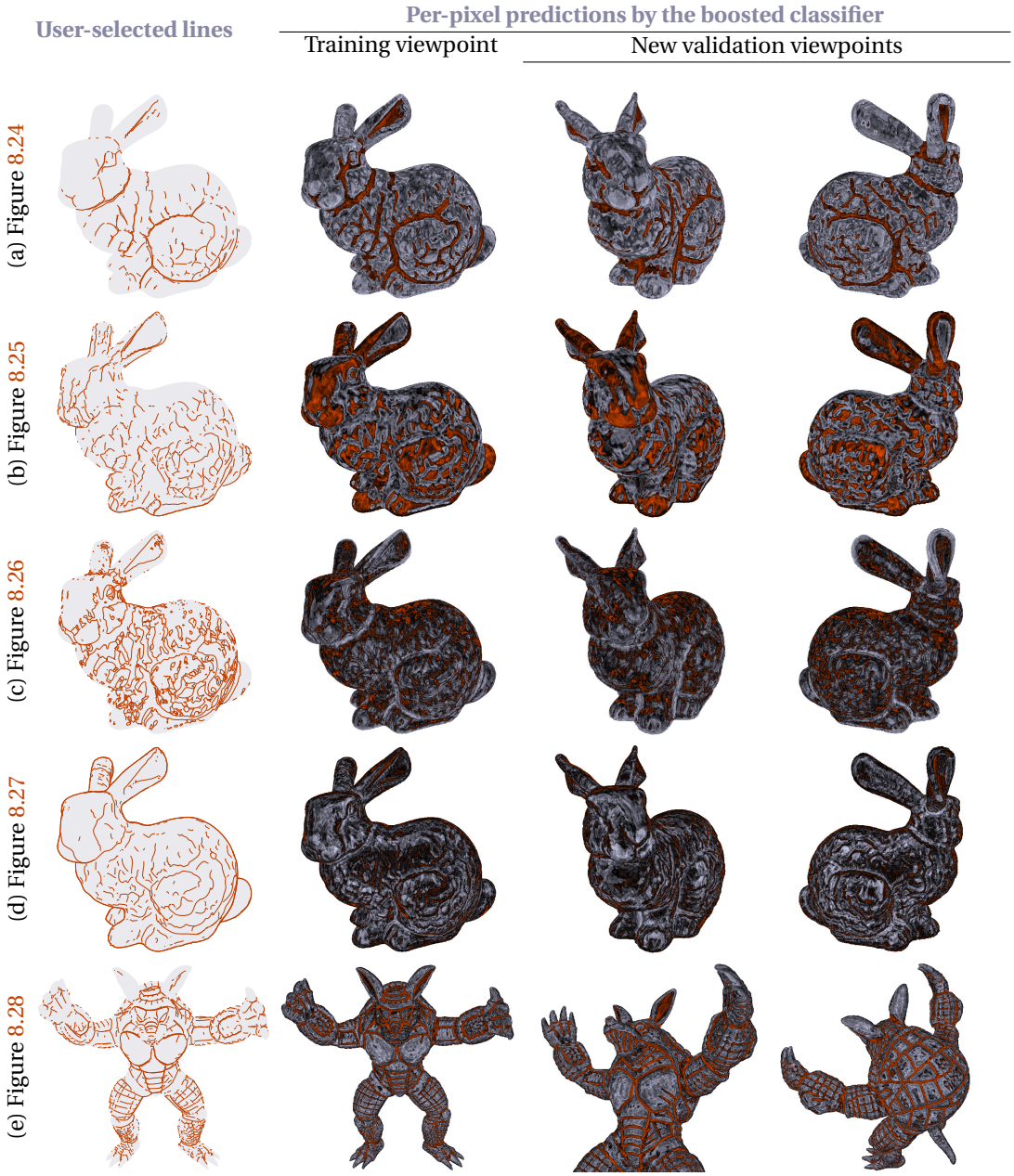
8.31 Example of Line Drawing Style Transfer. By applying our strong classifier to the signatures of strokes emerging from different models than the one used for its training, we can transfer the users' drawing styles to various geometric inputs. We owe this aptitude of our classifier to the ADABOOST learning approach which is renowned for its resistance to over-fitting while achieving impressively accurate classification performances. The line ratios of the different selections *(bottom rows)* show the balance between the line categories remain almost untouched in spite of the change of geometry, and that valley-like curves are deemed relevant by the classifier in all cases. As in Figure 8.29, the red asterisk *(∗)* indicates the row of ratios weighted by the length of the selected strokes.

Figure 8.32*(c)*), and occluding and suggestive contours (*cf.* Figure 8.32*(d)*). In order to obtain these different line prediction maps, features made of the concatenation of all the G-Buffers' values at each possible screen location have been considered and handed over to the strong classifier for validation. Since some of its constitutive weak classifiers' queries are directional and thus require the candidate stroke's local tangential direction information – *e.g.* the normal curvature in the candidate stroke's local direction, or the perspective depth derivative magnitude across it –, we instead divide the set of all possible directions in 10 even steps, conduct the classification tests for all of these candidate directions and finally keep the maximal positive response.

One could expect to directly use these images (an example being given in Figure 8.33) in order to find the new lines we are looking for – by possibly finding ridges lines and proceeding to their vectorization after thinning and the chaining of their pixels. Alas, computing such images is way too expensive to straightforwardly resort to them and would compromise our ultimate goal to embed our classification system into an interactive drawing application. While the information they convey is crucial, we still have to exploit it in a different, faster fashion, even at the cost of approximations. Doing so would not be that great of an issue since, as part of an interactive framework, our new lines could easily undergo some editing by the user and then also participate in the training process in their turn.

8.33 Line Prediction Map.

Keeping these computation constraints in mind, we expose in the following paragraphs our triangulation-based approximation scheme aiming at the discovery of strokes that satisfy our acceptance criterion while limiting the number of classification queries at the very most. First, we still need to find a discrete set of valid positions as the starting point of our computations. For that aim, we test randomly generated positions and



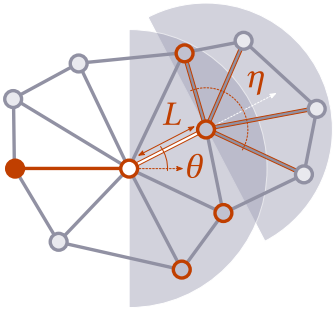
8.32 Visualizing the Strong Classifiers' Inferred Mapping. Above are representations as false-colored images of the predictions of the different strong classifiers whose results have been presented in Figures 8.24 through 8.28. In order to establish these, the strong classifiers have been run over each pixel of the G-buffers, the concatenation of the geometric data stored at each pixel constituting a complete feature vector to which all the weak classifiers' queries can be addressed. Red pixels correspond to pixels where the classifiers return a positive response, contrary to grey pixels representing irrelevant locations. The pixel saturation encodes the confidence of the prediction: the more colorful the pixel, the more pronounced is the classifiers' output at the corresponding location.

record the ones retained by the strong classifier. We call such locations *seeds* (*cf.* Figure 8.35*(b)*). To find these, we keep on randomly sampling the imaging plane and test candidate positions against the classifier until a sufficiently large amount of seeds has been reached – 2,000 in all our examples. Once gathered, we compute the Delaunay triangulation for the seed point set as a way to know the inter-seed adjacency relationships and efficiently traverse this set by navigating from one point to its nearest neighbors (*cf.* Figure 8.35*(c)*).

Ultimately, the strokes we are looking for are to follow connected paths through the triangulation. As such, before its traversal, we still need to assess the relevance of the triangulation edges with respect to the classifier since we aim at the detection of edge chains that remain in the valid portion of the line prediction map. We carry out this stage similarly to our stroke validation step (*cf.* Figure 8.17), by sampling each edge segment

and collecting all geometric data lying at each of these sample locations to form the segment's representative feature vector which is finally passed to the classifier in order to evaluate its relevance.

Once the edges that fail the classification test have been discarded (displayed as dashed lines in Figure 8.35*(c,d)*), we can now walk through the seed points' Delaunay triangulation and establish chains of edges that would then constitute the final strokes' *skeleton* (portrayed as the red polylines in Figure 8.35*(d)*). We conduct our edge chaining as a greedy process. First, we sort the valid edges in order of decreasing confidence – confidence referring here to the continuous value returned by the classifier whose sign indicates the tested observation's category – and grow the chain from its two extremities. Once a new seed point is reached, one half of its outgoing edges leading to unvisited seeds are considered and attached a score. A graphical representation of the details behind the computation of such scores is provided in Figure 8.34.



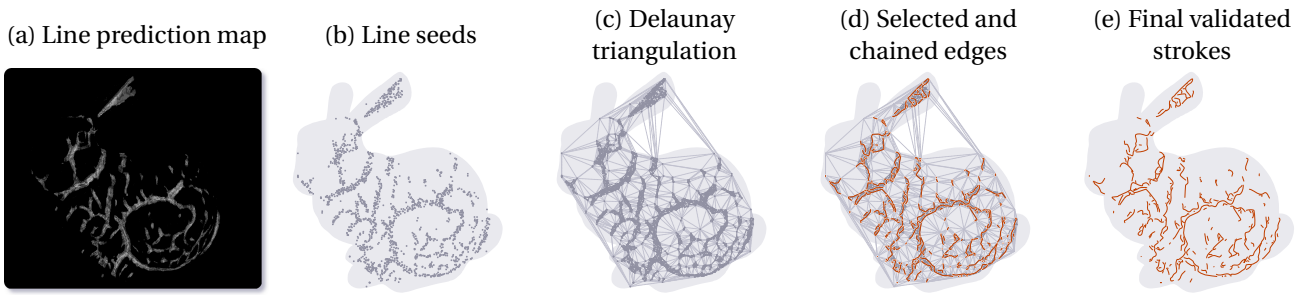
8.34 Edge Chaining.

Edges already belonging to the chain are displayed as thick red lines, whereas the extremities of the candidate edges appear as grey dots with a red ring. We chose candidate edges so that to prevent two successive chain segments to form an acute angle. Their end vertices therefore fall in the half circle (represented as the light grey region) centered on the last added seed (white dot circled in red) aligned with the previously accepted segment. A candidate edge's score accounts for the edge's length L, angular deviation from the ongoing chain θ and "continuity capacity" η. The so-called continuity capacity of a candidate edge designates the number of edge candidates it will introduce in the next chaining step. For a given candidate edge $\mathbf{e}$, all this information is combined into a single score value $s(\mathbf{e})$ as follow:

$$s(\mathbf{e}) = \alpha \cos\theta(\mathbf{e}) + \beta \frac{\eta(\mathbf{e})}{1+\tilde{\eta}} + (1-\alpha-\beta)\frac{L(\mathbf{e})}{10^{-6}+\tilde{L}},$$

where $\tilde{\eta}$ and $\tilde{L}$ denote the median vertex valence and the median edge length computed over the entire triangulation – only valid edges are taken into consideration for the latter –. These values are used in order to normalize the different factors' contributions and thus facilitate their tuning. α and β correspond to predefined weights lying in the $[0,1]$ interval and are meant to control these contributions. In all the presented examples, we chose $\alpha = 0.5$ and $\beta = 0.25$. Once identified, the candidate edge with maximal score is appended to the growing edge and all remaining edges' end vertices are marked as visited. The process continues until no further edge can be added and once one side of the initiating edge cannot advance any further through the triangulation, the same procedure is launched from its other side. Other edges are considered once the two sides of the current edge have been extended, and at the condition none of their two extremities has already be flagged as visited.

Once all edges have been examined, we fit splines made of successive Bézier curves to the sequences of seed points making up the computed chains (*cf.* Figure 8.35*(e)*). Results of this approach are presented in Figure 8.36. Note that the spline fitting method we currently use is interpolating the seed point sequences rather than approximating them, and that no filtering of the obtained strokes with respect to their length has been performed. Further investigating these two concerns could greatly improve the created drawings' overall appearance although they still prove rather satisfactory in their current state. Employing a more flexible spline fitting approach seems the most reasonable option for mitigating the broken aspect of our currently generated lines and that way, achieving more eye-flattering results. Indeed, for the very same reason that incited us to propose this point-based line generation – namely the costly evaluation of the complete line prediction maps –, the conceivable solution of performing a density-controlled relaxation of the seed points – similar to Llyod's relaxation method which is commonly used for adaptive sampling or stipple-based rendering for instance – cannot be retained.



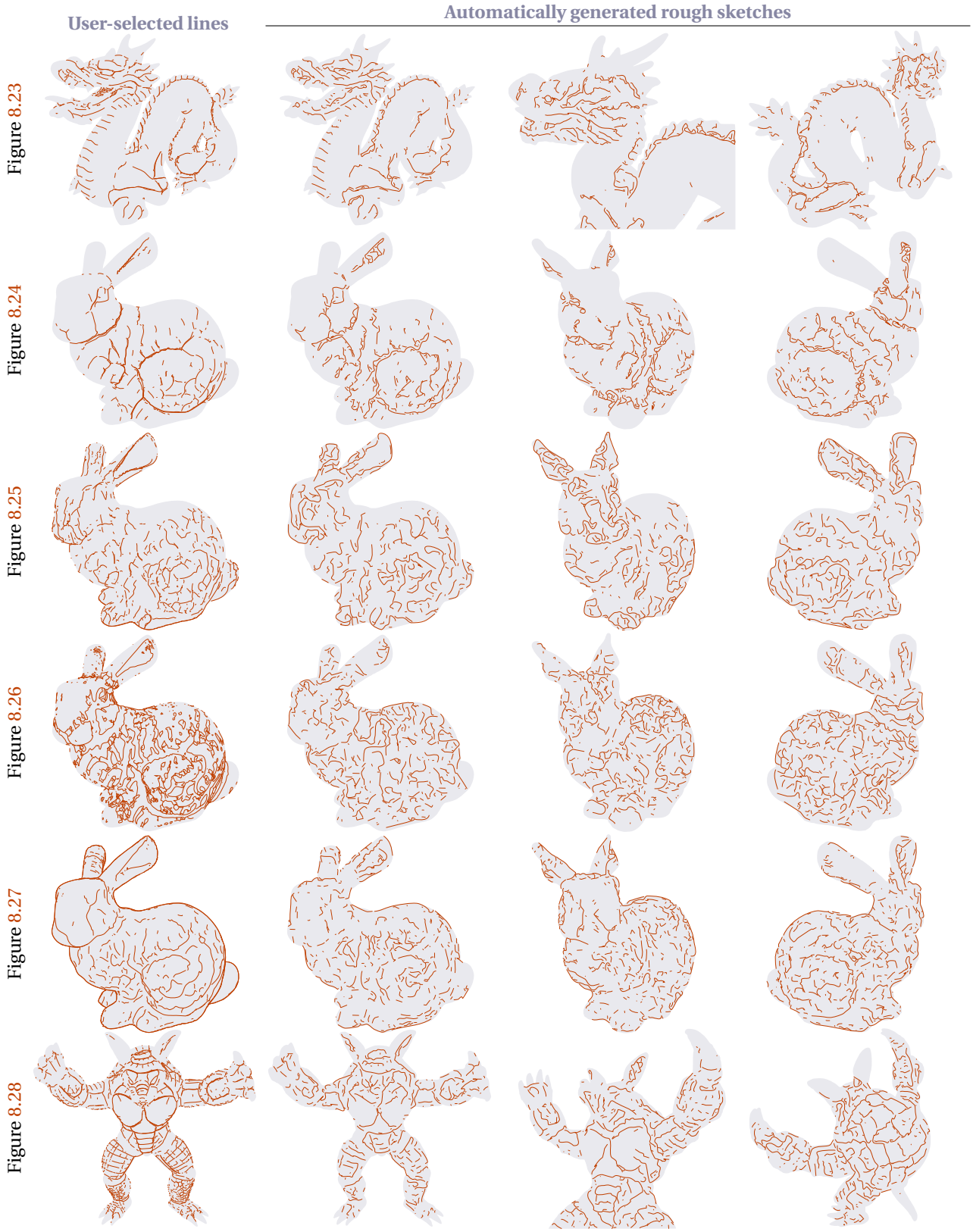
8.35 Overview of our Drawing Creation Process. The goal here is to find an as fast as possible way to exploit the spatial arrangement of our strong classifier's positive predictions (displayed in white on *(a)*). Indeed, we ultimately want to create new strokes running through these locations and that are therefore likely to pass the classification test. Alas, the computation of complete line prediction maps is prohibitive for our interactivity objectives. Consequently, in order to alleviate computation costs as much as possible, we propose a triangulation-based approximation of our problem. First, we draw a predefined number of positively-classified positions *(b)* and compute the Delaunay triangulation of these *seed points* in order to efficiently get their neighborhood system *(c)*. Valid edges of the triangulation – edges retained by the classifier, displayed in thick lines contrary to the rejected ones shown as dashed lines – are then chained in a greedy fashion. These chains reveal the generated strokes' *skeletons*, appearing as red polylines in *(d)*. New drawing strokes are then obtained via spline fitting. They are finally handed over to the strong classifier for final validation *(e)*.

8.3 Ongoing Research and Closing Remarks

In the last chapter, we presented our latest research dwelling on the line-based renditions of 3d geometry. Aiming more at its "artistic" representation in the context of non photo-realistic rendering than accurate shape communication, we therefore opted for having users and artists be an integral part of our method, for they are the ones who actually have to take the most determining aesthetical decisions. Our work bears obvious resemblances with Lum *et al.*'s approach [LM05], but also clearly extends it by recasting the whole question of line detection into a binary classification problem. Our main contributions therefore stem form that crucial difference as we do not aim at only *selecting* lines from the analysis of the users' inputs anymore, but at the complete *creation* of the drawing with an as seamless as possible interaction with them. By proposing the use of the ADABOOST learning meta-algorithm, our method can infer complex, user-dictated rendering rules regarding the positioning of the relevant lines not only with great accuracy, but also with improved robustness and generalization properties. Our work is henceforth among the few attempts at line drawing generation which attach so much importance to the artists' wishes. In the context of their user study *"Where Do People Draw Lines?"* [CGL+08], Cole *et al.* briefly explores the possible capture of their test subjects' drawing styles by constructing a decision tree taking as arguments some of the geometric measurements we also retain and aiming at the prediction of the likelihood a line crosses a particular image pixel. However, they did not conceive their test as an exploitable tool for the creation process of a drawing, but for the most part as a enlightening view for the analysis of the drawings once established, and the assessment of existing line drawing algorithms' relevance.

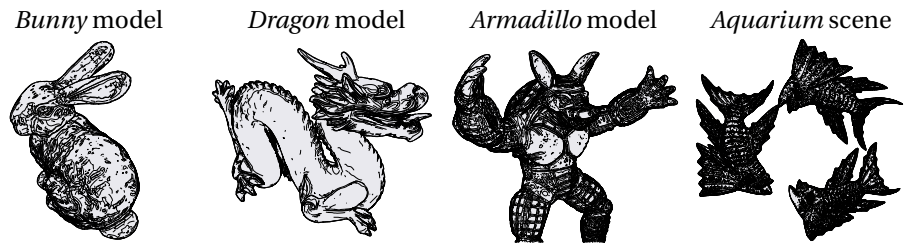
However, being part of our most recent research, the current state of our work lacks the level of polishing one could expect, and what was presented in this thesis was merely but a proof of its concept. In order to fully complete it, the most obviously and determining step would be the integration of our stroke classification framework to a complete sketch-based interface, and observe whether its decisions facilitate, or on the contrary impede its users' work flow. A second natural step would be to conduct a user study in order to truly validate our claims regarding the actual assistance our approach provides.

But aside from the engineering work needed in order to have a fully functional graphical user interface that would allow stroke tracing and edition, some additional work for the time-optimization of our code would also be required as demonstrated in Figure 8.37. While all presented cases correspond to quite cluttered drawings unlikely to have been manually drawn by an artist, the time taken for the training of our different classifiers clearly seem to compromise its direct use in an interactive drawing application. The ADABOOST



8.36 Examples of Prediction-Driven Drawings. Shown above are instances of automatically generated line drawings established in accordance with the sole consideration of the trained strong classifiers' outputs. Note that no length-based filtering of the resulting strokes has been performed here.

algorithm is by essence of linear complexity, and so is the expectation-maximization procedure used for fitting gaussian-mixture models to the weight distributions at each boosting round. While sequential by nature, Freund and Schapire's algorithm easily lends itself to parallelization since each weak learning procedure can be run on independent threads at each of its iteration. This could be prove quite beneficial for enhancing the performances of our current implementation which only provides stroke signature caching and the resort to tabulated gaussian functions as sole means of acceleration.



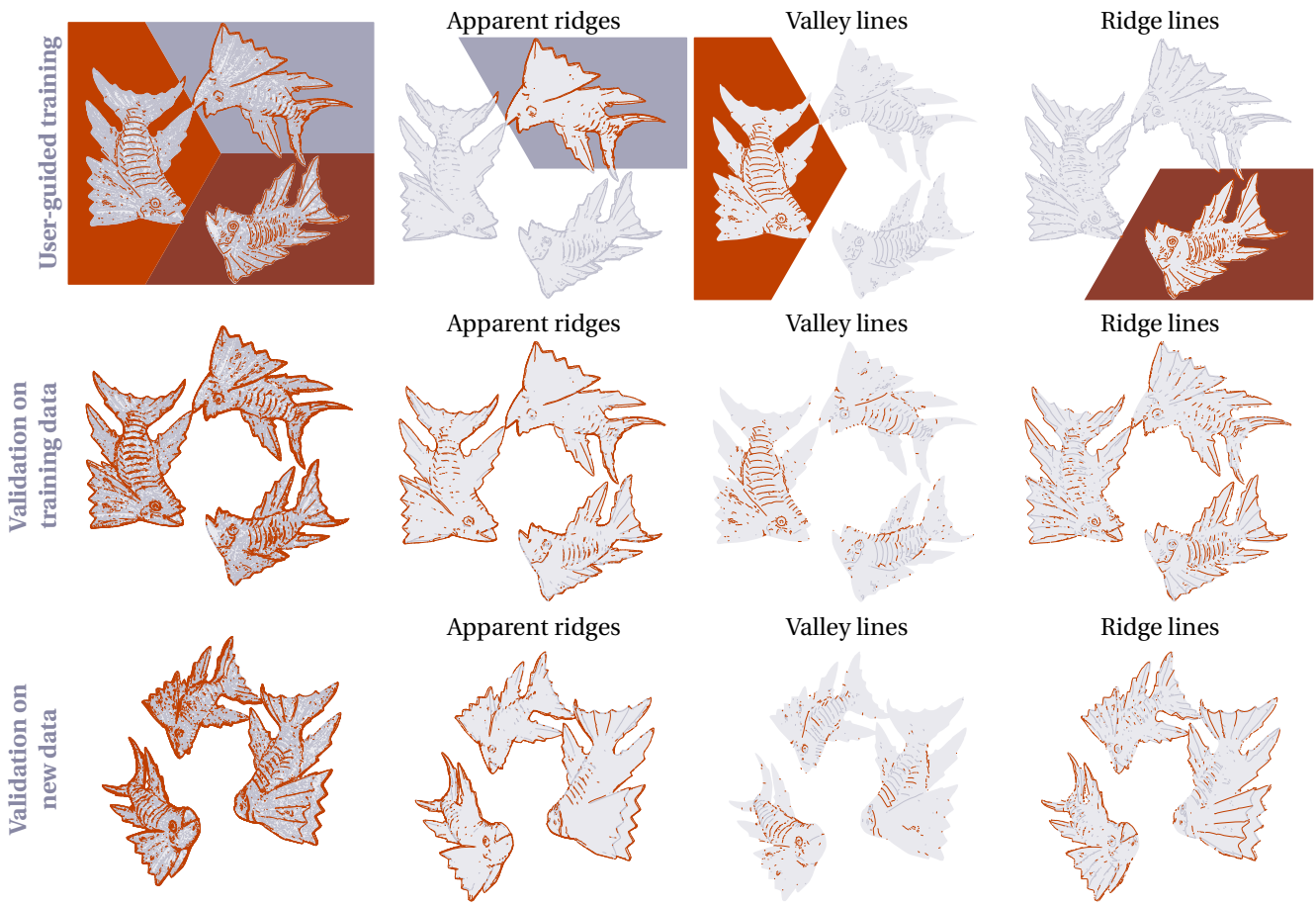
$N_{\text{strokes}} = 2,098$ 13.76s | $N_{\text{strokes}} = 4,495$ 55.27s | $N_{\text{strokes}} = 9,440$ 221.97s | $N_{\text{strokes}} = 25,685$ 1572.89s

8.37. Timings of our AdaBoost Learning Procedure. All experiments were conducted on a dual-core laptop (each processor being of frequency 2.53GHz) with 3.6Go of RAM. In all cases, classifiers made up of 20 trained rules of thumb are built and the mixture-of-gaussian fitting procedure involves up to 8 iterations.

Additionally, our current technique can be improved upon on a more theoretical level. The consideration of other and complementary features – such as textural statistical measurements for instance – naturally constitutes a way of improving our stroke representation, but also the resort to more intricate tests – statistical tests typically – could greatly improve the relevance of our classification results. But bearing in mind the concerns mentioned above, the direct addition of these supplemental descriptive features and learning procedures would obviously worsen the time performances of our technique. Instead, a careful selection is needed and additional tests must be conducted in order to keep only the most useful, and most often retained features and tests.

Another promising perspective for improvement would be to consider the locality of the users' strokes on the screen in order to refine the learning of our classifier. Alas, in the absence of any screen location-related information to be found in the stroke signatures, our current implementation cannot capture "spatially-varying" drawing styles: an example is proposed in Figure 8.38 which shows the outcome of our approach when the types of the selected lines directly depend to their locations in screen-space. For handling such cases, instead of changing the strokes' geometric footprints, we could have several strong classifiers trained for distinct screen locations, and combine their respective decisions in a way similar to a distance-based, radial basis function. The training stage of the classifier could directly account for such concerns by tweaking at each iteration the weight distributions according to the screen-space distance between a given observation and the different classifiers' assigned location. Once our technique can support this kind of position-dependent drawing styles, it could then easily overcome the issue – which is often encountered in most of automatic line detection methods – of finding a satisfactory global threshold. The assignment of the strong classifiers' locations is still open though, and could either be explicitly handled by the user, or possibly adapted by a cascade of classifier-kind of approach.

Finally, a last perspective for our example-based, interactive approach would be to focus of the visual appearance of the lines. Beside the need of a more visually-pleasing spline fitting for our line generation step, we could also propose a similar machine learning approach in order to capture the possible correlations between the strokes' appearance attributes – *e.g.* 2d curvature, stroke width – and the geometric features they depict – *e.g.* surface curvatures, shading –. Examples of deriving the strokes' visual aspect from information relative to the 3d setting of the scene have notably been explored by Goodwin *et al.* [GVH07]. By having the width of their strokes controlled by their iso-phote distance – which considers depth information, the surfaces' radial curvature, and lighting conditions –, they manage to infuse a very convincing, hand-made look to their automatically generated drawings. We believe our machine learning approach to line drawing capture could prove efficient for the two components of a line's visual aspect [SS02], from the finding of its path as demonstrated here, to the control of its style.



8.38 Capturing Spatially-Varying Drawing Styles. Currently one of the most severe limitations of our method's current implementation is its inability to capture screen position-dependent rendering styles. This issue is illustrated here with a rather unnatural training line selection *(top)* made of apparent ridges, valley and ridge lines. It was done so that each of these line categories would lie in a specific and localized portion of the screen. We see that even if our classifier identifies rather well the relevant types of lines, it however does not account for the specific layout they exhibited in the user's selection *(middle, bottom)*. We believe that the use of a series of trained classifiers, each of which would be in charge of a given area of the screen, may handle this issue, and ultimately lift the global thresholding issue of most automatic line extraction techniques.

CHAPTER 9

Conclusion

the present thesis explored the issue of example-based content generation in the context of non photo-realistic rendering. Computer-assisted creation of graphical content is now ubiquitous: creating and retouching images and videos by means of a computer is now almost indispensable to meet nowadays' quality standards and fulfill the wishes of an ever-growing audience. This trend pervades all domains of visual creations, from still visuals, to hand-drawn animations, to movies. The strength of this phenomenon has been witnessed very recently by the unmatched success of recent CG-heavy blockbusters, as well as the rapidly growing use of stereoscopic vision apparatus in theaters.

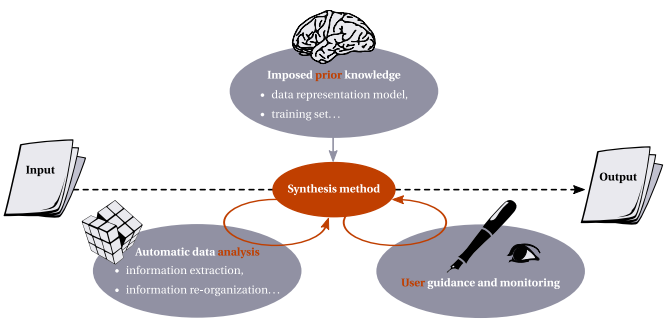
Partly owing it to their ever-increasing computational abilities, computers nevertheless require efficient tools to be fully exploited. Beside the purely procedural creation of content, most of them need inputs to work upon in order to either extend or enhance them, or guide the synthesis process. We referred to these algorithms as *example-based approaches*, often praised for the intuitiveness of use. Back in the introduction, we determined three main sources of information such methods could have recourse to (*cf.* Figure 9.1):

We also focused on very specific sub-domains of computer assisted rendering: non photo-realistic rendering, and perceptually-driven image enhancement. Both contexts are especially enlightening for our study for the subjectivity of their quality assessments. Unlike techniques aiming at the photo-realistic reproduction of the interplay of light and material, expressive rendering instead concentrates on the emphasis of the information to be conveyed, and henceforth strives to the identification of the *meaningful* parts of the input. This almost outguessing objective is by definition an ill-posed problem, and directly reveals the stakes of finding a good balance between the three aforementioned sources of information, for users cannot possibly be entirely overlooked. The discovery of a satisfactory point of equilibrium between them was thus at the core of this thesis.

9.1 Contributions and Perspectives

We briefly summarize in the following the contributions presented alongside the chapters of this thesis. Each of them explores a different positioning regarding the respective contributions of the three information sources detailed above. As a side note, it is interesting to observe that in some cases the reconsideration of our choices of emphasizing automation at the cost of user control, or data-driven analysis over imposed priors constitutes either compelling alternatives for improvement, either natural solutions to overcome our current methods' shortcomings.

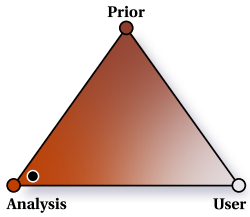
- the direct analysis of the provided input,
- the application of prior knowledge to it,
- or its process according to the user's guidance.



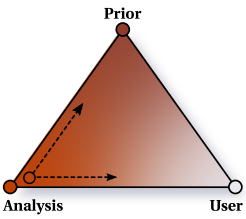
9.1 The Information Source Triangle.

9.1.1 Pattern Recognition for the Automatic Synthesis of Raster Shape Arrangements

In Chapter 3, we proposed a new by-example method for 2d shape arrangements in raster format. In that work, we aimed at providing an as automatic as possible approach which did not resort to any user inputs, nor assumptions regarding input features' structure or layout. Texture synthesis is the quintessential illustration of the "by example" approach, the visual complexity of textures being hard if not impossible to mathematically describe or handle. We aimed at focusing on a specific class of textures because of the variety of their possible visual appearance. We therefore explored the case of textures consisting in the arrangements of two-dimensional, partially overlapping shapes. In order to guide our analysis and go without external sources of information, we assimilated the relevance of the texture's features with their repetitiveness. We thus employed a computer vision inspired gait to our problem, which relies on the pairing of gradient-based descriptors, transformation clustering and image segmentation. If we were to visualize the positioning of our raster technique with respect to the three possible sources of information, it would lie at one extremity of the spectrum as it solely relies on the analysis of the sample. We chose to not require user intervention as the task at hand did not involve any specific aesthetical concerns, or predetermined assumptions to preserve the generality of our method.



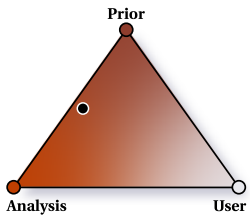
Possible Improvements Our current work can easily be extended before considering changing our complete automation objective. More robust and color-aware local descriptors along with more involving matching strategies could be explored in order to improve our detection of repetitive shapes. Our fairly limited transformation model between shape duplicates could also be changed (only rigid transforms are currently supported) in order to handle local deformations which could be detected during the transformation clustering stage. Finally, the use of an approximate optimization framework (such as the α-expansion algorithm for multi-label segmentation) could favorably replace our greedy approach for our final shape extraction stage.

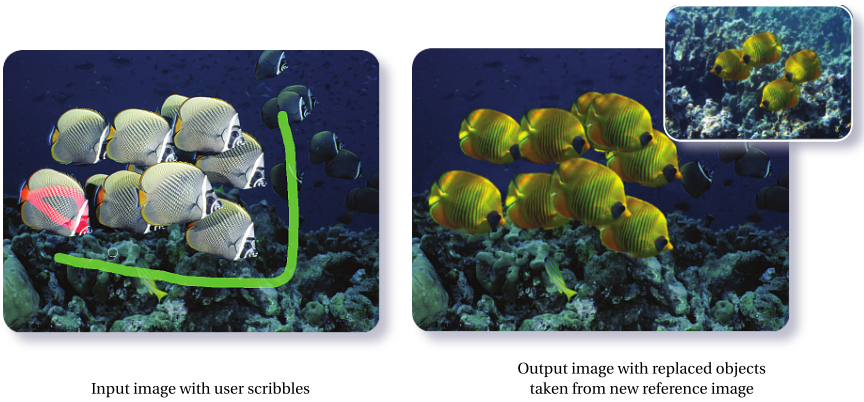


Revisiting the Balance between Information Sources Our research is characterized by an explicit will of automation for aesthetical concerns do not enter into consideration for a task such as texture synthesis, and lifts the constraints due to spatial layout models as in [LCT04, LLH04]. However, we reached a point where our method could easily be enhanced by slightly relaxing its automation ambition. Considering user annotations either for guiding the analysis step (such as in Cheng *et al.*'s impressive *RepFinder* tool [CZM+10]), either for the correction of our extracted shape in a later stage could greatly benefit our current technique for it to handle noise and partial matching. Granting more importance to prior knowledge could also constitute an interesting track to explore as repetition-driven models for image shape and appearance do exist such as the epitomic analysis of images [JFK03, KWR06]. However, even though one could marvel at their image compression and reconstruction performances [WWOH08, WHZ+08], they lack in their current stage the structural information required for the generation of new content. Drawing inspiration from such models and making them more amenable to our texture synthesis purposes nonetheless constitute promising albeit challenging leads for future work.

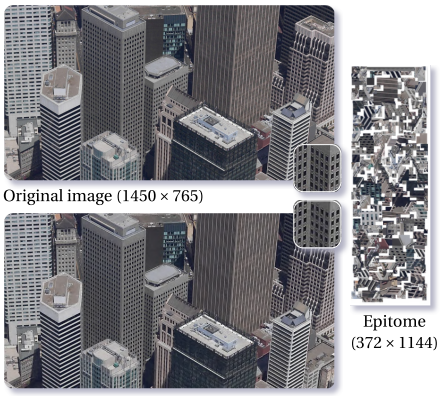
9.1.2 Spatial Modeling for the Automatic Synthesis of Vector Element Arrangements

One of the advantages of our shape-driven approach to texture synthesis was the establishment of a meta-representation of the input samples that no longer evolve at the scale of independent pixels, but rather at the scale of their actual constitutive shapes. Via this alternate representation, information of an higher level than color co-occurrences inside fixed-sized neighborhoods could be considered for re-synthesis: the





9.2 *RepFinder*: Finding Approximately Repeated Scene Elements for Image Editing [CZM+10].

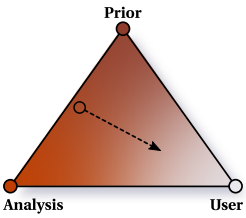


9.3 Factoring Repeated Content within and among Images [WHCO08].

relative positions of shapes are one of them and were first straightforwardly handled by greedily extending the Delaunay triangulation of their mass centers.

Research detailed in Chapter 4 went a step further in the analysis of the spatial layout of such arrangements in the context of vector textures. In contrast with existing shape arrangement synthesis methods, we performed a global statistical analysis by fitting Strass-hardcore spatial models to our input arrangements. Prior log-likelihood maximization for the estimation of their optimal parameters with respect to the provided examples, we brought the arrangements' elements into a set of appearance categories enabling the capture of specific co-occurrences at the light of the elements' visual aspect. This appearance analysis step was made entirely automatic by the resort to *a contrario* clustering methods. As for the previous research, this work also aimed at complete automation, with a much more pronounced reliance to prior knowledge however through the use of a predefined model for the analysis of its inputs.

Possible Improvements As most model-based methods, our vector arrangement synthesis technique is naturally bound by the limits of the complexity of its model. Ours especially aches for simplicity as statistical fitting is easily prone to suffer from the curse of dimensionality due to the voluntarily limited inputs. Even though the outputs obtained by our method exhibit more variety than triangulation-based techniques', our model performs rather poorly on strongly regular inputs. Considering element interactions of an higher order than pairwise relationships could be explored to better handle structured arrangements. Our early appearance categorization stage could also be extended: currently analyzing low-level features drawn by Julesz'experimental studies, it could be altered so as to consider other appearance features and extended to handle not only line-based elements. A last, and certainly most important improvement would be to deal with the issue of arrangements showing a pronounced spatial clustering of their elements by providing a multi-scale analysis strategy.

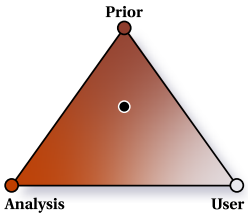


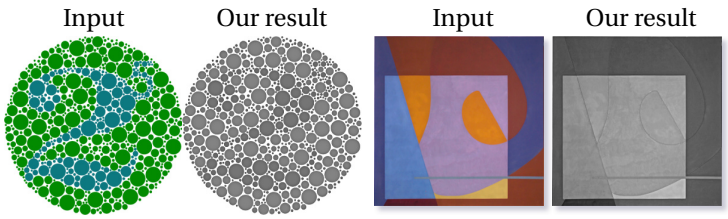
Revisiting the Balance between Information Sources Entirely absent from all our method's processes, users could be given a more than beneficial role to play. In a corrective step, they could notably monitor and correct if necessary our element appearance categorization. Ill-clustered elements could easily be assigned their correct category by the sole means of a couple of scribbles for instance.

9.1.3 Apparent Grey-Scale: A Fast Conversion for Images and Videos

Chapter 6 thoroughly presents our *Apparent grey-scale* technique for the conversion of color images to achromatic images. Unlike most previous methods emphasizing discrimination between grey values associated to distinct colors and solving the luminance value assignment issue by optimization, we instead favored the

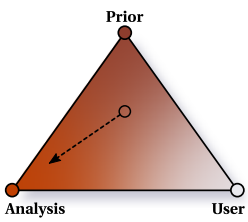
preservation of the *sensation* elicited by the color image in terms of brightness, and strove to propose a perceptually relevant color to grey-scale mapping. Several entoptic effects have been considered for that aim. Of utmost importance, the Helmholtz-Kohlrausch effect (stating that of two equiluminant colors, the one of maximal colorfulness tends to be perceived as brighter) is notably embedded into our computations via Nayatani's brightness predictor. In a second stage, lost local chromatic contrasts are introduced back by the addition of Cornsweet edges via adaptive, color contrast-controlled hierarchical unsharp masking. The Craik-O'Brien-Cornsweet effect is a strong optical trick increasingly used in computer graphics which consists in the elicitation of a wide-range illusory brightness perception by the localized edge-like perturbations of an image's luminance profiles. Taking advantage of these perceptual considerations, we proposed a simple grey-scale conversion approach of linear complexity which exhibits both spatial and temporal coherence and can therefore be directly applied to videos.





9.4. Failure Cases of our Current Grey-Scale Conversion Method. More than unfortunate turns of events, these two results clearly highlight the theoretical weak points of our local contrast enhancement step, namely its impossibility of assigning different grey values to spatially distant colors, and its moderate effect in presence of colors eliciting almost the same brightness percept.

Possible Improvements Our method suffers from the very localized character of its contrast enhancement step. While it performs good on complex, real-life imagery, it still may fail to deliver the required amount of enhancement for clear discrimination when colors either do not share a common boundary, either roughly share the very same perceived brightness according to the perceptual studies we rely upon. Our converter's performances have notably been thoroughly and independently studied by Čadìk's comparative studies [Čad08a], and while it confirms the general pertinence of our approach, also reveals its weakness in such cases. Managing to better handle these and allow for more contrast distortions while keeping our current lightness ordering and not spoiling the perceptual relevance of our results certainly constitute interesting future work. Lastly, our contrast enhancement resorting to a high-frequency reinforcing filter is likely to create halos and emphasize image noise. This becomes especially detrimental to the quality of heavily compressed videos processed with our method. Part of these concerns were subsequently dealt with by Ritschel *et al.* [RSI+08] in the context of three dimensional scenes. We could however still address these issues in the more common case of 2d inputs and draw inspiration from advanced, noise-limiting filtering techniques.

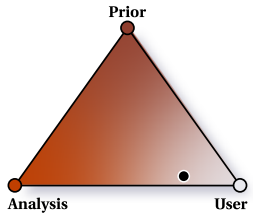


Revisiting the Balance between Information Sources In our approach, users do not take the center stage but still enjoy some control over the results especially at its second critical step for the adjustment of the local contrast enhancement. They are being provided only an as limited as possible set of parameter though, and their task is voluntarily confined to parameter tweaking for adjusting the enhancement strength and spatial range regarding the final output's resolution and display conditions. While one could explore the possibility of having them fulfil a more capital role (something closer to recent scribble-based approaches for image edition, like color transfer for instance [WHCO08]), we would rather privilege the analysis of the inputs in order to solve the limitations of our approach for it would make the conversion task uselessly tedious. But one has to bear in mind that one of our method's main strength lies in its linear complexity, considering wide-range chromatic contrasts (provided we avoid the resort to sampling that would endanger its frame-to-frame coherence) would make it of quadratic complexity.

9.1.4 Machine Learning for Interactive Line Rendering

The last part of our research, exposed in Chapter 8, revolves around the user-guided creation of line drawings for the expressive renditions of 3d geometry. In contrast with all other techniques presented thus far, it con-

ferred a privileged place to its users by proposing an interactive framework where artists start their creations by directly drawing strokes over the screen-projected models. Our method would then draw inspiration from their inputs to anticipate their actions. This user-centric vision of the problem of line-based renditions of 3d geometry has very scarcely been effectively handled, and could prove to solve many issues outlined in recent user studies [CGL+08, CSD+09]. For our "drawing anticipation" purposes, we considered our question of line extraction from a machine learning perspective, and cast it as a binary classification problem. By training clas--sifiers on users' inputs and feedbacks, our technique establishes correspondences between relevant strokes and the surface properties at their locations. We used the widely-acclaimed ADABOOST learning procedure to build surprisingly accurate classifiers from a set of simple rules of thumb involving the surfaces' geometrical properties. Users' drawings can then be extended by running the trained classifiers in the remaining parts of the scene.



Possible Improvements As unfinished research, the current state of our work naturally leaves much to be improved upon. Its actual integration into a fully functional graphical interface would be a first step. Conducting an user study for the assessment of its suitability when applied to a more realistic application scenario would also complete the validation provided in this document. Depending on the outcome of such an experiment would also hint what improvements to bring and whether the balance between the information sources would be necessary. On a more immediate level, we could further improve the geometric description of the strokes by either adding or proposing more discriminative features, as well as devising other learning procedures for proposing more efficient rules of thumb. A last interesting perspective would be to propose a similar example-based learning for the strokes' appearance attributes and try to unveil possible correlations with the scene's local geometrical properties.

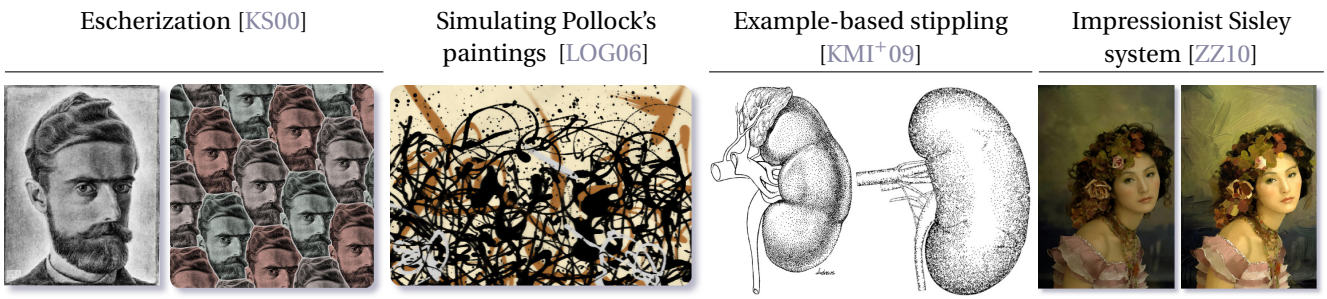
9.2 Discussions

More than the simple collection of independent research, the aim of this thesis is to provide a reflection on the adequate collaboration between machines and their users, and how example-based applications can take advantage of the sources of information at their disposal. Either thanks to the input itself by way of its raw analysis, to its understanding by fitting models to it, or to the explicit user indications, techniques can employ different strategies for extracting what constitutes their inputs' *essence* and infuse it to their outputs. But taking a specific strategy has direct impacts on the method's efficiency and applicability and should be carefully been decided with respect to its final application purposes. The successes and pitfalls encountered during the course of my thesis and detailed in the present document are illustrations of this fact, and give rise to a questioning that is discussed in the following.

In our terminology, an *example-based* technique qualifies methods that do not create content from scratch, but are instead provided examples they are supposed to either imitate, either enhance. The concept of *imitation* refers here to the ill-posed problem of inferring the properties of an unknown generative process from the visual aspect of its outputs. On the other hand, *enhancement* implies the improvement of the inputs themselves. Similarly, it then requires a moderate knowledge of the inputs' generative process for preventing their alienation during their transformation, as well as the consideration of objective or subjective criteria (may they be of perceptual or aesthetical nature) for quantifying the very notion of improvement.

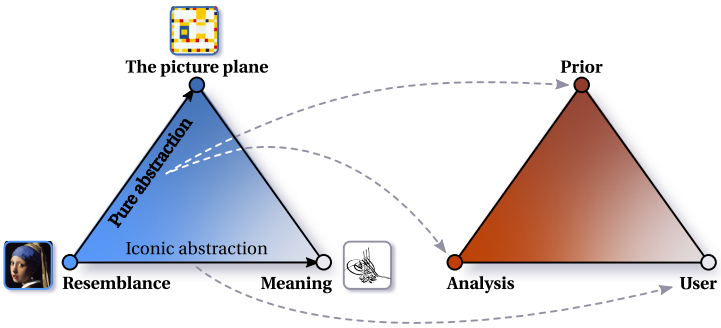
9.2.1 From Example-Based Synthesis to Artistic Style Capture

Most of the research I conducted evolves in the domain of expressive rendering. So far, this field has been presented from its sternest side: the identification and manipulation of the informative content of imagery. But only alluding the expressive rendering's quest for functional realism [Fer03] and passing over its intensive, art-driven research would not do it justice. Indeed, its literature abounds with work aiming at the aesthetical processing of imagery, and the devising of relevant methods for that purpose. Its artistic velleities shed another



9.5 "Reverse-Engineering" Artists. Intensive research in non photo-realistic rendering strives to the emulation of a specific traditional medium, but instead at the automatic reproduction of the drawing styles of specific artists. Of all the techniques presented above, the computer-assisted imitation of Jackson Pollock's paintings using fluid mechanics is certainly the most interactive approach, for it assumes the form of a painting interface. In all others systems, users are only required to provide a very loose guidance.

light to our reflection, as in such cases the concept of "generative process" becomes ambiguous and may refer to several realities. It can indeed refer to the medium employed for the creation of the inputs (a plethora of thoughtful publications focusing on the compelling emulation or simulation of real-world artistic media) but also to the penholder's mind. The creation by example therefore easily mixes itself up with the question of *style capture*, and along with it the assistance by the computer (if not automation) of part of the artists' work at the conception stage of their creations. Some artists of very distinguishing styles, notably the ones whose artworks reveal their creator's implication with mathematics such as Escher's crystallography-reminiscent divisions of the plane or Pollock's fractal-like paintings, already saw their styles and computationally emulated (*cf.* Figure 9.5).



9.6. Drawing Correspondences with McCloud's Big Triangle [McC93]. In his illustrated essay *"Understanding Comics"*, Scott McCloud visualizes the space of all visual communication as a triangle in which each point represents a specific style. He also identifies two main forms of visual abstraction which, we believe, cannot be treated similarly. Since *pure abstraction* mostly involves shape and space manipulation, it can be performed thanks to input analysis and imposed knowledge alone. Conversely, as *iconic abstraction* directly refers to the depicted scene's meaning, it cannot go without user feedbacks to reveal the extent of its potential.

The will to get computationally close to some artists' fine touch also motivated much research aiming at in-depth understanding, prediction, and modeling of the *abstraction* concept. The artists' aptitude of privileging the visual details of certain subparts of an image at the expense of others, without altering its overall meaning or informative content represents one of the most formidable and enduring challenges in expressive rendering. Indeed, abstraction intuitively appears as entangled with the meaning of the representation it acts upon, and therefore requires its prior understanding before being applied to it. In his examination of the comic industry and art form *Understanding Comics*, McCloud distinguishes between two forms of abstraction: the iconic abstraction, and the pure abstraction [McC93]. McCloud notably analyzes the spectrum of the forms of visual communication, and conceptualizes it as a two-directional space traversed by way of one of these two abstraction processes. His representation space assumes the form of a triangle whose extremities correspond to different modalities of visual communication: the photo-realistic depictions whose understanding is conveyed thanks to their resemblance to the retinal images elicited by real visual stimuli; more stylized representations that are nowhere to be directly found in nature but still succeed in communicating a specific meaning; and finally, the picture plane that encompasses all art forms departing from both resemblance and meaning, and often falling into the category of abstract art. Navigating through this triangle corresponds to the stylistic modifications one could apply to a given scene, and finding ways to conveniently do so constitutes part of the

research dedicated to style capture. Back from the perspective of our own information triangle, we can establish correspondences between these stylistic processes and our information sources: while *pure abstraction* could potentially be handled by the sole input analysis or the enforcement of predetermined rendering rules (prior) iconic abstraction can only be left up to the user for it directly deals with the meaning of the produced imagery and therefore shall not be made automatic.

9.2.2 Augmenting Rather than Automating

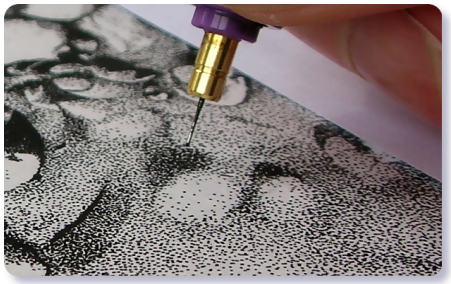
The fight for domination between automation and user interaction has been raging alongside this thesis, and as an obviously still open problem, reappears in its conclusions. While it may appear as a well-worn issue, the question of finding the correct level of automation is surprisingly barely taken into consideration in practice, and instead reveals two, rarely coincident attitudes towards the issue. Some researchers endeavor to provide strictly automatic methods, often at the cost of intricate and lengthy computations, and almost consider having users remain on the sidelines as the ultimate evidence of the efficiency of their techniques. Conversely, another approach consists in delegating any difficulty or hesitation encountered along the course of the algorithm – even when possibly removable by way of pure theoretical work – to users. Natural pitfalls are then a systematic resort to users and an undue emphasis granted to interfacing work. We believe that in some cases, research should proceed in a two-step gait: first, by trying to automate its computations as much as possible; second, at the light of the decisions that actually *should* be up to users, by providing them with handles to the higher-level representations that emerged from the input's pure analysis. Our work in texture synthesis, I believe, reached that critical point where part of the controls should be handed back to users for improvement.

This balance issue between artificial intelligence and user interaction – and also the fact the former in most cases take the upper hand – has been mentioned by Rob Cook during his 2009 Steven A. Coons award speech. He stressed the need to attach more importance to *interactive* techniques and expressed his belief in their capital role in the computer graphics research. Opposing "singularity believers"[1], Cook advocated a more careful examination of the possibilities offered by truly interactive techniques, notably for the creation of visual imagery. In his perspectives on the future of computer graphics, he also praised the potential usefulness of "illustration assistants" that would provide users with higher level controls over their creations thanks to the machine's artificial intelligence. This idea, along with his urge to trade the classical "engineer's model of art" for a more intuitive "artist's model of art", were explored during our line drawing generation technique which is conceived so as to constitute a "smart" drawing assistant to its conclusion.

Privileging automation over human intervention seems a rather reasonable choice every time subjective concerns such as aesthetics or expressiveness enter into consideration for the assessment of any method's success. This becomes especially true for most creation of visual content which requires designing skills. Interestingly enough, such questionings are not confined to the rendering domain, but pertain all computer graphics domains in which the resort to the computer resulted in the automation of tasks formerly handled by artists. An example is the recent worries expressed by part of the acting community that fears that an always facilitated resort to digitally created characters and re-mapped motion capture data might represent a threat to its craft. Such concerns have been very recently rekindled with the release of James Cameron's *Avatar* feature movie. Never before was the presence of virtually created content – from environments and effects, to creatures and protagonists – so screen-invasive. The imaginary planet of Pandora, its fantastic bestiary, as well as its luxurious landscapes definitely became one additional character of the movie, and quite stole the spotlight from the peripeteia of the so-called actors. Hence several questions naturally arise: Are actors actually that necessary? And what kind of performance can one deliver once surrounded by green walls and with mocap markers all over? But according to directors, actors of flesh and blood are by no way endangered by the advent of technology, and make a clear distinction between reusable motion capture and performance cap-

[1] The *technological singularity* is a concept thoroughly discussed in Raymond Kurzweil's books *The Age of Spiritual Machines*, and *The Singularity Is Near: When Humans Transcend Biology*. It refers to a point in the human evolution where technology know a sudden and rapid growth, substantially altering our living conditions and causing unpredictable impacts on the society. Kurzweil sees the technological singularity as a natural consequence of the breakthroughs in genetics, nanotechnology and artificial intelligence.

Stippling,
Christine Farmer.



Pastel oil painting on glass,
Aleksandr Petrov.

Sand animation,
Ferenc Cakó

9.7 Beauty Born out of Hardship. Stippling, paint on glass animation, sand animation are all examples of unusual media of a daunting difficulty. Yet, the will and dedication of some artists make possible their transformation into artworks whose beauty is strengthened by their difficulty and ephemerality.

ture which requires all the skills of a professional to avoid the overplay or gesture exaggeration. Similarly, key framing (which is the computer-assisted animation technique closest to traditional hand-drawn animation) still occupies a favored place in top-level studios such as *Dreamworks* since only skilled artists can indulge to their creations the expressiveness automatic mocap fails to deliver.

9.2.3 "It's Clever, But is it Art?"

In his keynote speech at the *NPAR'09* conference, Pr. Brian Wyvill questioned the artistic nature of his fractal-based algorithm for the non photo-realistic depictions of images, and wondered whether such automatic results could constitute genuine artworks. More than a thoughtless joke, his interrogation reveals the identity crisis faced by part of the expressive rendering's scientific community, and constitutes one of the most painful self-interrogation possible: What is the actual usefulness of most of non photo-realistic rendering methods?

Computers have rightfully earned their rank of artistic tools. The possibilities they provide have reached a level of sophistication that finally enables the concretization of the artists' wishes. Just as photography at its time, computers have become the extensions of the artists' hands, and now allow the re-transcription of their subjectivity and sensitivity. But is it the same with current research in expressive rendering?

Recent expressive rendering techniques indisputably propose wealth of intelligence and inventiveness, and enable the progressively easier creation of visually stunning results. There is no denying of that. And still, most of these impressive techniques seem almost ignored by its target audience: the community of professional artists. Why is it so? Could it be because of the aforementioned will to automatize as much of the creation process as possible? Progressively departing from the sole alleviation of the tedious repetitive tasks, has expressive rendering research encroached on the artists' playground to a point that would hinder its widespread use? This questioning is quite related to the one mentioned thus far: the case-by-case discovery of a proper balance between the three different information sources an example-based can consider. But the question does not only involve the quality of the results anymore, but instead concerns such methods' actual application.

When looking at artists' work, one observes their natural tendency of being rather unadventurous towards new media and techniques, leaving the fruit of the technological breakthroughs to a community of avant-garde artists. Once accepted and adopted, the said medium is then exploited to its very limits. This phenomenon can be observed on all media, from traditional art forms (*cf.* Figure 9.7), to computer-assisted ones (*cf.* Figure 9.8). It obviously attests the artists' demand for control over any medium so that they could add their distinctive fine touch to their creations. The audience also happens to be especially and extremely positively sensitive to such artistic gaits, to the point providing a care close to craftsmanship is by no way limited to the production of works only found in art galleries, but can sometimes be witnessed in commercial products such as video games (*cf.* Figure 9.9).

Could artists be reacting as some actors did, thinking that technology is trying to replace them rather than

Finger painting on *Apple iPad*,
Kyle Lambert, 2010.

Pixel art,
Yuriy Gusev, 2010.

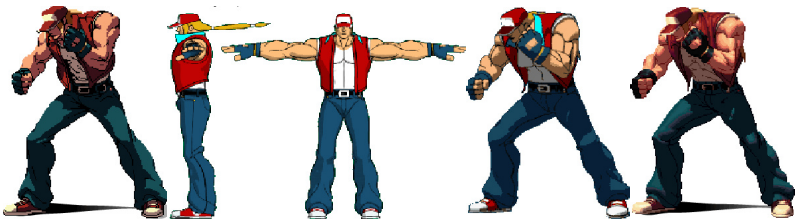
Vector art,
Mel Marcelo, 2008.

9.8 Beauty Born out of Hardship (Continued). From Figure 9.7, one could imagine that it is the tangible nature of artistic media that drove artists to go through such difficulty for creating their artworks. But this tendency is as visible in computer-assisted creation. The most emblematic example is doubtlessly pixel art which in spite of the technological advances of virtual canvases, consists in the careful coloring of each individual pixel, often with a voluntarily constrained palette.

A Scanner Darkly,
Richard Linklater, 2005.

Odin Sphere,
Vanillaware, 2007.

The King of Fighters XII, SNK Playmore, 2009.



9.9 Taking the Hardest Path. It would also be wrong to think that such meticulosity is only reserved to artists' portfolios. Recent productions involving high production budgets showed a similar sense of artistry and display a very distinctive, lavish appearance. The case of video games is particularly interesting as such artistic endeavors blur the frontier separating "commercial" and "pure" forms of art. Such efforts are especially laudable as they often come at the price of increased production costs.

facilitate their work? Facts would tend to say so, as for now only tools very similar in both aspect and control to traditional media seem to be in the professional artists' good books. Among them, virtual canvas softwares such as *Adobe Photoshop* or *Corel Painter* naturally take the king's share of the computer applications that made their way to the artists' hands. Techniques and softwares dedicated to vector illustration and animation (*e.g.* *Adobe Illustrator*, *Inkscape*, or former *Macromedia Flash*) already know a more moderate albeit fairly respectable success while already beginning to depart from the brush use metaphor and getting more revealing of their true computational form. But leaving the case of 3d modelers aside, very few techniques aiming at the *creation* of non photo-realistic content enjoy a widespread use. Similarly, expressive *rendering* methods also seem to experience a similar fate, with only the simplest of them (*e.g.* cell shading, or image-based contour detection) being integrated into professional and amateur productions. This is all the more disconcerting as an obvious interest for the expressive rendition of 3d is growing as shown by recent short movies or video games cinematics.

If the demand is truly here, maybe it is the supply that is unadapted to its audience. If we consider the issue from the perspective of industry, we observe a noticeable difference in the objectives and quality assessment of their products. Upstaging algorithmic intricacy, users' wishes and needs draw all the attention and the "goodness" of a method is often directly dependent on its quickness. Taking full advantage of modern computers' processing speed, such techniques embrace Cook's urge of exploiting the possibilities offered by interactivity, and count on users to overcome the issue of generality that plagues any fully automatic method. But focusing on interaction and celerity alone cannot possibly constitute fully satisfactory answers, for research is mostly fueled by the intellectual challenge. Finding new ways of interaction, and purposefully choosing the appropriate moments for resort to it seem nonetheless to represent wise principles to follow in order to ensure the adoption by artists of the methods proposed by subsequent research.

Appendices

APPENDIX A

Survey of Color Specification Systems

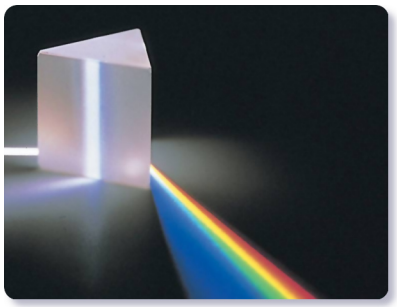
Since the main objective for Part II is to devise a improved color to grey-scale conversion, its fundamental theoretical core lies in the understanding of color perception. Isolating relevant perceptual effects, characterizing them and finally finding computational ways to account for these during the conversion process are keys to achieve satisfactory results. As such, the proper introduction of the terms used in colorimetry and the research devoted to their comprehension are the first mandatory steps towards a suitable problem statement. We hence decide to propose this appendix dedicated to colorimetry, which is to introduce the basic concepts behind nowadays' computational handling of colors, and disambiguate a terminology that does not tolerate much approximation. Superfluous for readers who are confident in their expertise of the field, it is mostly aimed at curious readers.

A.1 When Light Meets the Eye...

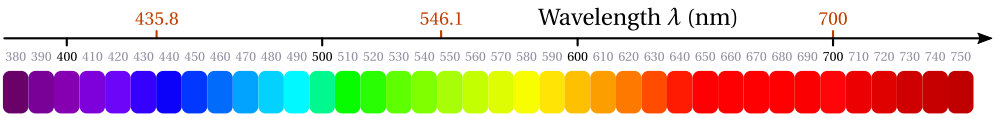
If we consider the problem of color perception in its entirety, dwelling on the nature of light and the rudimentary functioning of the eye is not superfluous as advanced perception models strive for modeling intricate phenomena occurring at the earliest stages of human vision.

Contrary to its simplifying ray-modeled counterpart in geometrical optics, light from a physical optics standpoint is an electro-magnetic radiation characterized by a wavelength λ. We are here naturally mostly interested in light our eyes are sensitive to, thus restricting our concern to a restricted category of radiations of wavelengths spanning the $380nm$-$770nm$ range (*cf.* Figure A.1). While the visible spectrum may appear fairly limited, the human eye nevertheless accounts for most of the radiations emitted by the stars, and most importantly the Sun.

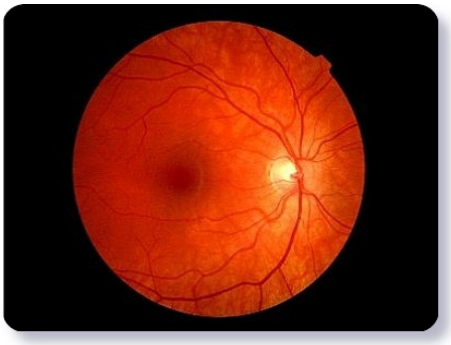
Newton first demonstrated the compound nature of white light that he splits into an infinite number of colored lights using a prism (*cf.* Figure A.2). The resulting spectrum appears as the juxtaposition of seven dominating colored lights, and Newton further strengthened his demonstration by re-combining these seven color distinct components into the original white light. The prism does not have any influence on the *color* of the lights traveling through it as proved by similar experiments involving individual light components taken separately and left unchanged by its action. White light is hence referred to as a *polychromatic* light, whereas its indivisible color components are *monochromatic* lights. The effect of the prism on white light is only due to its high density that slows down radiations of shorter wavelengths, then increasing diffraction and finally continuously separating the different colors.



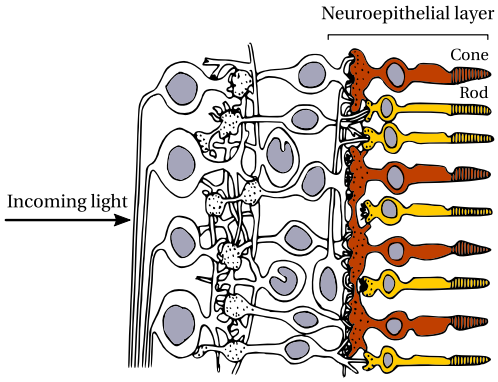
A.2 White Light Split into a Spectrum.



A.1 Visible Radiations and their Associated Wavelengths. Squeezed between ultra-violet and infra-red radiations, the visible portion of the spectrum is quite restricted. Nonetheless, the number of perceivable colors is unlimited and studies estimate that a healthy eye can distinguish about eight billions of colors. The wavelengths displayed in red correspond to the monochromatic lights employed during color matching experiments conducted by the CIE in 1931.



A.3 Retina in Middle Age. Cradle of human vision, the retina generates nerve impulses once hit by incoming light that are then conveyed to the brain for further processing. The optic disk *(lighter spot)* from which radiates all veins reveals the exit of the optic nerve towards the brain. The fovea centralis *(darker spot)* corresponds to locations of most detailed vision.



A.4 Structure of the Retina [DB66]. By observing monkeys' retinae stained with lead citrate through electron microscope, Dowling and Boycott stumbled upon an intricate set of cells. The *rods (yellow)* and *cones (red)* are the cells granting us vision and properly modeling and predicting their behavior is at the core of research in colorimetry.

A.5 Retinal Neuroepithelial Layer. This photograph obtained via a scanning electron microscope reveals the interweaving of the ubiquitous *rods (yellow cells)* and the sparser *cones (red cells)* responsible for the achromatic and chromatic vision respectively. It also illustrates the direct connection between their names and shape.

Light propagates, scatters, is eventually absorbed, reflected or transmitted by objects and when reaching the eye yields our perception of the scene. Eyes are hence easily comparable to photometric receptors whose photographic films would be the retina (*cf.* Figure A.3). It is the photochemical reaction caused by the contact of the incident light that passed through the cornea and lens with the retina that dictates our visual perception of the outside world. An object's color hence corresponds to the radiations that are reflected by its surface and impinge on the viewer's retina. By observing the microscopic structure of monkeys' retinae, Dowling and Boycott identify up to seven synaptic relationships between their constitutive cells [DB66] as illustrated Figure A.4, and among them, discover the *rods* and *cones* cells. Weaved together to constitute the retina's neuroepithelial layer, these cells are at the very heart of visual perception (*cf.* Figure A.5).

Three kinds of cone cells differing in the photo-sensitive proteins they contain exist, each of them reacting to radiations of distinct wavelengths. Cones are responsible for our perception of color, whereas rods which present a much more acute sensitivity to light ensure the perception of luminosity. All convert light into nerve impulses that leave the retina in direction to the brain through the optic nerve, and are then passed to the different areas of the visual cortex for spatiotemporal integration and ultimately the understanding of the different visual stimuli which entered the cornea.

A.2 Characterization of the Eye's Brightness Responsivity

As a modeling matter, the rough "functioning" of the human eye can be evaluated by finding ways of quantifying the overall responsivity yielded by the retinal cells' reactions when considered as a whole. For any photoreceptor, *responsivity* is theoretically defined as the receptor's output divided by the input radiant energy, and

in the specific case of the eye, the only possible output to assess is the *sensation* of brightness elicited by a specified visual stimulus. Since we talk here about highly subjective concepts such as "visual sensations", an appropriate experimental set-up for measuring such percepts is the stimulus matching framework. Brightness matching consists in using a predetermined monochromatic reference light of known wavelength and having its brightness adjusted by observers so as to match with that of an arbitrary test light. A brightness match is established once the sensations caused by the two stimuli can be merged according to the observer in terms of brightness. Once done, the radiant energy of both the reference and the test lights are recorded, and their ratio (representing a measure of brightness per unit radiant energy) constitutes the responsivity value of the observer's eye to the test light's wavelength.

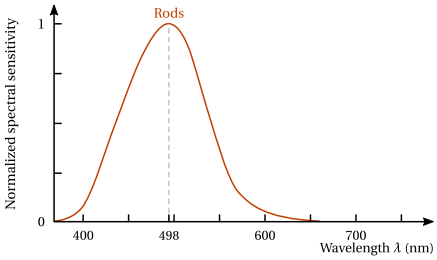
In order to get a full figure of merit for the eye, the experiment must be performed for test lights of various wavelengths. The ensuing responsivity curve, expressed as a function of the test light's wavelength, is then normalized by its maximal value in order to yield the *spectral luminous efficiency* of the eye $V(\lambda)$. Experimental issues arise when matches involve lights of too distinct wavelengths as the difference in terms of color biases the brightness comparison. The solution advocated by the CIE (*Commission Internationale de l'Éclairage*) is to rely on the "flicker method". Indeed, chromatic differences and brightness disparities between two colored lights become unnoticeable when the flickering frequency exceeds $30Hz$ and $50Hz$ respectively. Exploiting the existence of these two separate thresholds, the flicker method typically resorts to frequencies between $30Hz$ and $50Hz$ in order to discard distracting color shifts while keeping the brightness changes.

Naturally, the established spectral luminous efficiency curves are valid only for the observer that underwent the experiment. But many other factors also restrict the field of validity of these measurements and hinder their careless generalized use: among these, the width of field of view with which the observer perceives the stimuli, or how misaligned with respect to the fovea the stimulus hits the retina. But the most decisive experimental parameter is the illuminating conditions under which the experiment is conducted. Indeed, illumination has a tremendous impact on human vision as explained by the different levels of sensitivity between rods and cones. Rods are indeed sensitive to lights whose strength can be a hundred times lesser than the minimal strengths for which cones react. In bright illuminating conditions, the reaction of the rods gets saturated and they become inactive, leaving way to the cones that are the only photo-receptors ensuring vision. When cones too are made inert because of too strong incoming light, discomfort ensues. Under dim illuminating conditions, the cones are barely reactive to the limited luminous flux, and the much more sensitive rods take precedence again. The former case is called *photopic vision*, whereas the latter is referred to as *scotopic vision* both states corresponding to luminance levels above $3cd/m^2$ and below $0.003cd/m^2$ respectively. The in-between state for which both kind of photo-sensitive cells work in pair corresponds to the *mesopic vision*. This direct dependence between the functioning of the eye and its surrounding lighting environment is to be accounted for during brightness matching experiments since rods and the three different categories of cones present sensitivity peaks at shifted wavelengths (*cf.* Figures A.6 and A.7). Therefore, the CIE has established two spectral luminous efficiency curves for the two extremal viewing conditions, back in 1924 for photopic vision $V^*(\lambda)$ and in 1951 for scotopic vision $V(\lambda)$ (*cf.* Figure A.8). It should be noted that similar measurements for mesopic vision have also been made and that they highlighted a responsivity behavior that cannot be predicted by way of direct interpolation between $V(\lambda)$ and $V^*(\lambda)$ [ST86]. However, these additional luminous efficiency curves are yet to be validated by the CIE in order to constitute international standards [SS08].

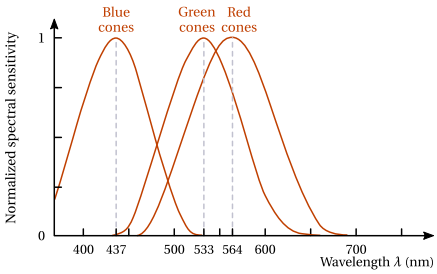
A.3 Exploiting Experimental Perceived Brightness

The careful examination of the two spectral luminous efficiency functions – coined *standard photometric observers* since their adoption by the CIE – is of capital importance in our context since they could be considered as sound brightness predictors. Besides, as functions of wavelength, they naturally account for the possible correlations between the stimuli's colors and their perceived brightness.

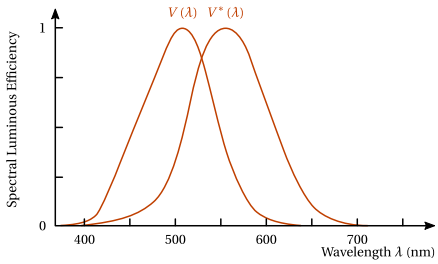
However, several hurdles prevent their straightforward application as a practical color to grey-scale converter. A first difficulty comes from the direct dependence of the eye's luminous efficiency functions on the



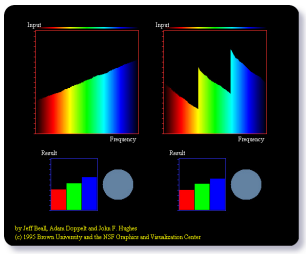
A.6 Spectral Sensitivity Function of a Rod Cell.



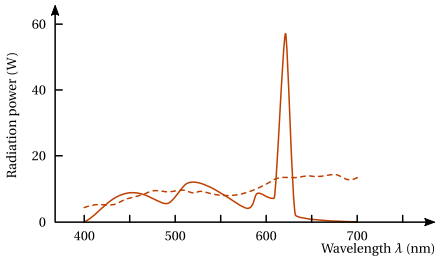
A.7 Spectral Sensitivity Functions of the Cone Cells.



A.8 Luminous Efficiency Functions for Scotopic and Photopic Vision.



A.9 Artificial Metameric Color Matching.



A.10 Metameric Skin Color Matches. *(solid: actual color, dashed: TV color)*

surrounding illumination conditions. Not only the existence of the two separate CIE-approved curves prevents the conception of an unified approach, but also the triggering between the scotopic and photopic vision conditions would not constitute a trivial task without user notification. Besides, there is no standardized photometric observer for mesopic vision because of the complex evolution of the relative contributions of the different photo-sensitive cells with the conditioning luminance. Properly capturing this evolution would require many experiments at different luminance levels, and many already proposed brightness responsivity curves exhibit a non additive behavior. This is actually detrimental for the practical use of such functions as the overall brightness of mixtures of monochromatic lights (lights whose spectral color distribution is not a Dirac) becomes unpredictable.

In addition, *metamerism* also hinders us from directly using established spectral efficiency functions. Discovered by Wilhelm Ostwald, metamerism designates the matching of perceptually equivalent colors that in reality exhibit distinct spectral power distributions. Is provided Figure A.9 an illustration example that has been obtained by using the applet coded by Baell, Doppelt and Hugues from the Brown University. The fact cone responses do not uniquely characterize a complete spectral power distribution is due to the fact the eye has only three color-dedicated cells at its disposal, each of which integrating luminous power over a broad range of wavelengths (*cf.* Figure A.7). Different combinations of radiations can consequently yield the very same sensory response, and cause a metameric color match. Metameric matches are especially frequent for duller and darker colors, and can represent an actually complicated issue for industries whose color tolerances are low.

Examples include the making of paints, dyes, fabrics, but also quality printing for which papers are often enhanced with brighteners. In the case of object colors, the difficulty is increased by the fact perceived colors depend on the light sources illuminating these objects and in a slight abuse of language, metamers often refer to objects whose apparent color seem sensitive to changes in illuminant. This phenomenon is also subject to the field size of observation and naturally the observer's cone sensitivity functions. While being a burden for the making of specifically colored material, most color reproduction techniques such as photography, television or even lesser quality printing heavily rely on metameric color matches in place of spectral color matches. As a result, the spectral reflectances of an object and of its reproduced counterpart usually greatly differ.

A.4 Specification of Colors and Principal Color Spaces

Although spectral luminous efficiency functions are essential tools for the conversion of radiometric quantities into their photometric counterparts, they cannot be straightly used in our case. When considering the application scenario of our technique, inputs consist of digitalized color images without any assumption about the means of their acquisition. A first step towards the handling of the color information they convey is to investigate existing color specification systems – namely color spaces.

Color specification aims at finding means for the quantitative expression of colors, and once all visible colors have been expressed in a comparable and consistent manner, enables the establishment of common rules and formulae. Two main categories of color spaces co-exist: color appearance systems, and color mixing systems. While the former historically appeared before the latter, the order of presentation is here reversed in order to better reflect nowadays' usage of such spaces.

A.4.1 Color Mixing Models

Color mixing models include some of the most commonly used color spaces, many of them have long been certified by the CIE and thus served as international standards for color specification since then.

A.4.1.1 Additive Color Mixing and Grassmann's Laws

Color mixing models describe colors by the amounts of mixed color lights necessary to obtain a color match. Such a specification approach assumes the fact that any desired light color can be reproduced by the mixing of a set of light color primaries to a certain extent. Naturally, one cannot pick random color primaries if one wants to specify all visible colors that way. In accordance to the wavelengths associated with the different cones' sensitivity peaks, primaries always incorporate red, green and blue monochromatic lights [R], [G] and [B] respectively. They constitute the reference stimuli with respect to which all other colors will be defined. The stimulus elicited by any colored light can therefore be approximated by the stimulus induced by the interaction of these reference stimuli with appropriate mixing ratios.

Here, visual stimuli are expected to respect Grassmann's empirical laws of proportionality and additivity, hence complying to additive color mixing. Roughly speaking, Hermann Grassmann proved through color matching experiments that any color sensation, not necessarily emanating from monochromatic lights, can be matched against the linear combination of different light colors. If considering the set of previously chosen primary stimuli, Grassmann's law can be stated as follow:

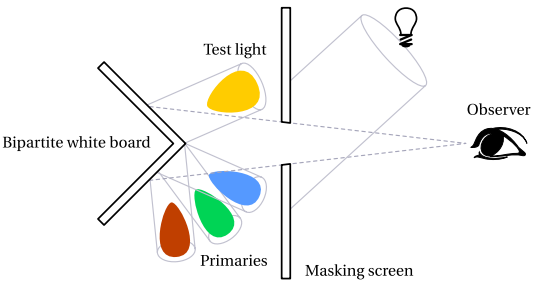
$$[\mathsf{F}] = R\,[\mathsf{R}] + G\,[\mathsf{G}] + B\,[\mathsf{B}]$$

where [F] designates a potentially heterochromatic test visual stimulus, the = sign means a color match established by visual test and (R, G, B) corresponds to the set of radiant powers assigned to the reference stimuli to establish the match. Close to all color spaces used nowadays actually stem from Grassmann's research conducted during the 19th century.

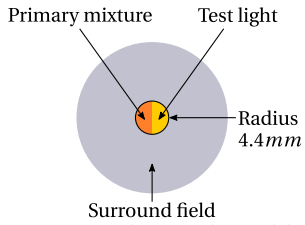
A.4.1.2 CIE 1931 RGB Specification System

Via Grassmann's theory, describing colors with respect to a fixed set of primaries can therefore be achieved by conducting direct color matching experiments involving monochromatic lights. Contrary to color appearance systems described later on, the influence of the surroundings and of the lighting conditions are ignored. In such cases, we talk about color *sensations* in place of color perception as the latter term usually implies that the surroundings' contribution has been considered. The experimental set-up must therefore discard the perceptual effects of the surroundings, and for that aim, color stimuli often consist of a uniformly colored area observed through a dark aperture. Figure A.11 gives a schematic view of such color matching experiments.

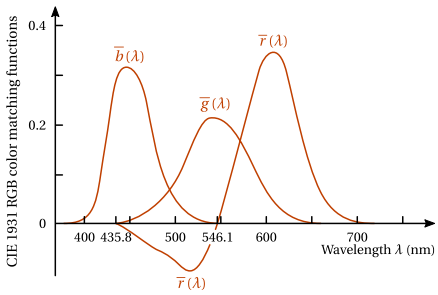
Through the aperture and as observed by the test subject, the complete stimulus is made of two directly adjacent colored halves corresponding to colored lights projected onto a perfectly reflective white board (*cf.*



A.11 Upper View of the Visual Color Matching Apparatus.



A.12 Stimulus with Field of View of 2° Diameter.



A.13 Color Matching Functions for the CIE 1931 RGB Color Space.

Figure A.12). The stimulus's first half emanates from the test light whose perceived emitted color is assessed. The second half results from the additive mixture of the primary lights. The reference stimulus's radiant powers (R, G, B) are then tuned by observers until its both sides become undistinguishable. The *chromaticity coordinates* of the monochromatic test light are then defined as:

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B}.$$

Since the relationship $r + g + b = 1$ between coordinates always holds, couples of the form (r, g) are sufficient to unambiguously refer to a given color.

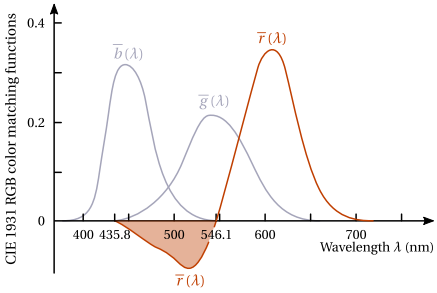
By using monochromatic test lights and having their wavelength sweep the visible spectrum, one can describe chromaticity coordinates as functions of wavelength, and trace the observer's *spectral color matching functions*. Chromaticity coordinates are naturally directly dependent on the chosen primaries' wavelengths, and the color specification would be incomplete if these are not defined and known. The CIE advocated the use of Wright's and Guild's experimental data as an international standard. The reference stimuli thus correspond to the following wavelengths $\lambda_{[\mathsf{R}]} = 700nm$, $\lambda_{[\mathsf{G}]} = 546.1nm$ and $\lambda_{[\mathsf{B}]} = 435.8nm$, and the reference white results from the following mixture: $1.0000\,[\mathsf{R}] + 4.5907\,[\mathsf{G}] + 0.0601\,[\mathsf{B}]$. Stimuli were observed through a 2° field of view as illustrated in Figure A.12. Wright conducted his experiments between 1928 and 1929 while Guild's occurred in 1931, and both involved respectively seven and ten test subjects with normal vision. Their *averaged* results, presented in Figure A.13, since then constitute the color matching functions of the CIE1931 RGB color specification system.

A.4.1.3 CIE 1931 XYZ Specification System

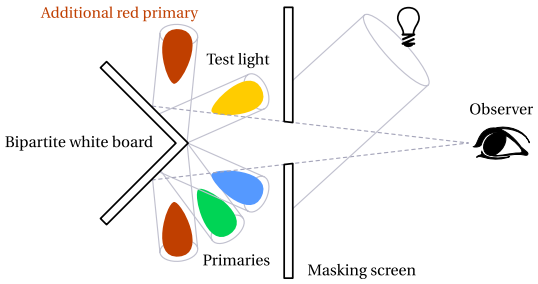
Even today, the CIE 1931 RGB specification represents the most widely used color specification system, but this "monopoly" came at the price of a slight alteration.

A first inconvenience of the original RGB specification is the presence of a negative portion in its red color matching function $\bar{r}(\lambda)$ that unnecessarily complicates computations (*cf.* Figure A.14). This aspect of the specification is also highly non-intuitive: this curve describes the normalized evolution across the visible spectrum of the red primary's radiant power.How could such a radiometric quantity be found negative? Cases associated with negative red chromaticity coordinates actually correspond to experimental cases where an additional monochromatic red primary light had to be used in conjunction with the test light to make the color match possible (*cf.* Figure A.15).

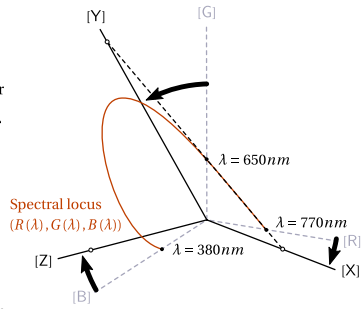
The CIE 1931 XYZ specification corresponds to a basis change in ways all three spectral curves always exhibit positive values. Its unveiling hence boils down to finding an appropriate linear transform that rotates the original $([\mathsf{R}], [\mathsf{G}], [\mathsf{B}])$ stimulus basis into a new one $([\mathsf{X}], [\mathsf{Y}], [\mathsf{Z}])$, and maps the $\left(\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda)\right)$ functions to positive functions $\left(\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)\right)$. Other requirements were employed to restrict the range of acceptable transforms and improve the specification's meaningfulness. One these, the $\bar{y}(\lambda)$ function is made to coincide as closely as possible to the spectral luminous efficiency for photopic vision $V^*(\lambda)$. The resulting



A.14 Negative Portion of the CIE 1931 RGB Red Matching Function.



A.15 Modified Experimental Set-up for Color Matching. In cases where test lights appear too dim or primary mixtures too bright, an additional red primary light is added to the test light to enable matching.



A.16 Graphical Interpretation of the CIE 1931 RGB to XYZ Conversion.

Y-coordinates then conveniently encode photometric quantities, and the fact colors of a same Y-value can be considered as inducing stimuli of a same luminance is a valuable feature.

The last, more technical constraint that fully fixed the conversion transformation was to make the line linking the [X] and [Y] output stimuli tangential to the $(R(\lambda), G(\lambda), B(\lambda))$ spectral curve of monochromatic colors at $\lambda = 650nm$. A graphical interpretation of the conversion process is given Figure A.16. The desired transformation is henceforth defined as:

$$X = 2.7689R + 1.7517G + 1.1302B, \qquad Y = 1.0000R + 4.5907G + 0.0601B, \qquad Z = 0.0000R + 0.0565G + 5.5943B. \tag{A.1}$$

Since the conversion consists in a linear operator, Grassmann's laws remain valid and color specification follows the same scheme as before:

$$[\mathsf{F}] = X\,[\mathsf{X}] + Y\,[\mathsf{Y}] + Z\,[\mathsf{Z}].$$

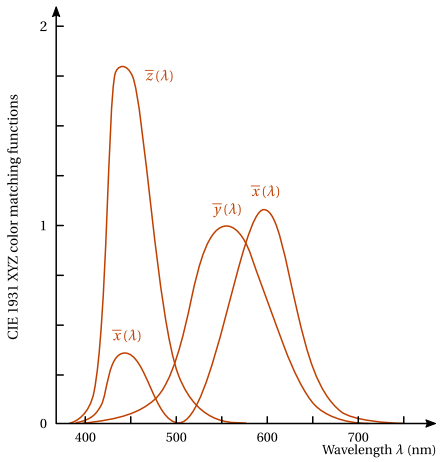
The set of coefficients (X, Y, Z) is called the *tristimulus values* of the test stimulus [F].

Defined with respect to the new reference stimuli ([X], [Y], [Z]), averaged color matching functions now always assume positive values (*cf.* Figure A.17). According to Equation A.1, [Y] now corresponds to the CIE 1931 RGB specification's reference white, whereas [X] and [Z] theoretically denote imaginary colors devoid of luminance. Chromaticity coordinates are defined as the normalized tristimulus values:

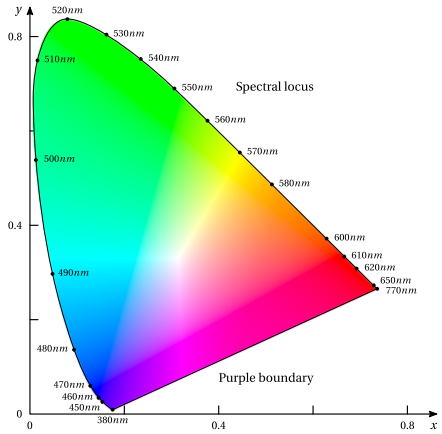
$$x = \frac{X}{X+Y+Z}, \qquad y = \frac{Y}{X+Y+Z}, \qquad z = \frac{Z}{X+Y+Z}, \qquad \text{with } x+y+z = 1,$$

and in spite of its slight redundancy, the (Y, x, y) notation is often encountered to specify colors.

By displaying all colors over a 2d plane according to the (x, y)-parametrization, one obtains the *chromaticity diagram* for the CIE 1931 XYZ specification (*cf.* Figure A.18). Connecting the coordinates established for all monochromatic lights results in the *spectral locus*. By closing the curve between the coordinates associated with the two extremal monochromatic lights (of wavelengths equalling to $380nm$ and $770nm$), one draws the *purple boundary* whose colors continuously vary from blue to red by assuming various shades of purple. All real colors obtained through the mixture of monochromatic lights have their coordinates falling into the closed region of the diagram delimited by the spectral locus and the purple boundary. For applicability purposes and due to the generalized used of the CIE 1931 XYZ specification, close to all subsequent color spaces propose conversion formulae to-and-fro the XYZ specification. Therefore, the shared links between these new color spaces and the XYZ specification find graphical interpretations over the chromaticity diagram such as the hue selection of the Coloroid color space for instance.



A.17 Color Matching Functions for the CIE 1931 XYZ Color Space.



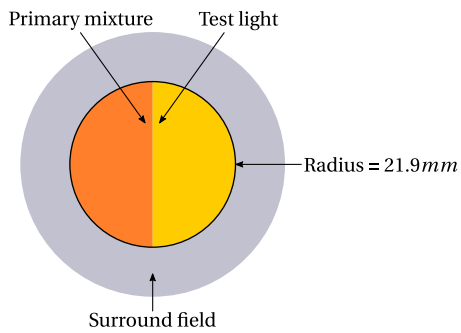
A.18 CIE 1931 XYZ Chromaticity Diagram.

A.4.1.4 Refining upon the CIE 1931 XYZ Specification

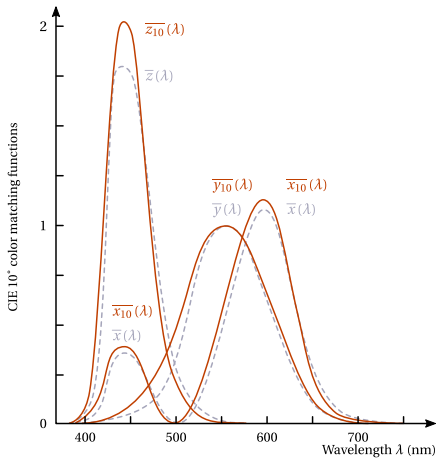
Although widely used, the XYZ specification proposed by the CIE still has a certain number of shortcomings. These mainly arise from the experimental decisions made during the visual color matching tests. Naturally, no additive color space can pretend to account for the entirety of our color perception because of the highly complex nature of human vision, its subjectivity and the intricacy of the mechanisms it relies on. By lifting previous assumptions, identifying new phenomena and conducting further experiments, the understanding of color perception is a step-by-step process, and since its origin, several incremental refinements have been provided to the initial CIE 1931 XYZ color matching functions $\left(\overline{x}(\lambda), \overline{y}(\lambda), \overline{z}(\lambda)\right)$.

Towards Wider Stimuli... A first concern about the viability of the generalized use of the original XYZ specification comes from the fact it has been established by matching colors observed with a field of view of 2°. Indeed, further observations tend to prove the original color matching functions can hardly be directly used to estimate the perceived colors of stimuli corresponding to larger fields. Nevertheless, the choice of resorting to viewing angles of 2° is neither random, neither irrelevant as it corresponds to stimuli hitting the retina at its foveal region which corresponds to the retinal location of highest visual acuity and maximal cone cell concentration. But the presence at this very place of the macular yellow pigment greatly influences the spectral matching functions established for stimuli impinging this specific spot. Therefore, previously estimated color matching curves become unadapted for wider stimuli. This is experimentally observed by the appearance of the Maxwell spot at the center of the visual field during color matching experiments involving stimuli observed under greater viewing angles. This phenomenon remains imperceptible in daily life thanks to the eye's adaptation. In 1964, newer set of experiments conducted in parallel by Stiles and Burch and Speranskaya, involving 49 and 27 individuals respectively, enabled the estimation of color matching functions valid for visual fields of 10°. The main difference between the two sets of experiments is the handling of the Maxwell spot, Stiles and Burch neglecting it, Speranskaya hiding it. These revised functions are noted $\left(\overline{x_{10}}(\lambda), \overline{y_{10}}(\lambda), \overline{z_{10}}(\lambda)\right)$ and the ensuing color system, named the CIE 10° Colorimetric System, is recommended to replace the CIE 1931 XYZ Colorimetric System when viewing angles exceed 4°.

Inter-Observer Variability Another theoretical limitation of both the CIE 1931 XYZ and CIE 1964 10° specifications comes from the *averaging* across observers of the matching functions, and hence the loss of the variations between test subjects which result from the metamerism phenomenon. In 1989, the CIE reconsidered the matching results gathered by Stiles and Burch in 1959 and incorporated the variations between the 20 subjects to a new set of color matching functions $\left(\Delta\overline{x}(\lambda), \Delta\overline{y}(\lambda), \Delta\overline{z}(\lambda)\right)$ modeling the deviations to the standard reference observers of the CIE XYZ 2° and 10° specifications. These tabulated deviation functions account for several factors such as the variations between observers, metamerism and age dependence.



A.19 Stimulus with Field of View of 10° Diameter.



A.20 Color Matching Functions for the CIE 1964 10° Color Space.

A.4.2 Color Appearance Models

Color appearance models correspond to the second principal color specification approach. Contrary to color mixing specification systems that rely on the visual matching between colored *lights*, they usually involve colored *materials* and are hence defined with respect to these. As a practical consequence, their gait and results are easier to apprehend. Their direct reliance to a tangible support, typically color pigments or colored tiles, explains the disparate sources of research and variety of theories that emanated from physicists to artists to manufacturers.

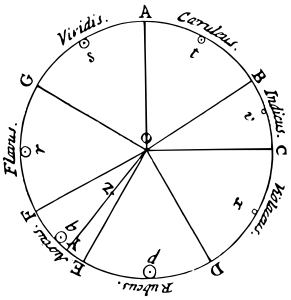
A.4.2.1 Early Stages of Colorimetry and First Models

Devising color theories is actually much anterior to the formalizations proposed by the CIE, and anticipated nowadays' findings in human vision and technology allowing scientifically sound experiments. First efforts at organizing colors and displaying palettes in a schematic fashion under the form of tables or charts can be traced back to Middle-Age pictorial art. Obvious first examples include painters' endeavors to state rules and conventions on harmony between colors and visualize the range of the tints at their disposal by experiencing with the mixing of specific pigments. But this long lineage of *ad hoc* color theory knew a renewed interest in 1966 with Newton's experiment (*cf.* Figure A.1). Formidable by the scope of his results, Newton also impresses by the scientific rigor he shows for the conduction of his experiment which marks the birth of colorimetry as a full-fledged scientific field. And as a new-born science, many debated and contradicted Newton's and Hyugens' formalization of light as a wave whose length conditions its perceived color once it entered the eye. It is interesting to see examples aiming at invalidating Newton's theory according to which white light is made out of all possible colors, that were created using paints such as Jacques Gautier d'Agoty's color bar in Figure A.22.

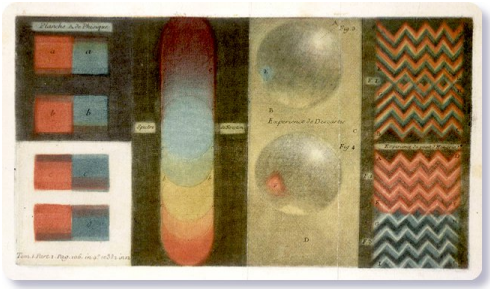
The most critical attack formulated against Newton is the demonstration proposed by Johann Wolfgang von Goethe in his *Color Theory* published in 1810. Alas, Goethe's conclusions stem from experimental approximation and misunderstandings of Newton's results. Goethe considers color not as the consequence of the shape of its carrying light wave, but as the interaction between light and darkness. While defined by Newton as the absence of light, Goethe sees in darkness the polar opposite of light, and grants it an active role in the arising of colors. Yellow would be light tainted by darkness, whereas blue would be darkness weakened by light. These two colors hence constitute the primaries from which Goethe draws all other possible colors as illustrated by his color wheel in Figure A.23.

A.4.2.2 Making it Complete while Keeping it Simple

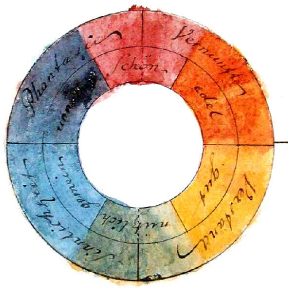
A key feature of most color appearance models is that regardless of the complexity of their underlying theory about the nature of light or the origin of colors, they need to be of practical use, and thus specify colors in an efficient, understandable way. In that sense, contributions by artists and painters gave valuable insights



A.21 Newton's Color Wheel.



A.22 Gautier's Contradiction to Newton's Light Theory.



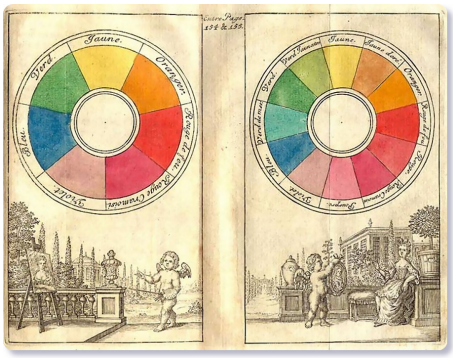
A.23 Goethe's Light Wheel.

about the visual characteristics of colors and how to possibly organize, and order them. Thanks to their skills and artistic sensitivity, they laid terms to the different controllable attributes of colors and also proposed their own takes for color specification. Conflicting theories of light and its debated containment of all colors gave rise to several color ordering representations, most of them taking the shape of hue-varying circles called *color wheels*. This continuous, self-reflexive representation is mainly Newton's formalization subsequent to his observation of white light's spectrum, and even though it was heavily questioned, many alternative theories keep on displaying colors according to a circular layout such as Goethe's color circle (*cf.* Figure A.23).

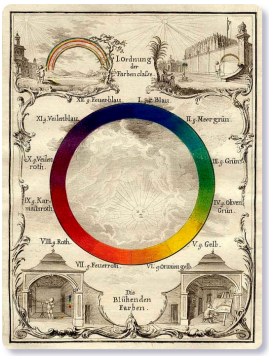
Hue Originally composed of seven primaries, Newton's color wheel is soon taught in painting treatise of his times, and eventually added tints (*cf.* Figure A.24). Many illustrations from such books show wheels made of twelve colors whose general layout even if more complicated still observes the respective importance Newton's gave to his seven dominant hues (*cf.* Figure A.25). Chromatic circles such as Schiffemüller's still relies on a nominative nomenclature to differentiate and designate colors. Such a detailed and subjective classification was still bearable due to the limited number of tints. Obviously, such color wheels do not incorporate all possible colors attainable either by the physical mixture of pigments, or even less by the mixture of lights. The only color attribute they account for is a correlate of *hue*. Although lacking in terms of completeness, such representations nonetheless constitute the earliest one-dimensional color scales.

Colorfulness The second visual attribute of colors to be embedded in such geometrical color representation is the notion of *colorfulness* or intuitively speaking, how vivid a given color appears. Moses Harris' two shaded wheels are among the first color appearance models to incorporate this aspect of colors. He further details hue variations by pinpointing eighteen distinct hues and adds a second dimension appearing as the distance to the wheel's center. Hue portions are then divided into crowns, the further a crown lies from the center, the duller it appears. Harris's first "prismatic" wheel stands for isolated colors and is built upon the red, yellow and blue colors as its primaries (*cf.* Figure A.26). His second "compound" wheel is obtained by mixing orange, green and purple and aims at the study of color contrasts commonly encountered in paintings (*cf.* Figure A.27). Among with the intuition to incorporate considerations of color relationships other than adjacency, Harris also introduced a numerical denomination of colors, going a step further towards more mathematical, objective means of color specification.

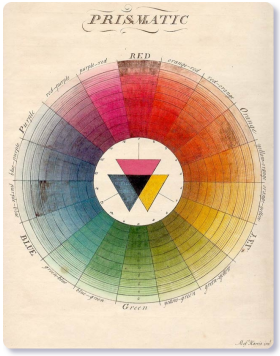
Luminosity In spite of their improvement in terms of color description, Harris' dual color models still miss a last component to enable the entire spanning of the range of all possible colors. This remaining component corresponds to the colors' *luminosity*, how relatively bright and dark they appear, and was introduced by astronomer Tobias Mayer in 1758 in his lecture entitled *"De affinitate colorum commentatio"*. His purpose was to illustrate and determine the number of colors the human eye could distinguish. Mayer claimed that human vision is intrinsically limited, and that very subtle color variations cannot be differentiated easily, if at all. He illustrated his allegation by choosing three primaries, red, yellow and blue (via paints made of vermillion, massicot and azurite pigments) and mutually mixing them according to a 12-step gradation after darkening or



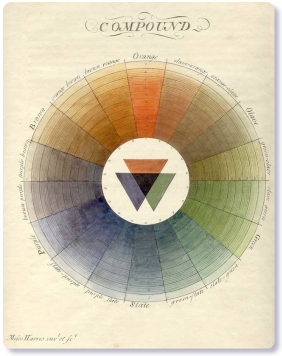
A.24 Boutet's Painting Treatise Improving over Newton's Color Organization. Source: *Traité de la peinture en miniature*, 1672.



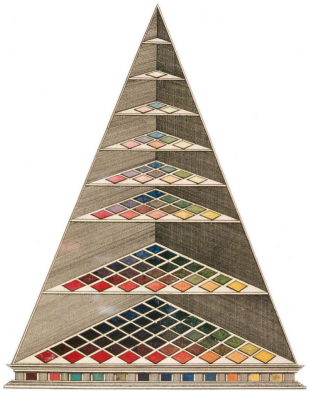
A.25 Schiffemüller's Depiction of Newton's Color Wheel. Source: *Versuch eines Farbensystems*, 1772.



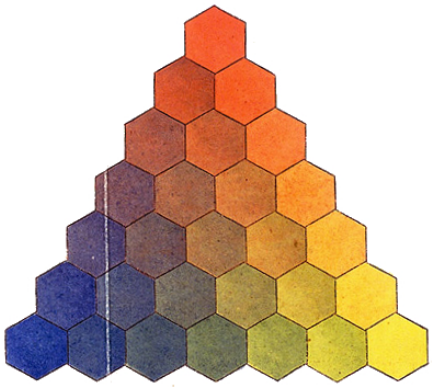
A.26 Harris's Prismatic Shaded Color Wheel. Source: The Natural System of Colors, 1766.



A.27 Harris's Compound Shaded Color Wheel. Source: The Natural System of Colors, 1766.



A.28 Mayer's Pyramidal Color System by Lambert.

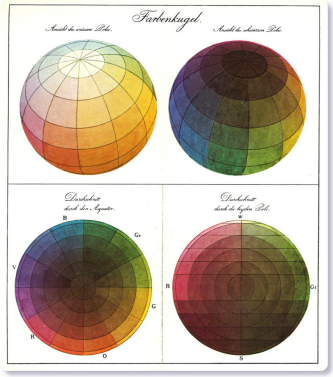


A.29 Lichtenberg's Rendition of a Triangular Section From Mayer's Pyramid.

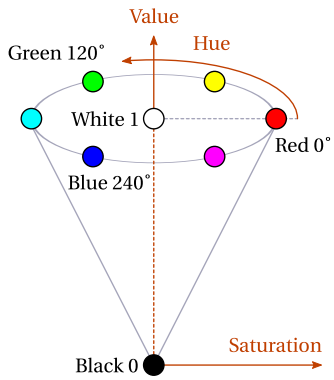
A.30 Real-Life Example of a Mayer's Triangle for Mixing Irgalan Dyes.

lightening using black or white paints. The choice of using twelves of primary colors is not fortuitous as Mayer suggested it was the minimal perceivable amount of change colors could undergo before ending up unnoticed. Painter Johann Heinrich Lambert executed some of the most faithful representations of Mayer's color pyramid, some of them being dated around 1772 as a *post-mortem* homage such as the one presented in Figure A.28. Even more years after Mayer's demise in 1762, physicist Georg Christoph Lichtenberg published a simplified version of Mayer's system consisting of a single triangular slice which illustrates the different mixing combinations between primaries according to a 7-step graduation only (*cf.* Figure A.29). In spite of its anomalies (such as over-darkening of its center colors when black is progressively added at the finest levels of the pyramid), Mayer's specification is still used nowadays, notably in the dying industry (*cf.* Figure A.30).

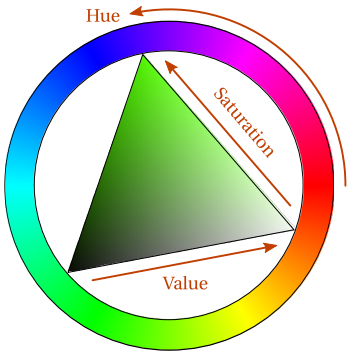
Mapping Colors to 3d Shapes Harris introduced chroma as an additional dimension independent from hue with his shaded wheels, while Mayer attached luminosity to the main axis of his pyramid. However, both systems still provide an incomplete color description. Mayer's representation is especially misleading as it appears as a 3d volume while its 2d cross sections actually only account for the hue changes caused by the uneven mixing of primaries. Encouraged by Goethe, painter Philipp Otto Runge exposed his theory of color, and proposed the first geometrical representation of a color space as a real 3d volume whose different dimensions encode the three attributes of color independently. Constrained by the circular topology of the hue attribute, Runge conceptualized his color space as a three-dimensional sphere whose parametrization controls the color



A.31 Runge's Color Sphere. Source: *Farbenkugel*, 1810.



A.32 Ray Smith's Hue-Saturation-Value Specification.



A.33 Typical 2d Representation of Ray Smith's Hue-Saturation-Value Model.

specification: the azimuthal angle determines hue, the radial distance describes the color chroma, and the polar angle fixes its luminosity (*cf.* Figure A.31). More than the blend between Harris' and Mayer's theories, Runge's endeavor to model the behavior of subtractive color mixtures pioneered nowadays' user-centric color spaces. Picturing the changing appearance of progressively mixed colors as trajectories in 3d volumes is therefore not novel at the light of the different color specifications proposed across history. Runge first identified the three basic attributes of color, namely its hue, chroma and luminosity, that he matches to the three components of a spherical coordinate system. But, the casting of color spaces into 3d shapes is by no way limited to spheres, nowadays' representations mostly involving cones such as Alvy Ray Smith's HSV conical system. Ray Smith follows his predecessors' steps and makes his color specification expressed in terms of hue, saturation (comparable to chroma) and value (analogous to luminosity). His specification innovates with respect to previous color models as it is not defined in terms of color mixing, but instead proposes an entirely mathematical framework relying on non-linear transformations. Created and published in 1978 while its creator worked at *Xeroc PARC* on the conception of the *SuperPaint* drawing software, the HSV model is now found in nearly all possible softwares involving user-specified colors, and its user friendliness earning Ray Smith a *SIGGRAPH* award in 1990 for his "seminal contributions to Computer Paint Systems".

A.4.2.3 About the Importance Attached to Perception

By the way they historically arose, color appearance models managed to specify colors according to various coordinate systems that may have varied in terms of geometrical complexity, but always remained associated with intuitive color-related notions. It is the empirical nature of the experiments necessary for the set-up of these systems (mostly driven by painters' work and sensitivity) that grants them such comprehensibility and therefore usefulness. Another crucial and appealing aspect of these models is the natural embedding of perception related considerations often lacking even in modern models obtained by visual matching of colored lights.

Because they rely on the visual inspection of colored materials (stains of color paints or small colored tiles) color appearance systems can only account for the visual impressions caused by the appearance of these materials in specific illumination conditions. The manipulation of actual physical entities during experiments, typically color chips, makes factually impossible the isolation of the visual test stimuli from all perturbing factors with the same ease and accuracy as when considering focused color lights observed through an aperture. For these reasons, color appearance specifications are subjective, and defined up to the illumination conditions under which the experiments were conducted.

One cannot possibly explain the subjectivity of color appearance models only as an experimental inconvenience, understanding human perception was actually an integrant part of the involved research. As his primary objective, Mayer strove to understand the perception of colors, and characterized notions such as visual discrimination and just-noticeable-differences between color mixtures in the process. Goethe also heav-

ily blamed Newton's wave-based light model and its colored manifestations for being too simplifying as it only consisted in a purely mathematical, phenomenological explanation which disregarded perceptual concerns. Therefore, color perception is at the heart of color appearance models not only as a matter of fact, but also as a scientific will. Great insights on human perception have been made explicit, quantified and embodied by color appearance models, to the point some of them have been subsequently injected into the CIE 1931 and CIE 1964 standard observer models.

A.4.3 Towards Perceptual Uniformness

A.4.3.1 Munsell Color System

The color specification proposed by American professor and artist Albert Henry Munsell falls into the category of color appearance model and is the first and maybe most successful attempt at perception-observing color sampling. While keeping the intuitive classification of color by its attributes of hue, chroma and luminosity. Munsell strives to accounting for our color perception and its direct incorporation by ensuring the perceptual uniformity of its dimensions. Munsell opts for a 3d semi-polar coordinate system but instead of relying on geometric distances that would have given his model a cylindrical form, strictly observes perceptually relevant distances which gives his model a more intricate shape (*cf.* Figure A.36). Munsell's perception-aware system had a tremendous impact on artists and manufacturers thanks to its intuitiveness, but also on color scientists for its solid theoretical ground. Indirectly, the Munsell color system shaped subsequent standard CIE uniform spaces such as the CIE $L^* a^* b^*$ and CIE $L^* u^* v^*$ colorimetric models whose validation data sets are made of color samples equidistantly spaced in the Munsell space.

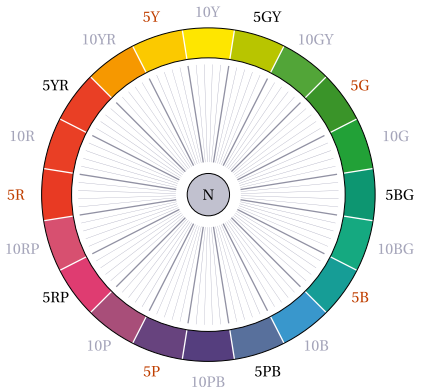
A scale is *perceptually uniform* if a small perturbation to its component value is roughly equally perceptible across the range of that value. Ensuring this property to his model's three independent components is the main goal behind the experiments conducted by Munsell in 1905, and revised after his death in 1929 and 1942 to profit from modern techniques' increased measurement accuracy for the estimation of the subjects' visual responses. The Munsell system is still commonly used nowadays, and is recommended for the specification of colored surfaces by the American Standards Organization.

The Munsell system is built from a color circle initially made of five dominant hues, red (R), yellow (Y), green (G), blue (B) and purple (P), disposed at equal perceptual distances from one another, and in such a way that the merging of opposing colors results in a neutral grey (N). Once determined and globally arranged, five additional hues created by the mixtures of the basic ones are also added to the system's hue circle: yellow-red (YR), green-yellow (GY), blue-green (BG), purple-blue (PB) and red-purple (RP). The process is iterated until a set of 100 hues is established, the continuity of the Munsell hue coordinate A being then guaranteed by interpolation.

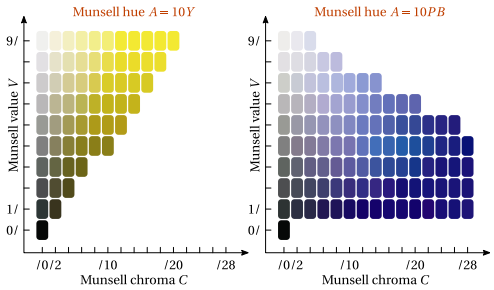
The Munsell chroma component C is attached to the radial distance to the central axis, basic hues and their five first mixtures being arbitrarily assigned a chroma of 5. It is an open scale experimentally established and directly dependent on the hue.

Lastly, the Munsell's correlate to luminosity, the Munsell value V, is associated with the vertical axis and varies from black to white in a 10-step gradation. As the $H-$ and $C-$ dimensions, these steps are carefully chosen to properly reflect perception variations in a way that the square root of the measured reflected intensity varies at constant steps. It should be emphasized that the lightness models we use for our grey-scale conversion algorithm directly inherits the Munsell value's perceptual soundness.

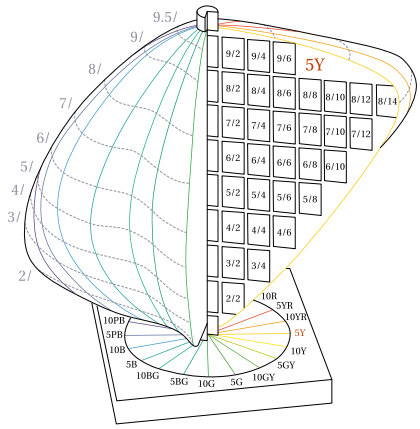
By trading geometrical simplicity (such as the symmetry of his model's shape) for psycho-cognitive meaningfulness, Munsell achieved a impressive leap towards perceptually-relevant color specification. Munsell's experiments consisted in having subjects asserting the perceptual equidistance between color chips surrounded by an achromatic area observed under the CIE C illuminant (average daylight in international standards). Two series of experiments were carried out involving around 1,550 chips and 1,250 chips for glossy and diffuse surfaces respectively. For practical use, tables between the color samples used by Munsell and their tristimulus values have been established since then, enabling the conversion between Munsell color coordinates, com-



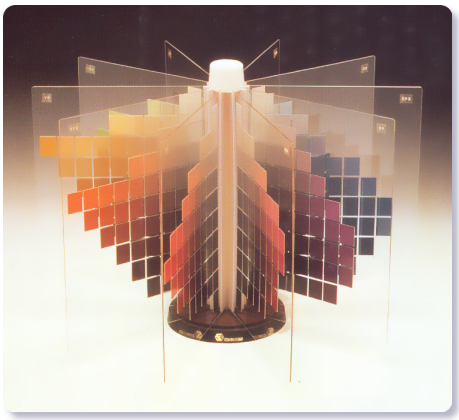
A.34 Munsell's Color Wheel at the Middle Value Plane.



A.35 Constant-Hue Vertical Sections of the Munsell Color System. Munsell attached a primordial importance to the perceptual relevance of the spacing between his color samples. As a consequence, colors sharing the same hue can exhibit completely different distributions in terms of chroma and value.



A.36 Schematic 3d View of Munsell's Colorimetric Specification.



A.37 Physical Representation of Munsell's Color Tree.

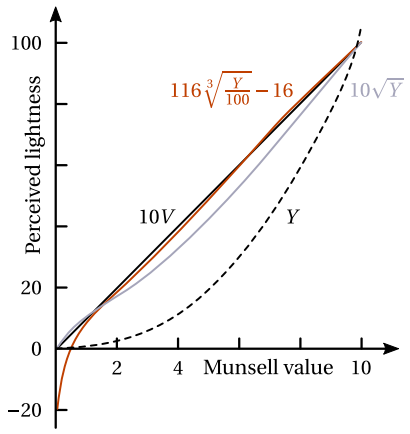
monly written as HV/C, and their counterparts in the CIE XYZ specifications. Brought together and placed according to their HV/C coordinates, the tiles form what is commonly called the Munsell color tree, as sections of constant hue seem to grow from the central axis, or the Munsell color atlas. A schematic and real-life color atlases are presented in Figures A.36 and A.37 respectively.

However and in spite of the impressive amount of tiles used for assessing perceptual uniformity, their finite number forces the resort to interpolation which is especially detrimental for the hue dimension. Extrapolation also became necessary as chroma subsequently got progressively exceeded by modern colored materials. This also partly explained the need of re-conducting experiments in the 40s.

A.4.3.2 CIE Uniform Color Spaces

Perceptual uniformity is obviously a tremendously capital feature: if guaranteed, trajectories through the color space, interpolation and distances between colors are then associated with perceptual relevance and become useful tools for color analysis and design. Munsell's focus on perceptual uniformity is more than pertinent and the CIE soon also accounted for this concern in the establishment of its subsequent colorimetric standards.

Achromatic Perceptual Uniformness Munsell's handling of perceived luminosity is an especially insightful finding as he expressed his value coordinate V as a non-linear function of the colored surface's reflected intensity which can be roughly assimilated to its luminous reflectance and hence its tristimulus value Y. This is a key observation behind nowadays' definition of *lightness* which is a perceptual attribute of color corresponding to its relative brightness. Besides Munsell's measurements and experimental validation, other findings dissuade to directly use the Y tristimulus value as a good correlate of perceived brightness such as Weber's law which states that for changes in a visual stimulus's intensity I to be noticeable, they must grow alongside the inten-

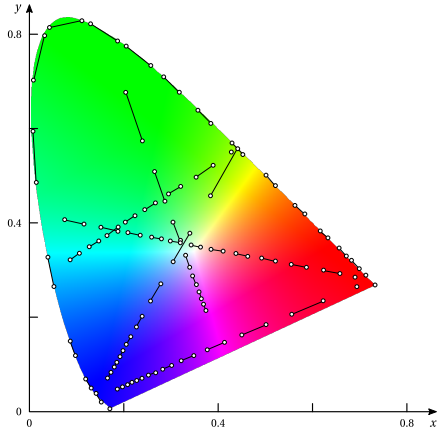


A.38 Towards a Uniform Lightness Scale. The Munsell value V is the best experimental estimate encoding the brightness of reflective surfaces in terms of perceptual evolution. It was hence used as training data to fit mathematical expressions that would then serve as computational lightness models. Here are graphed the luminous reflectance Y *(black curve)*, the Coloroid luminosity *(grey curve)*, and the uncorrected version of the CIE lightness L^* *(red curve)* assuming negative values close to the origin.

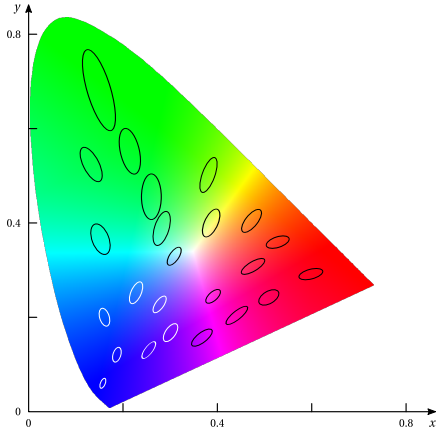
sity value, alternatively the ratio $\frac{\delta I}{I}$ must remain constant. Weber law captures the phenomenon that for our perception of intensity to change linearly, the corresponding intensity's evolution is logarithmic. Intuitively, it accounts for the observation that intensity changes are more easily detectable in bright environments than in darker places. Therefore, the relationship between our perception and the tristimulus value Y is highly non-linear. The Munsell value V from which Y can be mathematically approximated by a fifth-order polynomial, satisfies this requirement as experimentally attested, and so does the logarithm of Y and to some extent, roots of Y.

Chromatic Perceptual Uniformness Devising perceptual uniformity across chromatic planes is a much more involving process as the interplay between colors and factors dependent on the viewing conditions are harder to identify and embed in a mathematical unified framework. As explained in Section A.4.1.2, visual color matching experiments required observers to decompose the color sensation elicited by a monochromatic light by modulating the ratios of a primary light mixture in such a way both sensations would become indiscernible. No attention was given to the relative spacing *between* test colors, and their spatial organization in the resulting color spaces is directly (and up to a linear transform in the case of the XYZ specification) linked to the experimentally determined mixing ratios. The independent processing of test colors thus the perceptual uniformity of such specifications which do not consider pair-wise relationships (typically distances) between colors. The unsuitability of the CIE XYZ specifications to the human perception has notably been demonstrated by experiments carried out by Wright and MacAdams in 1941 and 1942 respectively. Their research proves the non-uniformity of the CIE xy chromaticity diagram when considered as a "map for colors" which directly endangers the CIE standards' practical usefulness. Wright considered equiluminant color pairs exhibiting small perceived differences and examined their respective chromaticity coordinates (x, y) once displayed on the chromaticity diagram. The resulting lines, coined the *Wright lines*, irrevocably highlight the xy diagram's non-uniformity since their length considerably, and unpredictably, varies with their position on it (*cf.* Figure A.39). MacAdam's subsequent findings also comfort Wright's: after repeating matching experiments for a set of 25 equiluminant test colors, he visually analyzes the variability of the results once plotted on the xy diagram by modeling standard deviations as ellipses. Again, the shapes of these ellipses change throughout the diagram as illustrated in Figure A.40, and since MacAdam proved the proportionality between the experimental standard deviations and the just-noticeable-difference for monochromatic lights, this again reinforces the fact that the CIE XYZ color system cannot be considered as perceptually uniform.

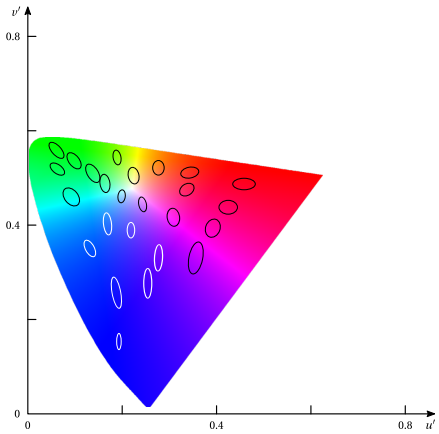
CIE 1960 UCS Diagram In order to leverage the severity of the problem, MacAdam proposed two formulae distorting the xy diagram. Thanks to their utmost simplicity, they gave rise to the internationally validated CIE 1960 UCS (Uniform Color Space) diagram before knowing a minor improvement by Eastwood in 1975 which consisted in the multiplication of the v-coordinate by 1.5 (*cf.* Equation A.2). Even if improvements can be appreciated by considering the more circular shapes of the MacAdam ellipses once expressed in these new



A.39 Wright Lines in the xy Chromaticity Diagram. Linking pairs of perceptually confusing colors, the Wright lines reveal the important amount of non-uniformity of the xy chromaticity diagram in terms of perceptual distances.



A.40 MacAdam Ellipses in the xy Chromaticity Diagram. MacAdam's confusion ellipses correspond to the variance observed when performing multiple visual matching experiments.



A.41 MacAdam Ellipses in the $u'v'$ Chromaticity Diagram. The $u'v'$ chromaticity diagram partially succeeds in achieving better perceptual uniformness. The initially too important portion of the xy diagram dedicated to greenish colors has been reduced, whereas the previously condensed blue region has been widened.

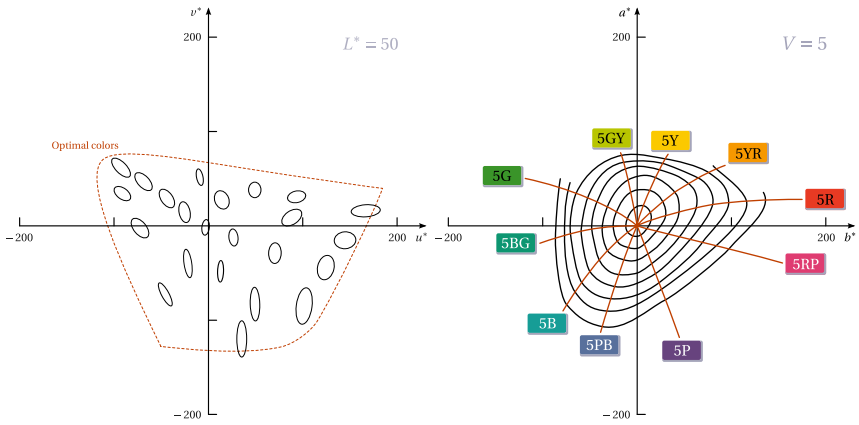
spaces, more intricate processing seems necessary to reach truly satisfactory results in terms of uniformity.

$$u = \frac{4x}{-2x+12y+3}, \quad v = \frac{6y}{-2x+12y+3}; \qquad u' = \frac{4x}{-2x+12y+3}, \quad v' = \frac{9y}{-2x+12y+3}. \tag{A.2}$$

CIE 1964 $U^*V^*W^*$ Specification One of the first CIE-attested uniform color space is Wyszecky's model from 1963. Its conversion formulae from the tristimulus values come from the two following observations. First, he approximates the perceptually linear Munsell value as the cubic root of the luminous reflectance Y, thus accounting for Weber's law. Second, he notes that higher chroma values are commonly reached for higher lightness levels, and makes his formulation reflect this direct dependency. Again, the diffuse and tangled interactions between perceived lightness and color is directly linked to our grey-scale conversion issue at the only difference that we are mostly interested into the characterization of the opposite effect: the influence of chroma over lightness. Reposing on the expression of the tristimulus values in the CIE 1960 UCS diagram, Wyszecky's work led to the CIE 1964 $U^*V^*W^*$ color system:

$$W^* \approx 10V = 25\sqrt[3]{Y} - 17, \qquad U^* = 13W^*(u - u_n), \qquad V^* = 13W^*(v - v_n), \tag{A.3}$$

where (u_n, v_n) are the chromaticity coordinates of a perfect white reflector. Note that color specifications built upon the CIE XYZ observers and aching for perceptual accuracy cannot stand as absolute spaces anymore. Similarly to color appearance models, they become relative, defined up to a reference white under a specific illuminant. After the incorporation of Eastwood's alteration to the CIE 1960 UCS diagram, Wyszecki's $U^*V^*W^*$ specification is the direct ancestor of the much more famous CIE $L^*u^*v^*$ specification homologated in 1976 by the CIE.



A.42. Evaluation of the CIE 1976 Uniform Color Spaces. Since they were conceived independently and considering separate data sets, the improvement towards perceptual uniformity proposed by the CIE $L^*u^*v^*$ and CIE $L^*a^*b^*$ spaces sure is noticeable, but it is not obvious to decide which one shows best performance. The CIE $L^*u^*v^*$ space comes from Wyszecki's $U^*V^*W^*$ space warped in a way to round off the MacAdam ellipses' shapes *(left)*. On the contrary, the CIE $L^*a^*b^*$ space builds upon the Adams-Nickerson color metric fitted to Munsell's color scales. Therefore, displaying colors of constant Munsell hue, or Munsell chroma traces almost straight lines, or circles on its color planes *(right)*.

CIE $L^*u^*v^*$ and CIE $L^*a^*b^*$ Color Spaces An inattentive glance at the color image processing literature may give the deceiving impression that both CIE $L^*u^*v^*$ and CIE $L^*a^*b^*$ can be used interchangeably. This would naturally lead to misunderstandings at best, or erroneous results at worst. Although both models do share the same lightness dimension L^* (an improved and corrected version of Wyszecki's W^*−coordinate), their encodings for the joint effects of chroma and hue originate from different experiments, resort to distinct vision models and apart from their common objective to perceptual chromatic uniformity, are barely comparable. The differences in the mathematical definitions of both systems' color coordinates, denoted (u^*, v^*) and (a^*, b^*), highlight this fact:

$$\begin{aligned} L^* &= 116 f\left(\tfrac{Y}{Y_n}\right) - 16, \quad & u^* &= 13L^*\left(u' - u'_n\right), & v^* &= 13L^*\left(v' - v'_n\right), \\ & & a^* &= 500\left(f\left(\tfrac{X}{X_n}\right) - f\left(\tfrac{Y}{Y_n}\right)\right), & b^* &= 200\left(f\left(\tfrac{Y}{Y_n}\right) - f\left(\tfrac{Z}{Z_n}\right)\right), \end{aligned}$$
$$\text{with } f(x) = \begin{cases} \sqrt[3]{x} & \text{if} \quad x > \left(\frac{24}{116}\right)^3, \\ \frac{841}{108}x + \frac{16}{116} & \text{otherwise.} \end{cases} \tag{A.4}$$

The f scalar function prevents the expression of lightness fitted to the Munsell value to reach negative values for dark colors (*cf.* red curve in Figure A.38). Even if it seems to introduce a discontinuity to the resulting expression of L^* [1], its use has been recommended by the CIE since 1976. Uniform color spaces have to sacrifice some convenient properties they could have inherited from the CIE XYZ observers. Their resort to Munsell's samples during fitting requires the prior knowledge of a reference white (X_n, Y_n, Z_n) and since they are non-linearly defined in terms of tristimulus values, they cannot be traversed as additive spaces anymore. For practicability purposes, their expressions are still kept relatively simple for the sake of invertibility. Independently from CIE $L^*u^*v^*$, the CIE $L^*a^*b^*$ model stems from Adams's research for a uniform color difference formula that he attempted to express as the Euclidean distance in his chromatic-valence space. Its main modeling constraint was to ensure that for all lightness level taken independently, its remaining dimensions kept on describing a UCS diagram and for that aim, it heavily relied on the non-linear relationship between the luminous reflectance Y and Munsell value V, Y being approximated by a fifth-order polynomial of V. After revision of its initial formulation in 1942, Adams's space was gifted with a carefully weighted metric: the Adams-Nickerson formula which played an important role in its industrial applicability. The CIE $L^*a^*b^*$ specification directly derives from Adams and Nickerson's work, and therefore does not share any common trait with the CIE $L^*u^*v^*$'s chromatic management. Significant differences hence exist between the two, each of the two models exhibiting strengths, weaknesses and are all the harder to rank as their validation sets also differ. One of the advantages of the CIE $L^*a^*b^*$ system is its improved stability with respect to the chosen illuminant which makes this space almost accounting for the phenomenon of color consistency. Color consistency designates the human

[1] Bruce Lindbloom dedicated an article on his web site concerning the discontinuity of the CIE 1976 lightness L^*: http://www.brucelindbloom.com/LContinuity.html.

visual system's ability allowing perceived colors to remain unchanged under varying illumination conditions. This benefit is absent from the CIE $L^* u^* v^*$ system which on the contrary shows an important sensitivity to the choice of reference white. Conversely, the CIE $L^* u^* v^*$ model naturally proposes a perceptual correlate of saturation as the ratio of chroma over lightness, whereas such computations would be irrelevant if transposed to the CIE $L^* a^* b^*$ coordinate system. We exploit these two color spaces' strengths for our grey-scale conversion by alternatively using these during its different stages.

A.4.4 Towards Aesthetical Uniformness

An underestimated and noteworthy color space is the *Coloroid* color specification created by Professor Antal Nemcsics's Coloroid appearance color system which results from years of research at the Budapest University of Technology and Economics from 1962 to 1980. Employed by one of our direct competing related work on grey-scale conversion, the Coloroid may not have be bestowed the status of an international CIE standard, but its many strengths have earned it the registration as Hungarian Standard MSZ 7300 since August 2000. One of its distinguishing and appealing feature is the simply incredible amount of experimental data it relies on, which involved around 70,000 test subjects and are yet to be fully processed and dutifully accounted for. But more importantly, the Coloroid outshines other color appearance models is in its attempt to make its dimensions reflect a new perceptual concern: the aesthetic uniformness.

For a scale to be *aesthetically uniform*, it has to appear to change evenly when considered in its entirety, from its starting point to its ending point, with its extremities observed simultaneously. If satisfied, aesthetic uniformity encompasses perceptual uniformness as the latter can be considered as a localized version of the former. Less subjective than expected, Nemcsics's definition of aesthetics is directly linked to his specification's targeted application cases which is to assist architects and designers for selecting harmonious color sets in the context of color planning. Therefore, keeping with the intuitive hue-chroma-luminosity decomposition is of much importance for the sake of useability. The Coloroid system proposes to represent colors in a semi-polar coordinate system, and comparatively to Munsell's system, have its components correspond to hue A_C, saturation T_C and luminosity V_C.

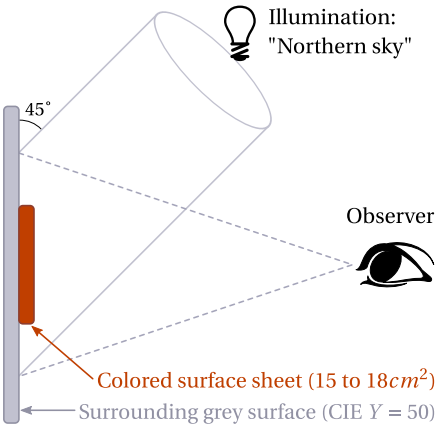
The choice of the Coloroid basic hues required most of the experimental care for them to observe Nemcsics's concept of aesthetic uniformness. Observers were instructed to order 160 colored surface samples in a circular fashion so that the resulting scale appears to change uniformly when viewed as a whole. The colors differed by approximately 6 and 12 in terms of Munsell value and chroma differences respectively. A schematic view of the set-up used for these experiments is provided in Figure A.43. The user-provided color rings then condition the system's basic hues, and unlike Munsell who proposed his model before the CIE XYZ specification, Nemcsics express them with respect to the CIE standard observers. All basic hues are specified by tracing over the xy diagram straight lines radiating from the reference white's chromaticity point, and determined as the intersection points obtained when crossing the diagram boundaries A.46 [2]. By recording the angles formed by these lines with a reference axis (Nemcsics uses the straight line pointing towards the spectral locus's red extremity), Nemcsics can propose correspondences between the CIE XYZ specification and his own system.

Nemcsics further accounts for the CIE XYZ findings by modeling each color in his space as the additive mixture of a limit color $(X_\lambda, Y_\lambda, Z_\lambda)$ with an absolute white (X_W, Y_W, Z_W) and absolute black (X_S, Y_S, Z_S). Nemcsics' limit colors refer to the maximally-saturated colors interpolated between the basic hues using Nemcsics' angular parametrization over the xy diagram. Once determined, any color sharing the limit color's hue is expressed as:

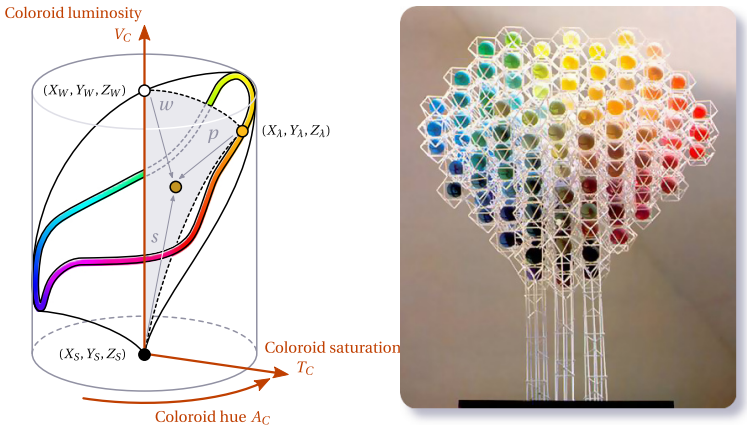
$$X = pX_\lambda + wX_W + sX_S, \quad Y = pY_\lambda + wY_W + sY_S, \quad Z = pZ_\lambda + wZ_W + sZ_S, \quad p + w + s = 1, \tag{A.5}$$

where the blending ratios (p, w, s) are called the color, white and black contents respectively. Rather naturally, Nemcsics makes any color's Coloroid saturation T_C directly proportional to its color content.

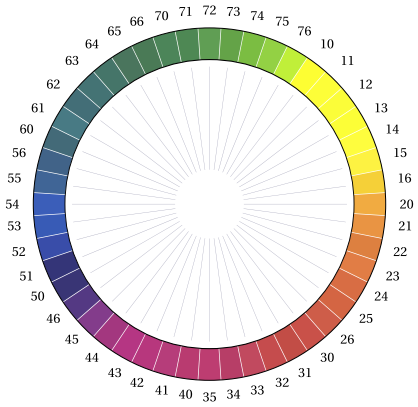
[2] Note that although the spectral locus' shape is left unchanged, the purple boundary is replaced by the straight junction between colors of wavelengths $450nm$ and $625nm$, hence excluding the strongest tints of red, blue and purple from the specification.



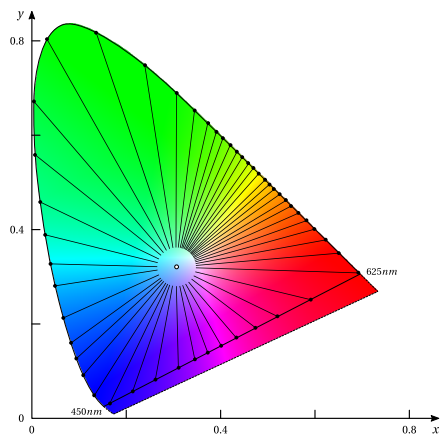
A.43 Schema Detailing the Viewing Conditions in Nemcsics's Experiments.



A.44 Schematic 3d View of Nemcsics's Colorimetric Specification. Built upon a semi-polar coordinate system, the Coloroid system proposes similar dimensions as the Munsell system: hue, saturation and luminosity. The color ribbon lying on the cylinder nappe corresponds of the continuous set of limit colors interpolated between the experimentally determined basic hues. The basic hues theoretically correspond to the stimuli caused by monochromatic lights, hence most of the limit colors stand out of the display gamut.



A.45 Coloroid Basic Hues. Determined experimentally by asking observers to create color circles of constantly evolving hue, the Coloroid 48 basic hues condition most of the aesthetical uniformness of Nemcsics's system. One can observe that some regions of the xy diagram are finely sampled (yellow and blue regions) to the detriment of others (red and green).



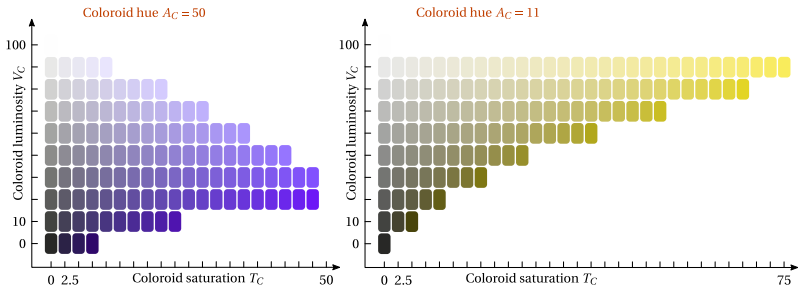
A.46 Coloroid Basic Hues over the xy Chromaticity Diagram. Nemcsics's constitutive basic hues are carefully registered as a collection of angles describing lines radiating from the D_{65} point.

The absolute white and absolute black colors respectively correspond to the colors of perfectly reflecting and perfectly absorbing surfaces respectively under a D_{65} illumination. They constitute the extremities of the vertical luminosity axis chosen so as to range from 0 to 100 and whose values roughly follow the *square root* of the limit color's luminous reflectance Y_λ. Therefore, the Coloroid luminosity V_C is not a as perceptually satisfying brightness correlate as the CIE 1976 lightness L^* in the sense that it less accurately follows the Munsell value V (*cf.* grey curve in Figure A.38).

Another perception related let-down come from Nemcsics's handling of the saturation dimension. In contrast with Munsell's chroma dimension, it does not reflect the non-linear relationship between saturation and hue anymore since all limit colors lie on the nappe of the system's enclosing cylinder. All limit colors hence share a same maximal saturation regardless of their hue whereas Munsell's empirical scales clearly show that this assumption does not hold, greenish blue always appearing duller than purple or red for instance.

By the will of its creator, the Coloroid scales' only requirement is to appear to an observer as complete and as exhibiting gradual change. A requirement notably fulfilled by its hue dimension and which represents

A.47 Constant-Hue Vertical Sections of the Coloroid Color System. Contrary to the Munsell chroma (and as misleading as the above plots may be), the Coloroid saturation always describes the same range, regardless of the hue. Here, only in-gamut colors can be shown, therefore compromising the display of the colors reaching out high saturation values.



a valuable feature at the light of Nemcsics's purpose, aesthetical uniformness being especially important for the planing of colors of greatly differing hues. This is why Nemcsics grants more attention to the perception of scales as a whole, even if it means losing perceptual relevance when considering small color differences.

A.48. Nemcsics' Coloroid Space as a Color Planing Tool [NNN05]. The main aim of the Coloroid system is to constitute an intuitive and helpful resource for persons involved in environmental color design, architects especially. Hence, many harmony rules involving the "aesthetically uniform" Coloroid hue dimension have been devised, notably by Neumann.

A.4.5 Surround Effects and Eye Adaptation

One of the most elaborated color specification system is Fairchild's CIECAM color appearance model, first proposed in 1997 and later enhanced in 2002. It represents the CIE's most formidable and recent efforts to embed many perceptual effects into a color specification system. The purpose of such a color appearance model is to bridge the gap between the representation of a color using psychometric measurements – via its tristimulus values – and the estimation of its perceived appearance attributes. As the obvious non-uniformity of the xy chromaticity diagram in terms of perceptual distance between colors demonstrated us, the CIE 1931 and 1976 XYZ specifications cannot – and are not meant to – be used as a way to predict colors' appearance characteristics. The visual intuition of artists across History, as well as the intensive research in colorimetry summed up the appearance of a color as the following set of inter-dependent and complementary attributes:

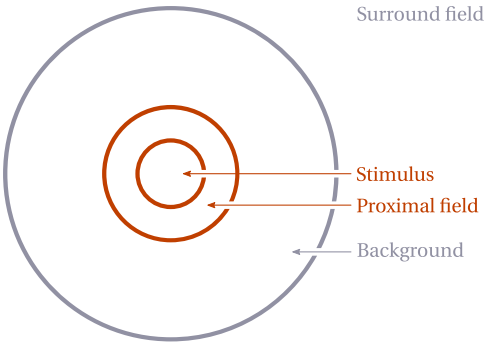
- *hue*, *i.e.* the degree to which a stimulus can be described as similar to/different from stimuli described as red, green, blue or yellow;
- *colorfulness*, *i.e.* the perceived quantity of hue content in a stimulus;
- *brightness*, *i.e.* the perceived quantity of light emanating from a stimulus;
- *saturation*, *i.e.* the colorfulness of a stimulus relative to its own brightness;
- *chroma*, *i.e.* the colorfulness of a stimulus relative to the brightness of a stimulus appearing white under the same viewing conditions;
- *lightness*, *i.e.* the brightness of a stimulus relative to the brightness of a stimulus appearing white under the same viewing conditions.

These are the definitions provided by Fairchild himself in his lectures dedicated to color appearance modeling and it is up to the color appearance model to propose mathematical formulations for these from a color's tristimulus values.

The CIECAM02 space originates from the disparate research on color appearance modeling, a field where contributions stacked up from the independent work of scientists on specific perceptual effects and dependencies between the aforementioned attributes. As an attempt to unify and bring together models such as the ones proposed by Hunt, Natayani or Fairchild, the CIE proposed a first version of its CIECAM model in

1997, along with a revised version five years later. Contrary to its ratified uniform spaces that are mainly locally relevant for the consideration of small color differences, the CIECAM is intended to constitute a full color appearance estimator accounting for the various factors influencing the viewer's perception such as the illuminating conditions, the observer's adaptation level to it or the effect of the surround.

Consequently, the CIECAM model implies a much more complicated visual stimulus model than the ones presented thus far (*cf.* Figure A.49). Since the core of the stimuli corresponds to a 2° field-of-view color patch, the CIECAM correlates are expressed in terms of the CIE 1931 XYZ specification. The *proximal field* corresponding to a further increase of 2° of the visual field can be assigned a color mismatching the central stimulus's in order to study surround effects. Beyond lies the *background* that represents an increase to 10°, and finally the *surround field* that corresponds to the peripheral vision. Taken as a whole, the proximal field, the background and the surround field support the eye's adaptation and are thus referred to as the *adaptation field*.



A.49 CIECAM Stimulus Model.

First, the stimulus's tristimulus values undergo a first transformation in order to get rough estimates of the cone cells' responses which are more amenable to the embedding of the adaptation effect. Then, a chromatic adaptation transformation accounts for the influence of the surroundings on the viewer's perception and requires the specification of the adapting white point's chromaticity values and luminance which condition the adaptive color shift and the transformation's strength respectively. Once adaptation has been accounted for, signals are further transformed using the Hunter-Pointer-Estévez cone response formula and applied non-linear compression as a way to mimic the human visual system's processes. In order to retrieve perceptually meaningful measurements, signals are combined in such a way they would yield an achromatic and color-opponent responses. This signal decomposition is directly influenced by psycho-cognition findings in human vision, and the red-green and blue-yellow color opponency computationally reflects the processing performed by the retina's bipolar cells that transmit excitatory and inhibitory signals to the spatially integrating ganglion cells.

From the achromatic and the two color-related responses, the CIECAM appearance model proposes elaborate formulae to express all six appearance attributes of the stimulus color. Of particular interest in our case, it notably allows the estimation of the color's perceived brightness. Paradoxically, the strength of the CIECAM model turns out to be its weakest point. Its unmatched sophistication and intricacy make it a very powerful color appearance predictor, but also greatly hinder its use in a broader scale. The mathematical complicatedness of its formulae left aside, it relies on the prior knowledge of the observer's viewing conditions and the background field's luminous reflectance. As a practical matter, these are usually unknown unless for a very narrow range of highly constrained application scenarios. Consequently, and as appealing relying on such an elaborate model would have been, using the CIECAM brightness as a guide for our perceptually accurate grey-scale conversion is alas not a viable solution for our case. This is all the more distressing as the CIECAM02 model does account for important perceptual effects such as the Helson-Judd effect[3], and more importantly the Helmholtz-Kohlrausch effect.

[3]The Helson-Judd effect refers to the perceptual propensity to have lighter achromatic surfaces take on the illuminant's hue, whereas darker surfaces assume its complementary hue.

APPENDIX B

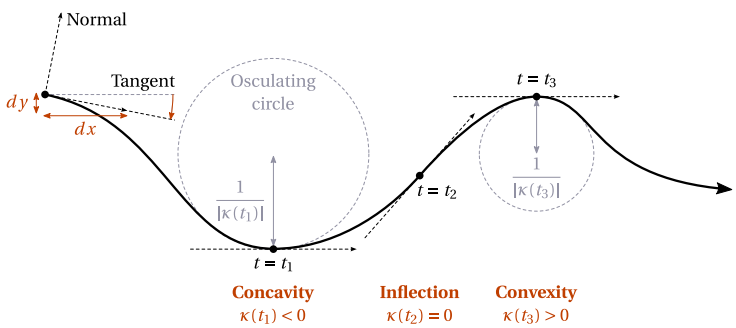
Differential Geometry: Concepts and Notations

Most existing object-space techniques for the automatic line renditions of geometry require ways to characterize, understand and describe the spatial behavior of a 3d shape. This information is fully encoded by the spatial distribution of its *normals* over its surface. Care must hence be provided to surface locations exhibiting salient features revealed by specific changes in this normal vector field. Mathematically speaking, this vague notion of changes refers to the concept of derivatives. When computed over a function, derivatives specify its infinitesimal change with respect to one of its variable. In our case, we mostly consider the multivariate function described by the object normals scattered throughout space, and henceforth introduce the notion of surface curvature.

B.1 A Detour to Flatland

Considering the case of a two-dimensional space, the *curvature* of a plane curve of parametric equations $\big(x(t), y(t)\big)$ encodes how it locally deviates from flatness. A curvature of 0 indicates that it is comparable to a straight line while a higher curvature reveals a sharper bend. It is intuitively related to the variations of the sequence of angles formed by its tangents along the course of the curve, and is written as the derivative with respect to the curve's arc-length parameter s of its tangential angular function: The curvature sign conveys important information regarding its local shape: a positive, or negative curvature value reveals that it is associated with a locally convex, respectively concave portion of the curve. Conversely, inflection points as well as straight segments are characterized by a null curvature. When defined, the reciprocal of the absolute value of curvature also specifies the radius of the *osculating circle* to the curve at the point of evaluation. Curvature at a given curve point can hence be reversely defined as the inverse of the radius of the best fitting circle to the curve as that position. This alternate formulation only unveils the curvature magnitude though, its sign requiring a globally consistent assignment with respect to the entirety of the curve. All these considerations are graphed in Figure B.1.

$$\kappa(t) = \frac{d\tan^{-1}\left(\frac{dy}{dx}\right)}{ds}.$$



B.1 2d Planar Curvatures. The curvature sign depends on the convention taken for the orientation of the normals relatively to the curve itself. Here, concavities are characterized by negative curvature values *(at t_1)*, whereas convexities are revealed by positive curvature values *(at t_3)*. Contradicting most of the differential geometry literature, this choice facilitates the understanding of the curvature of 3d surfaces whose normals are expected to point outward the enclosed shape.

B.2 Back to Sphereland

Curvature is naturally as interleaved to derivation in the three-dimensional world as it is in 2d, and appears through the derivation of the surface's normal field. But when considering such multivariate functions, derivatives assume the form of directional derivatives. A directional derivative describes the rate at which a given function, when considered from a specific point, changes when its arguments perform an infinitesimal move along a specified direction. It is here denoted $D_{\mathbf{x}} f\left(\mathbf{p}\right)$ where f, $\mathbf{p}$ and $\mathbf{x}$ respectively represent the differentiated function, the point of evaluation and the differentiation direction.

For extending curvature from curves to surfaces, one has to trade the curve's arc length parameter for two independent parameters. A *parameterized surface element* is thus defined as a map $\mathbf{f}$:

$$\begin{array}{rccc} \mathbf{f}: & \mathscr{U} & \longrightarrow & \mathbb{R}^3 \\ & (u,v) & \longmapsto & \mathbf{f}(u,v) \end{array},$$

where $\mathbf{f}$ is the parametrization of the surface defined over $\mathscr{U}$, an open set from $\mathbb{R}^2$. Elements taken from $\mathscr{U}$ are called *parameters*, whereas elements of $\mathbb{R}^3$ are referred to as *points*. For it to be thoroughly studied, $\mathbf{f}$ has to be differentiable and of maximal rank for every parameter.

As planar curves' curvatures were earlier linked to changes in its tangents, the curvature of a surface is similarly related to the variations of its tangent planes. Since a surface is fully specified by its 3d normal field, the study of the latter proposes an intuitive way of following the same rationale. Given a point sitting over $\mathbf{f}$ denoted $\mathbf{p} = \mathbf{f}(u,v)$, the *unit normal vector* at $\mathbf{p}$ is defined as follows:

$$\mathbf{n}_{\mathbf{p}} = \frac{\mathbf{f}_u \times \mathbf{f}_v}{||\mathbf{f}_u \times \mathbf{f}_v||}, \quad \text{with} \quad \begin{array}{rcl} \mathbf{f}_u & = & \frac{\partial \mathbf{f}}{\partial u}(u,v), \\ \mathbf{f}_v & = & \frac{\partial \mathbf{f}}{\partial v}(u,v). \end{array}$$

By definition, the normal vector hence points to a direction orthogonal to the surface's tangent plane at the point of evaluation. As for the tangent plane, the couple of vectors formed by $(\mathbf{f}_u, \mathbf{f}_v)$ readily constitutes a basis of its underlying vector space.

Considered as a whole, the vector field made of all the outward normal vectors represents the surface's **first-order structure** as their computations involve the first derivatives of the surface, and it is encoded by its *Gauss map*:

$$\begin{array}{rccc} \mathbf{n}: & \mathscr{U} & \longrightarrow & \mathbb{S}^2 = \left\{\mathbf{x} \in \mathbb{R}^2;\ ||\mathbf{x}|| = 1\right\} \\ & (u,v) & \longmapsto & \mathbf{n}(u,v) = \frac{\mathbf{f}_u \times \mathbf{f}_v}{||\mathbf{f}_u \times \mathbf{f}_v||} \end{array}.$$

However, more than in the tangent planes, we are more interested in their variations, and higher-order structures thus need to be estimated. Analyzing the Gauss map's directional derivatives is then of uttermost relevance for approaching the notion of surface curvature, and proposing a meaningful assessment of the surface's local saliency. The analysis of the surface's *shape operator* answers such concerns. Given a point of the surface $\mathbf{f}(u,v)$, the shape operator at that point is defined as:

$$\begin{array}{rccc} \mathscr{S}_{(u,v)}: & T_{(u,v)}\mathbf{f} & \longrightarrow & T_{(u,v)}\mathbf{f} \\ & \mathbf{x} & \longmapsto & D_{\mathbf{x}}\left(\mathbf{n} \circ \mathbf{f}^{-1}\right)(u,v) \end{array} \quad [1] \tag{B.1}$$

and corresponds to the linear map that computes at $\mathbf{f}(u,v)$ the derivative of the surface's Gauss map $\mathbf{n}$ once viewed as a vector field along the surface $\mathbf{f}$. $T_{(u,v)}\mathbf{f}$ denotes the tangent plane of the surface $\mathbf{f}$ at point $\mathbf{f}(u,v)$. As unit vectors, normals are orthogonal to their derivatives which therefore belong to their associated tangent plane. The shape operator $\mathscr{S}_{(u,v)}$ is therefore a linear transform of the tangent plane $T_{(u,v)}\mathbf{f}$ to itself, and its application to a tangential direction thus represented as the product with a 2×2 matrix. If we express the tangential direction $\mathbf{x}$ in the $(\mathbf{f}_u, \mathbf{f}_v)$ basis of the tangent plane, the coefficients of the matrix associated with

[1] It should be noted that this definition differs in sign from the one usually found in textbooks. This choice is motivated by the fact we deal with surfaces of outward pointing normals.

$\mathscr{S}_{(u,v)}$ are obtained by projecting the second partial derivatives of $\mathbf{f}$ onto the normal line $\mathbf{n}(u,v)$:

$$\mathscr{S}_{(u,v)}(\mathbf{x}) = \begin{pmatrix} \mathbf{f}_{uu}.\mathbf{n}(u,v) & \mathbf{f}_{uv}.\mathbf{n}(u,v) \\ \mathbf{f}_{uv}.\mathbf{n}(u,v) & \mathbf{f}_{vv}.\mathbf{n}(u,v) \end{pmatrix} \begin{pmatrix} x_u \\ x_v \end{pmatrix}, \quad \text{with} \quad \begin{aligned} \mathbf{f}_{uu} &= \tfrac{\partial^2 \mathbf{f}}{\partial^2 u}(u,v), \\ \mathbf{f}_{uv} &= \tfrac{\partial^2 \mathbf{f}}{\partial u \partial v}(u,v), \\ \mathbf{f}_{vv} &= \tfrac{\partial^2 \mathbf{f}}{\partial^2 v}(u,v), \\ \mathbf{x} &= x_u \mathbf{f}_u + x_v \mathbf{f}_v. \end{aligned}$$

This formulation clearly reveals if needed that the shape operator of $\mathbf{f}$ at constitutes one of the many glimpses at the surface's **second-order structure**.

Conveying crucial differential information, the shape operator naturally appears in one of the most important local descriptor of the surface: its *second fundamental form*. Somewhat reminiscent of the osculating circle of a planar curve, the second fundamental form of a surface $\mathbf{f}$ at a point $\mathbf{f}(u,v)$, noted $\mathrm{II}_{(u,v)}$, is the quadratic polynomial expressed in the surface tangent plane that best approximates the surface's behavior at that point. Taking as arguments two tangential directions $\mathbf{x}$ and $\mathbf{y}$ at $\mathbf{f}(u,v)$, it is defined in terms of the shape operator as:

$$\mathrm{II}_{(u,v)}(\mathbf{x},\mathbf{y}) = \mathbf{x}.\mathscr{S}_{(u,v)}(\mathbf{y}) = \begin{pmatrix} x_u & x_v \end{pmatrix} \begin{pmatrix} \mathbf{f}_{uu}.\mathbf{n}(u,v) & \mathbf{f}_{uv}.\mathbf{n}(u,v) \\ \mathbf{f}_{uv}.\mathbf{n}(u,v) & \mathbf{f}_{vv}.\mathbf{n}(u,v) \end{pmatrix} \begin{pmatrix} y_u \\ y_v \end{pmatrix}, \quad \text{with} \quad \begin{aligned} \mathbf{x} &= x_u \mathbf{f}_u + x_u \mathbf{f}_v, \\ \mathbf{y} &= y_u \mathbf{f}_u + y_u \mathbf{f}_v. \end{aligned}$$

A convenient and highly valuable property of the shape operator is that it is actually independent of the surface parametrization $\mathbf{f}$. It is therefore often expressed in the following, more legible form:

$$\mathscr{S}_{\mathbf{p}}(\mathbf{x}) = D_{\mathbf{x}}\mathbf{n}\left(\mathbf{p}\right),$$

which hides all traces of the parametrization [2], and makes its intuition more explicit: the shape operator at a surface point $\mathbf{p}$ describes how the normal field evolves at the vicinity of $\mathbf{p}$ in a specified direction $\mathbf{x}$ and how it curves near it.

Belonging to the tangent plane at $\mathbf{p}$, the derivative in the tangential direction $\mathbf{x}$ of the normal field $\mathbf{n}$ at that point $D_{\mathbf{n}}\mathbf{x}\left(\mathbf{p}\right)$ can be expressed in the tangent plane's basis $\left(\mathbf{x},\mathbf{x}^{\perp}\right)$, with $\mathbf{x}^{\perp} = \mathbf{n}_{\mathbf{p}} \times \mathbf{x}$. This decomposition is particularly enlightening as the projection of $D_{\mathbf{n}}\mathbf{x}\left(\mathbf{p}\right)$ on $\mathbf{x}$ reveals how the surface bends along $\mathbf{x}$, whereas its projection on $\mathbf{x}^{\perp}$ exposes how it tilts along $\mathbf{x}$. The former coordinate is called the *normal curvature* of the surface at $\mathbf{p}$, the latter its *geodesic torsion* at $\mathbf{p}$ (*cf.* Figure B.2).

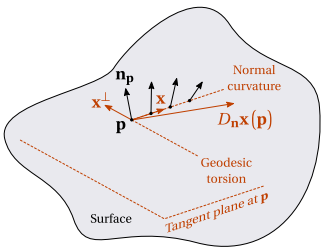
So here it is. The normal curvature of a surface at a given location $\mathbf{p}$ corresponds to the projection of the normal field's directional derivative onto the tangential direction of derivation. It is henceforth defined up to the direction $\mathbf{x}$ as:

$$\kappa_{\mathbf{n}}(\mathbf{x}) = \frac{\mathscr{S}_{\mathbf{p}}(\mathbf{x}).\mathbf{x}}{||\mathbf{x}||} = \frac{\mathrm{II}_{\mathbf{p}}(\mathbf{x},\mathbf{x})}{||\mathbf{x}||}$$

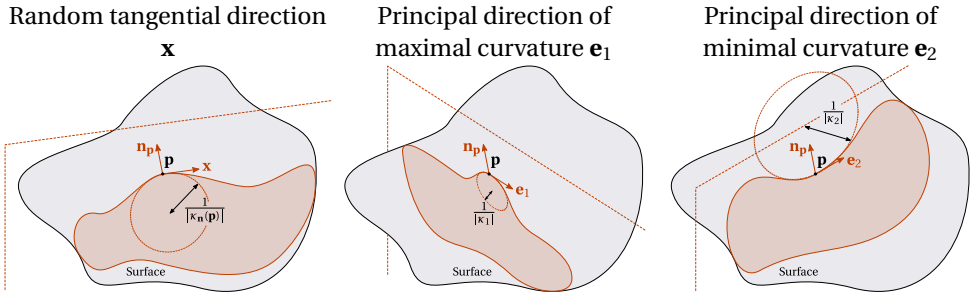
As commonly observed in differential geometry, the analysis of the properties of a surface estimated at a given location is easier to conduct through the study of the curves running across the surface and passing through the considered point. Normal curvature is no exception as the normal curvature at $\mathbf{p}$ in the tangential direction $\mathbf{x}$, also corresponds to the curvature at $\mathbf{p}$ of the planar curve resulting from the intersection of the surface with the plane spanned by the normal $\mathbf{n}_{\mathbf{p}}$ and $\mathbf{x}$. For every new tangential direction $\mathbf{x}$, the normal curvature at $\mathbf{p}$ is to differ and is directly dependent on the secund fundamental form's matrix, its spectral analysis revealing valuable insights on the behavior of the normal curvature at that location. Since $\mathrm{II}_{\mathbf{p}}$ is a symmetric bilinear form, so is its associated matrix which henceforth assumes the form of a diagonal matrix after rotation of the tangent plane's frame.

Let us denote (κ_1,κ_2) the diagonal coefficients of the matrix once rotated (with $|\kappa_1| \geq |\kappa 2|$), and $(\mathbf{e}_1,\mathbf{e}_2)$ the tangent plane's basis inside of which the matrix appears diagonal. It turns out that κ_1 and κ_2 represent the extremal values reached by the normal curvature at $\mathbf{p}$. They are called the maximal, respectively minimal, *principal curvatures* of the surface at $\mathbf{p}$ and correspond to the normal curvature computed along the $\mathbf{e}_1$ and

[2] And along with it possible concerns related to the injectivity of the Gauss map which is now directly defined over the surface's points.



B.2 Directional Derivative of the Normal Field.

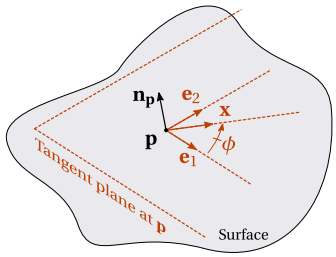


B.3 3d Normal Curvature. Varying alongside the tangential direction $\mathbf{x}$ employed to assess the surface normals' variation rate, the normal curvature $\kappa_{\mathbf{n}}$ remains equivalent in terms of magnitude to the reciprocal of the radius of the oscillating circle of the curve emerging from the section of the surface by the normal and $\mathbf{x}$.

$\mathbf{e}_2$ tangential directions respectively. Orthogonal to each other, the directions $\mathbf{e}_1$ and $\mathbf{e}_2$ are referred to as the maximal, respectively minimal, *principal curvature directions* of the surface at $\mathbf{p}$ (*cf.* Figure B.3).

The information conveyed by the surface's principal curvatures and directions is remarkable as their sole knowledge at a point allows the computation of the normal curvature $\kappa_{\mathbf{n}}$ and geodesic tension τ_{g} along any other tangential direction $\mathbf{x}$ (*cf.* Figure B.4).

$$\begin{aligned} \kappa_{\mathbf{n}}(\mathbf{x}) &= \kappa_1 \cos^2\phi + \kappa_2 \sin^2\phi, \\ \tau_{\mathrm{g}}(\mathbf{x}) &= (\kappa_2 - \kappa_1)\cos\phi\sin\phi, \\ \text{with } \phi &= \text{angle}(\mathbf{x}, \mathbf{e}_1). \end{aligned} \tag{B.2}$$



B.4 Euler Formula for Normal Curvature and Geodesic Torsion.

But more importantly, principal curvatures shine thanks to their surface descriptive power: not only their associated directions indicate the directions along which the normals' variations are extremal, but their sign also conveys crucial information about the local shape of the surface. Best revealed by their product κ_{g} and arithmetical mean κ_{h}

$$\begin{aligned} \kappa_{\mathrm{g}} &= \kappa_1\kappa_2 && \text{or } \textit{gaussian curvature}, \\ \kappa_{\mathrm{h}} &= \tfrac{1}{2}(\kappa_1 + \kappa_2) && \text{or } \textit{mean curvature}, \end{aligned}$$

this second-order information can serve by way of simple sign tests as an efficient and rather elaborated shape classification criterion (*cf.* Figure B.5). The information detained by the gaussian curvature alone is especially valuable as, contrary to normal, principal or mean curvature, it describes the surface from an *intrinsic* standpoint and is independent of the surface's embedding in the three-dimensional space.

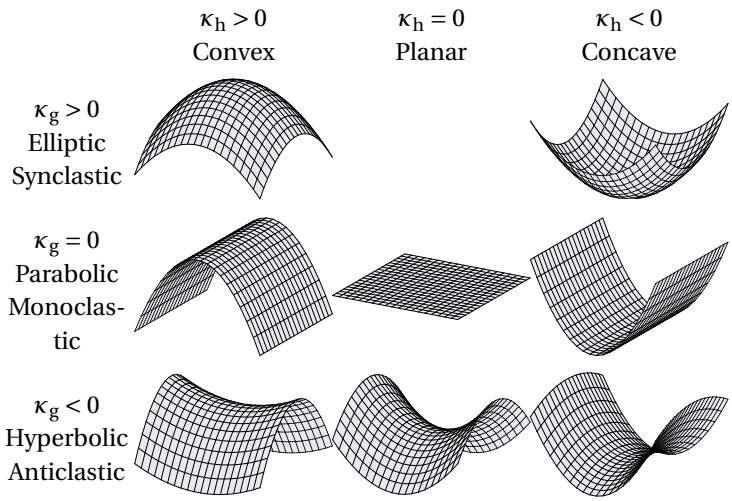
At the light of the insights conveyed by the surface's second-order structure, it comes with no surprise that existing line rendering techniques make an heavy use of the curvature information to find relevant line locations. But it only describes the surface's local shape. If one wants to draw *lines* out of it (other than zero-crossing loci), one needs to study the evolution of the surface's behavior which requires the estimation of the normal curvature's derivative typically in the direction of the candidate feature line. For that aim, the surface's **third-order structure** at $\mathbf{p}$ is unveiled by computing the partial derivatives of the second fundamental form $\mathrm{II}_{\mathbf{p}}$ in the tangent plane, resulting in a $2 \times 2 \times 2$ tensor $\mathrm{C}_{\mathbf{p}}$. For simplicity purposes, it is much more convenient to consider its computation in the tangent plane's basis $(\mathbf{e}_1, \mathbf{e}_2)$ where $\mathrm{II}_{\mathbf{p}}$ assumes a diagonal form:

$$\mathrm{C}_{\mathbf{p}} = \begin{pmatrix} D_{\mathbf{e}_1}\mathrm{II}_{\mathbf{p}} & D_{\mathbf{e}_2}\mathrm{II}_{\mathbf{p}} \end{pmatrix} = \left(\begin{pmatrix} D_{\mathbf{e}_1}\kappa_1 & D_{\mathbf{e}_2}\kappa_1 \\ D_{\mathbf{e}_2}\kappa_1 & D_{\mathbf{e}_1}\kappa_2 \end{pmatrix} \begin{pmatrix} D_{\mathbf{e}_2}\kappa_1 & D_{\mathbf{e}_1}\kappa_2 \\ D_{\mathbf{e}_1}\kappa_2 & D_{\mathbf{e}_2}\kappa_2 \end{pmatrix} \right)$$

The complete unfolding of the expressions behind the tensor of curvature derivative's coefficients reveals their

reliance on third-derivatives of the surface **f** at **p**, and their exact formulae can be found in [GU02]. The derivative of the normal curvature at **p** in the direction **x** can thus be obtained through the triple multiplication between **x** and the tensor of curvature derivative which after normalization of **x**, is expressed as follows:

$$\frac{D_{\mathbf{x}}\kappa_{\mathbf{n}}(\mathbf{x})}{||\mathbf{x}||} = \frac{\mathrm{C}_{\mathbf{p}}(\mathbf{x},\mathbf{x},\mathbf{x})}{||\mathbf{x}||^3}.$$



B.5. Description of the Surface's Local Shape. The computation of the mean and gaussian curvatures, respectively κ_h and κ_g, at a given point of the surface carries much information about its spatial behavior in the vicinity of that point. By the study of their sign, one can come up with a rough classification of surfaces that incorporates notions such as convexity versus concavity through the study of κ_h[3], or its overall shape by the analysis of the κ_g. Indeed, a positive gaussian curvature indicates that the normal curvature keeps a constant sign for all tangential directions and that the surface only touches its tangent plane at a single point. The surface (comparable to an eggshell) therefore bents the same way in all directions, hence the term *synclastic (top)*. Conversely, a null gaussian curvature reveals that the surface does not exhibit any bending in at least one of its principal direction of curvature. It thus corresponds to a *monoclastic* region, locally comparable to a cylinder *(middle)*. Lastly, if negative, the gaussian curvature attests a double zero-crossings of the normal curvature as the tangential directions rotate around the considered point. It locally curves in two opposite ways and thus corresponds to an *anticlastic* region which assumes a saddle-like shape *(bottom)*.

B.3 Differential Geometry in Practice

Nevertheless, the formulae scattered across Section B.2 all take for granted the availability of a surface parametrization assuming an analytic form. In such a purely theoretical context, one just has to differentiate it once for the normals, twice for its principal curvature magnitudes and directions, and finally thrice for its tensor of curvature derivative at the point of interest. But in practice, available geometric inputs strongly deviate from this idealistic canvas, and mostly involves complex surfaces for which no global parametrization is possible and whose representation in our case consists of triangle meshes. This seemingly insignificant drift from the ideal case is actually quite dreadful: the effective complexity of the models' topology, the presence of holes, the unknown valence of their vertices, the potential lack of consistent face orientation and above all the irregularity of the sampling of the actual surface pose a series of formidable challenges for the estimation of the curvatures and their derivatives.

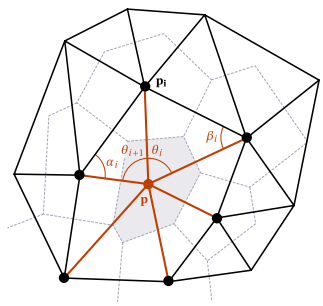
Straightforward heuristics exist for the rough approximation of the gaussian and mean curvatures at the vertices of triangle meshes (*cf.* Figure B.6). But their coarseness aside, they do not provide any information either on the directions of principal curvature or the curvature derivatives. Estimating these requires at least the approximation of the matrix of the second fundamental form at each vertex and many methods (few of which will be presented herein) have since then been proposed. They roughly fall into three main categories: curve sampling, patch-fitting and tensor averaging.

B.3.1 Curve Sampling

Curve sampling methods draw inspiration from the alternative definition of the normal curvature at vertex **p** in direction **x** as the radius of the osculating circle to the curve obtained by intersecting the surface with the

[3] Recall the signs of the principal curvatures have been reversed due to our alteration of the definition of the surface's shape operator (*cf.* Equation B.1).

B.6 Approximation of the Gaussian and Mean Curvature on Triangle Meshes. The integral gaussian curvature at a mesh vertex $\mathbf{p}$ can be assimilated to its *angle excess* computed as $\sum_i \theta_i - 2\Pi$. When equal to 0, it attests the surface behaves like a planar surface. Comparatively, the integral mean curvature can be approximated as $\frac{1}{2}\sum_i \left(\cot\alpha_i + \cot\beta_i\right)\left|\left|\mathbf{p}-\mathbf{p_i}\right|\right|$. In order to retrieve κ_g and κ_h, one still needs to normalize these two integrals by the area of integration, a reasonable choice for it being the area of the Voronoi cell of $\mathbf{p}$ *(light grey region)*. While coarse, this approximation has the advantage to only rely on the 1-ring neighborhood $\left(\mathbf{p_i}\right)$ of $\mathbf{p}$ *(red edges)* which guarantees good performances.



plane spanned by $\mathbf{x}$ at the normal at $\mathbf{p}$. Normal curvatures at $\mathbf{p}$ are computed for various tangential directions $(\mathbf{x_i})$ and are then used as constraints to a linear system of equations resulting from the following equality:

$$\kappa_{\mathbf{n}}\left(\mathbf{x_i}\right) = \mathrm{II}_{\mathbf{p}}\left(\mathbf{x_i},\mathbf{x_i}\right) = \begin{pmatrix} a & b \\ b & c \end{pmatrix}\begin{pmatrix} \mathbf{x_i}.\mathbf{u} \\ \mathbf{x_i}.\mathbf{v} \end{pmatrix},$$

with $(\mathbf{u},\mathbf{v})$ denoting a basis of the tangent plane at $\mathbf{p}$ common to all equations. The matrix coefficients (a,b,c) are estimated by least-square fitting, and the resulting second fundamental form's matrix can be made diagonal to determine the principal curvatures and directions at $\mathbf{p}$ [MDSB03]. More flexible "curve models" such as Agam and Tang's generated Bézier surface patches, can also be fitted to the vertices at the vicinity of $\mathbf{p}$ in order to compute information of higher-order while improving accuracy [AT05]. The main drawback of these approaches is their direct dependency to the sampling frequency and regularity of the tangential directions that control the collection of curves on which optimization is performed. Moreover, the nonexistence of clear adaptive sampling strategy for that purpose imposes the tuning of a parameter controlling the global density of the sampling.

B.3.2 Patch-Fitting

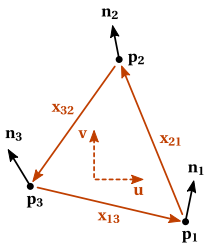
Probably the most intuitive approach of all three trends for curvature estimation on triangle meshes, patch-fitting techniques tackle the problem at its very core, namely the discrete nature of the input geometry. They find a remedy to the situation by locally fitting parametric surfaces independently at each vertex $\mathbf{p}$, and going back to the continuous case where curvature and its derivatives can be analytically computed using the formulae described in Section B.2. Since they belong to the surface's second-order structure, principal curvatures and their directions require the fitting of at least a quadratic surface $\mathbf{f}$ that will serve as a local parametrization of the surface in the tangent plane of the surface at $\mathbf{p}$:

$$\mathbf{f}(u,v) = \frac{a}{2}u^2 + buv + \frac{c}{2}v^2 = \begin{pmatrix} \frac{1}{2}u^2 & uv & \frac{1}{2}v^2 \end{pmatrix}\begin{pmatrix} a \\ b \\ c \end{pmatrix}.$$

By definition, (a,b,c) corresponds to the coefficients of the second fundamental form's matrix and are determined by linear least-square fitting once a system of equations involving the projections of neighboring vertices $\left(\mathbf{p_i}\right)$ onto the tangent plane at $\mathbf{p}$ have been gathered:

$$\begin{pmatrix} a & b & c \end{pmatrix}\begin{pmatrix} u_i^2 & 2u_iv_i & v_i^2 \end{pmatrix} = \frac{2}{\left|\left|\mathbf{p_i}-\mathbf{p}\right|\right|}\left(\mathbf{p_i}-\mathbf{p}\right).\mathbf{n}, \qquad \text{with} \quad \begin{array}{l} u_i = \frac{1}{\left|\left|\mathbf{p_i}-\mathbf{p}\right|\right|}\left(\mathbf{p_i}-\mathbf{p}\right).\mathbf{u}, \\ v_i = \frac{1}{\left|\left|\mathbf{p_i}-\mathbf{p}\right|\right|}\left(\mathbf{p_i}-\mathbf{p}\right).\mathbf{v}, \end{array}$$

where $(\mathbf{u},\mathbf{v})$ is the tangent plane's basis vectors. This method is directly comparable to the curve sampling approach. Patch-fitting still commands a marginal advantage in the sense that it lifts all sampling concerns and in place of a circle, approximate sections of the surface by parabolas. Higher-order parametric surfaces can naturally replace the quadratic surfaces for fitting. However, for such an optimization to be possible, more



$$D_{\mathbf{x}_{21}}\mathbf{n}(\mathbf{p}_1) = \mathscr{S}_{\mathbf{p}_1}(\mathbf{x}_{21}) \Leftrightarrow \begin{pmatrix}(\mathbf{n}_2-\mathbf{n}_1).\mathbf{u}\\(\mathbf{n}_2-\mathbf{n}_1).\mathbf{v}\end{pmatrix} \approx \begin{pmatrix}a & b\\ b & c\end{pmatrix}\begin{pmatrix}\mathbf{x}_{21}.\mathbf{u}\\\mathbf{x}_{21}.\mathbf{v}\end{pmatrix}$$

$$D_{\mathbf{x}_{32}}\mathbf{n}(\mathbf{p}_2) = \mathscr{S}_{\mathbf{p}_2}(\mathbf{x}_{32}) \Leftrightarrow \begin{pmatrix}(\mathbf{n}_3-\mathbf{n}_2).\mathbf{u}\\(\mathbf{n}_3-\mathbf{n}_2).\mathbf{v}\end{pmatrix} \approx \begin{pmatrix}a & b\\ b & c\end{pmatrix}\begin{pmatrix}\mathbf{x}_{32}.\mathbf{u}\\\mathbf{x}_{32}.\mathbf{v}\end{pmatrix}$$

$$D_{\mathbf{x}_{13}}\mathbf{n}(\mathbf{p}_3) = \mathscr{S}_{\mathbf{p}_3}(\mathbf{x}_{13}) \Leftrightarrow \begin{pmatrix}(\mathbf{n}_1-\mathbf{n}_3).\mathbf{u}\\(\mathbf{n}_1-\mathbf{n}_3).\mathbf{v}\end{pmatrix} \approx \begin{pmatrix}a & b\\ b & c\end{pmatrix}\begin{pmatrix}\mathbf{x}_{13}.\mathbf{u}\\\mathbf{x}_{13}.\mathbf{v}\end{pmatrix}$$

B.7 Rusinkiewicz's Estimation of the Curvature Tensor [Rus04]. Above are the linear constraints used to determine the coefficients of the second fundamental form's matrix for a given face of the mesh. Obtained tensors are subsequently averaged on a per-vertex basis once the tensors are brought to a common frame.

neighboring vertices need to be considered which requires the knowledge of more elaborated connectivity information, and as a side effect smoothes the surface's features. By incorporating the adjacent vertices' approximated normals, more complex and faithful models can be fitted to the geometry without distancing the 1-ring neighborhood of $\mathbf{p}$. As observed by Goldfeather and Iterrante, the cubic-order approximation of curvature yields substantively superior results than a direct quadratic fit only considering the neighbors' relative positions [GI04].

Both curve sampling and path-fitting are model-based approaches for the computation of curvature information on discrete geometry, and as such share common issues. Chief among these, the sensitivity to geometric noise and to the regularity of the tessellation. Indeed, irregularly sampled surfaces are unevenly approximated by the chosen model in regions of weak vertex density. Outliers also endanger the viability of the optimization framework and ambiguities due to specific vertex layouts (typically co-linearity) cause degeneracies. Lastly, the extent of the neighborhood used for optimization (directly linked to the fitted model's complexity) correlates the quality of the approximation and the elimination of noise, one unfortunately at the detrimental of the other.

B.3.3 Averaging

Curvature estimation by tensor averaging meets many desirable requirements and finds roundabouts to most issues plaguing other approaches : stability over 1-ring neighborhoods, no requirement over the topology and the absence of degenerate cases [Rus04]. In a recent extension suggested Batagelo and Ting, comparison tests showed the unmatched performance of Rusinkiewicz's estimates when confronted to irregular, or noisy surfaces. This method can be seen as a generalization of the approximation of vertex normals as the weighted averages of the normals from adjacent faces [Max99]. Similarly, the matrix of the local surface's second fundamental form can be found by least-square on a per face basis as explained in more details in Figure B.7. A second pass then computes the weighted average of these tensors for each vertex. Special care must still be taken so that the summed tensors are expressed in a common basis, and a relevant weighting of the different contributors to the average significantly improves the quality of the final estimate. This method is also perfectly extensible to higher-order surface differential quantities. While the estimation of the curvature tensors involves the derivatives by finite-difference of the normal in the directions imposed by the edges of the face, similar computations using the principal curvatures in the $(\mathbf{e}_1, \mathbf{e}_2)$ basis of the face's plane can straightforwardly be performed.

APPENDIX C

French Summary

Préface Voici une brève présentation des thématiques, problématiques et contributions de ma thèse. Cette dernière se place dans le contexte de l'informatique graphique, et plus précisément dans le cadre relativement spécifique et récent du rendu dit expressif. La communauté la plus susceptible de manifester de l'intérêt quant à un tel sujet est en très grande partie anglophone et il est dès lors plus simple d'y communiquer ses travaux et recherches en langue anglaise. C'est le choix que j'ai également effectué pour la rédaction de mon propre travail de doctorat et ce bien qu'ayant bénéficié d'une bourse d'état française. Ce parti-pris peut choquer car il exclut d'éventuels lecteurs moins habitués à la lecture de documents techniques en anglais. C'est pour cette raison que je propose, dans les pages qui suivent, une présentation brève, mais tout aussi complète de mes recherches en français. La structure de la présentation qui suit respecte le plan de mon manuscrit en parties et chapitres, et s'efforce d'en fournir un aperçu le plus fidèle possible en respectant toutes les réflexions et techniques détaillées dans le reste du document.

L'art digital : de nos écrans d'ordinateur aux toiles de cinéma

La création de contenus visuels via l'ordinateur n'est à proprement parler pas un phénomène nouveau, mais la soudaine étendue de son champ d'action l'est au contraire. En effet, très tôt dans l'histoire de l'informatique, les chercheurs et ingénieurs en charge de la conception et de la réalisation des premiers ordinateurs ont concentré leurs efforts sur l'opérabilité de telles machines, et sur l'échange d'informations visuelles pour s'assurer la pertinence et l'exactitude de leurs calculs, et cela continue de constituer une passerelle précieuse entre la machine et son utilisateur. Ainsi, le célèbre ordinateur *Whirlwind* développé au sein du *Massachusetts Institute of Technology* dès 1945 (probablement le premier ordinateur jamais créé au sens où nous l'entendons aujourd'hui) avait, entre autres spécificités, déjà proposé des sorties vidéo. Ainsi, la faculté de "dessin" des ordinateurs est naturellement apparue dès leur invention, et ce notamment à des fins de *communication* avec leurs utilisateurs humains.

Aujourd'hui, la donne a évidemment bien changé, mais il est cependant intéressant de constater la rapidité avec laquelle le but essentiel du support visuel a progressivement dévié de cette finalité initiale d'aide à la visualisation scientifique pour se tourner vers l'art. Effectivement, il n'aura pas fallu attendre bien longtemps pour voir les ordinateurs susceptibles de concevoir des images dont le seul but réside dans leur caractère purement esthétique, à défaut de véhiculer une information d'ordre scientifique, et ainsi donner naissance à l'"art digital". Les premières créations ont tout d'abord été l'oeuvre d'ingénieurs et d'universitaires, rares privilégiés à avoir accès aux premiers ordinateurs et possédant les connaissances requises pour leur mise en œuvre. Leurs expérimentations visuelles avec l'ordinateur ont notamment coincidé avec l'émergence des courants artistiques post-modernes dits "non-représentationnels". Parmi eux, le cubisme et l'*op'art* (ou *optical art*) qui se caractérisent par l'emploi de formes simples, hautement géométriques et dont la minutieuse répétition permet notamment aux œuvres observées dans leur ensemble de susciter des illusions optiques. Ainsi, de nombreux ingénieurs ont naturellement été amenés à détourner les techniques de visualisation scientifique des ordinateurs pour imiter de telles œuvres, à tel point que dès la moitié des années soixante, les premières expositions consacrées au potentiel de l'ordinateur dans le domaine artistique rencontraient un succès inespéré, comme la pionnière *Generative Computergrafik* de Stuttgart en 1965 ou bien évidemment la célèbre *Cybernetic Serendipity* de Londres en 1968. De même, l'accession de telles créations artistiques générées par ordinateur au marché de l'art s'opéra dès la décennie suivante.

Par définition, l'art digital désigne l'ensemble de la création artistique pour laquelle l'ordinateur, et par extension l'informatique et les mathématiques, ont contribué à sa conception. Défini plus par les moyens mis en œuvre pour sa réalisation que par sa finalité ou une unité de choix esthétique, l'art digital ne peut se ranger au même niveau que d'autres courants artistiques de Histoire de l'Art. Pourtant, nous en faisons l'expérience fréquemment, tant les ordinateurs se sont inscrits dans notre quotidien, voire le façonnent aujourd'hui. Il était donc naturel que la révolution numérique ait un impact crucial sur nos divertissements et modes de création et de représentation. Nous ne comptons plus à présent le nombre de créations exclusivement réalisées via le support numérique, et l'ordinateur est à présent considéré comme un médium artistique à part entière.

À son origine, l'art digital revêtait une forme plutôt académique, chaque œuvre étant considérée comme unique. En effet, l'ordinateur était placé au centre du processus de création, mais également de son message. Au cours de son évolution et placé hors de ce contexte, l'ordinateur a vite été (et il est encore pour certains) quasiment perçu comme une menace vis-à-vis du travail de l'artiste, l'automatisation qu'il propose étant assimilée à une solution de facilité opposée à la minutie nécessaire à la création artistique dans son sens noble. Ce n'est que très récemment qu'un renversement visible est apparu quant à la perception par les artistes de la machine et de sa présence au sein de la production artistique à plus grande échelle. La mâturité technique des machines et de leurs logiciels permettaient alors aux artistes de donner libre cours à leur imagination et d'obtenir des résultats inaccessibles à tout autre procédé artistique.

Parmi les facteurs d'ordre technique ayant joué un rôle déterminant dans ce changement, notons évidemment l'augmentation constante de la rapidité des processeurs qui a rendu possible l'accélération voire l'interaction de techniques de plus en plus complexes; l'augmentation de la résolution des périphériques d'entrée et de sortie (écrans, palettes graphiques) permettant une manipulation et un contrôle de plus en plus fin par les artistes; et enfin l'augmentation exponentielle de leur capacité mémoire. Ce dernier aspect s'est notamment révélé crucial pour garantir l'exploitabilité des outils numériques de création visuelle et permettre leur démarcation vis-à-vis des supports artistiques traditionnels. En effet, grâce à une capacité mémoire de plus en plus importante, il est à présent possible pour les artistes non seulement de sauvegarder leurs créations à tout moment, mais également en cas d'erreur de restaurer l'état précédent de leurs œuvres. La levée par l'ordinateur de l'irréversibilité du geste de l'artiste, caractéristique de la difficulté de l'utilisation de média artistiques tangibles, a marqué un tournant capital pour son adoption à grande échelle, notamment dans le cadre des applications type *virtual canvas* ("toiles virtuelles", littéralement). Certains pourront à juste titre défendre *mordicus* le support physique et la subtilité de ses effets contrôlables uniquement via le retour de force du pinceau sur la toile, mais on ne peut nier les nombreuses avancées en termes d'émulation à défaut de véritable simulation d'une partie de ceux-ci par la machine et ainsi leur progressive incorporation aux applications les plus avancées.

Il est difficile de définir exactement ce qu'est l'art par ordinateur car les utilisateurs sont nombreux et la démocratisation des ordinateurs touche un nombre extraordinairement varié de personnes originales. Nous en distinguons trois grandes catégories, certes caricaturées mais bien réelles : le "programmeur", l'"artiste", et enfin l'utilisateur qualifié d'"occasionnel" (*casual user* en anglais). Chaque groupe ayant ses caracteristiques et attentes, il est important de distinguer ces différentes cibles car nous considérons que la recherche en informatique graphique, et particulièrement en rendu expressif (qui place le vague concept de l'"artiste" au centre de ses considérations), doit tenir compte de cette pluralité de destinataires. Par essence, le "programmeur" est l'héritier direct des premiers utilisateurs des ordinateurs d'antan. Ayant une connaissance accrûe des aspects techniques de la machine et des modèles mathématiques qu'il manipule pour créer des images, il procède avant tout par exploration exhaustive des techniques mises à sa disposition. Un exemple caractéristique de création de contenu visuel abordée selon l'angle du "programmeur" résume tout ce qui retourne de la création procédurale. Cette dernière consiste en l'écriture de séquences d'opérateurs mathématiques sous forme de programmes dont l'exécution produit du contenu visuel, depuis la géométrie jusqu'aux images. Par opposition, un utilisateur plutôt "artiste" aura plus tendance à faire abstraction des contraintes que lui impose l'ordinateur et cherchera à gommer l'intervention de celui-ci dans son processus de création, utilisant une palette graphique, éventuellement rétro-éclairée, plutôt qu'une simple souris et ainsi il se rapprochera au plus près de la métaphore du tracé au crayon sur du papier. Contrairement au "programmeur", le champ

d'expertise de l'"artiste" ne se situe aucunement sur un terrain technique, mais réside dans sa maîtrise des média artistiques traditionnels et son sens aigu de l'esthétique. Ainsi, bien que loin de rechigner à expérimenter l'ensemble des possibilités offertes par les outils logiciels mis à sa dispotion, l'"artiste" ne pourra *a priori* exploiter, étendre, voire détourner ceux-ci à l'instar du "programmeur". Enfin, l'utilisateur "occasionnel" est, certainement en termes de nombre d'individus, la cible la plus importante, et la plus novice quant à l'usage de l'ordinateur. Apparue depuis la démocratisation des ordinateurs et séduite par l'offre de plus en plus importante de logiciels gratuits disponibles sur Internet, elle est avant tout animée par une volonté d'expérimentation des possibilités visuelles de ces machines. Ainsi, l'utilisateur "occasionnel" n'a pour ainsi dire peu ou pas expertise précise, qu'elle fût d'ordre technique ou artistique, et n'a que peu de patience face à un éventuel apprentissage. C'est pour cette raison que des logiciels de création disponibles en ligne tels *Sketch'up* de *Google* ou encore *LiveBrush* ont à redoubler d'ingéniosité pour réduire au maximum leur complexité d'utilisation et de compréhension, et ainsi parvenir à capter l'attention de ces mêmes utilisateurs.

En tenant compte d'une telle variété d'utilisateurs, il faudra porter une attention toute particulière aux modes d'interaction que les techniques proposées auront à offrir à ces derniers. Il faut distinguer deux grandes approches de création de contenu visuel assistée par ordinateur : une approche *ex nihilo* (ou dite *"from scratch"* en anglais), et une approche qualifiée d'approche par l'exemple. L'approche *ex nihilo* correspond aux cas où l'utilisateur produit du contenu visuel uniquement à partir des outils logiciels à sa disposition. Une fois de plus, les *"virtual canvases"* tels *Adobe Photoshop* ou encore *Corel Painter*, au plus proche de la métaphore de la peinture sur toile, représentent la quintessence de ce type d'approche. Ainsi, cette dernière est particulièrement adaptée et appréciée des utilisateurs experts ("artistes" et "programmeurs") car bien que difficile à apprendre et à maîtriser, elle leur permet un contrôle extrêmement fin tout au long de la création de leurs œuvres. Par opposition, les approches par l'exemple regroupent toutes les techniques qui se basent sur le traitement de données d'entrée fournies soit en vue de leur édition, soit en vue de la création de nouvelles données donnant l'illusion d'être issues du même processus génératif. De telles méthodes sont bien plus accessibles aux utilisateurs "occasionnels" même si leur contrôle indirect via l'exemple d'entrée soit plus délicat. Parvenir à proposer des méthodes par l'exemple susceptibles de convenir au plus grand nombre possible est donc un enjeu particulièrement intéressant abordé dans cette dissertation.

Nous nous plaçons ici dans le cadre du rendu expressif, ou encore qualifié de non photo-réaliste. Cette tendance récente en informatique graphique a connu un rapide et constant succès, notamment dans le domaine de l'industrie vidéoludique mais également, bien qu'à moindre échelle, cinématographique. Comme son nom l'indique, le rendu non photo-realiste fait sécession de la quête par la majorité de la recherche en informatique graphique de l'imitation du réalisme photographique et place aux contraire les critères de diligence de communication visuelle ou de qualité esthétique voire artistique au centre de ses considérations. En choisissant d'étudier la question de la création par l'exemple de contenu visuel en rendu expressif nous nous attaquons à la question de la méthode à employer quant à l'analyse des données d'entrée. En faisant ainsi abstraction de la grille de lecture que nous offrait l'hypothèse du photo-réalisme, il nous faut trouver différents angles d'approche pour extraire l'information quant à la signature visuelle des données d'exemple et ainsi trouver les moyens les plus adaptés pour la préserver et l'intégrer à nos résultats.

Les exemples concrets de création par l'exemple que nous abordons dans cette thèse s'intéressent à la plupart des aspects du dessin: les textures, les couleurs, et enfin les lignes. La première problématique est abordée dans la partie I et se consacre à la re-synthèse de textures représentées sous la forme d'images *raster* (grilles de pixels) ou d'arrangements de motifs vectoriels. Un exemple de texture est donc fourni en entrée de notre algorithme et c'est à ce dernier de parvenir à extraire suffisamment d'informations à partir de cet échantillon (souvent petit) en vue d'en établir la meilleure caractérisation visuelle possible et parvenir à créer de nouvelles textures qui, sans présenter de répétitions flagrantes, doivent donner l'illusion de provenir de la même source que l'exemple fourni en entrée. La partie II de ma thèse est consacrée à un autre aspect de la création par l'exemple, à savoir l'édition d'images, et plus précisément leur désaturation. Ici, l'enjeu réside dans le fait de trouver un juste équilibre entre la préservation des contrastes d'origine purement chromatiques visibles dans l'image avant décolorisation et le respect des stimuli achromatiques naturellement imposés par les couleurs de cette dernière. En effet, une approche se focalisant exclusivement sur la reproduction exacte

par les valeurs de gris du résultat des différences de couleurs de l'exemple aurait pour résultat de distordre la gamme dynamique de celui-ci au point qu'il deviendrait dur de percevoir le résultat comme une version décolorisée intuitive de l'exemple. Enfin, la question du rendu de géométries 3d par des dessins au trait est traitée dans la partie III. Ici, l'objectif est de permettre à l'utilisateur d'initier le dessin d'une forme géométrique représentée sous forme de maillage et de laisser la machine analyser, décoder, et anticiper le reste de son dessin. Toute la difficulté réside alors dans le caractère interactif de notre méthode et l'automatisation par la machine de l'identification des caractéristiques géométriques que l'utilisateur est susceptible de vouloir faire ressortir via les courbes qu'il trace.

Ces différentes problématiques permettant une exploration relativement exhaustive du champ d'action de la création par l'exemple en rendu expressif, nous amène naturellement à la question des sources potentielles d'information mises à notre disposition pour analyser nos données d'entrées (textures, distributions de couleurs, géométries 3d). Nous en distinguerons trois, parfois complémentaires, parfois concurrentes, mais chacune influençant immanquablement les propriétés de toute méthode se basant sur elle :

- **L'analyse pure de l'échantillon** L'analyse "pure" de l'exemple correspond au cas où ce dernier est considéré de manière isolée. Ainsi, il revient à la méthode de réorganiser la masse finie d'informations qu'il contient afin d'en proposer une représentation de plus haut-niveau qui se prêtera plus volontiers à la création de nouveaux visuels. Une telle approche est un véritable défi car une analyse pure se doit de ne formuler aucun *a priori* quant à la représentation de l'échantillon qui dès lors lui apparaît comme une masse brute d'informations à explorer et réordonner. Un tel défi relevé confère directement de grandes possibilités d'automatisation à la technique recourrant à ce genre d'analyse. Le pendant négatif de cette démarche est naturellement le manque de contrôle direct de l'utilisateur, ce qui peut dans les tâches les plus artistiques éventuellement poser problème.

- **Le recours à une connaissance *a priori*** La connaissance *a priori* regroupe toute hypothèse formulée en amont de l'exécution de la méthode et qui reste constante au cours de celle-ci. Typiquement, elle revêt la forme de modèles organisationnels ou perceptuels, et permet une lecture extrêmement poussée de l'échantillon passé en entrée de la technique. En contrepartie, cette dernière perd dramatiquement en généralité et selon le degré avec lequelle elle repose sur de telles hypothèses, risque d'échouer une fois confrontée à un exemple déviant du modèle prédéterminé.

- **La supervision de l'utilisateur** Enfin, il arrive parfois qu'il soit plus judicieux de recourir à l'utilisateur au cours de l'exécution d'une méthode plutôt que de tenter un automatisation d'emblée. L'intervention de l'utilisateur peut prendre différents aspects depuis l'ajustement de paramètres jusqu'à la correction progressive, et potentiellement interactive de résultats préliminaires. Néanmoins, il ne faudrait pas que l'appel à l'utilisateur se fasse de manière systématique, au moindre choix technique ou à la moindre ambiguïté que rencontrerait la méthode. En effet, si elle veut viser large, la méthode se doit de formuler des requêtes claires, intelligibles et adaptées à son cœur de cible. Il ne faudrait pas ainsi confondre finesse de contrôle avec démission de la machine au profit de l'utilisateur au risque de rendre la méthode aussi ardue d'accès qu'une méthode de création *ex nihilo*.

Ces différentes sources d'extraction de l'information contenue au sein de l'échantillon d'exemple sont au cœur de ma présentation, chacune de ses parties détaillant une application de création par l'exemple en rendu expressif et ainsi une exploration locale du triangle de l'information défini par celles-ci.

Synthèse par l'exemple de textures

Les premières contributions de ma thèse sont consacrées à la synthèse de textures qui représente certainement le contexte d'application par excellence de la création de contenu visuel par l'exemple. Ce choix semble judicieux car, en informatique graphique, les textures sont omniprésentes. En effet, l'idée sousjacente aux textures fut, bien que simple, révolutionnaire : elle consiste à plaquer directement sur la géométrie à l'aide d'images

des détails de matériaux qu'ils seraient trop coûteux voire impossible de modéliser explicitement à la surface de la géométrie d'un modèle 3d. Ainsi, dans la plupart des rendus observés aujourd'hui et qui affichent une finesse extrême, la majorité de l'information visuelle est en réalité portée efficacement par l'utilisation et le rendu des textures. Cette technique astucieuse a depuis été éprouvée et constitue certainement l'une technique absolument incontournable aujourd'hui, à tel point que l'on peut à présent constater un renversement de tendance dans le rapport de force entre la modélisation géométrique et de sa simulation par des textures. En effet, la plus simple des formes géométriques peut être littéralement transfigurée par l'apposition d'une texture. Également en exploitant le canal *alpha* des images (contrôlant la transparence de ses pixels), les *imposteurs* permettent de donner l'illusion de modèles 3d d'une grande finesse alors qu'il ne s'agit en général que d'une collection de quadrilatères savamment texturés.

Mais il convient de définir plus précisément une texture : par opposition à une image au sens général du terme, une texture se distingue par son homogénéité et sa stationnarité statistique. Malgré ce cadre mathématique à base de contraintes sur les statistiques de ses couleurs, il reste très difficile d'identifier explicitement et de formuler de manière algorithmique claire ce qui confère à une texture son caractère visuel. Le paradigme de sa capture non-paramétrique par l'exemple a donc très vite été exploré, bien que les techniques de synthèse procédurale ou la création manuelle continuent d'être utilisées dans les milieux professionnels. Cependant, ces approches sont réservées aux utilisateurs experts. La synthèse par l'exemple prend le contrepied de ce genre d'approches : la création de textures peut dans certains cas s'apparenter à une véritable tâche artistique requérant l'intervention de *texture artists*, mais, généralement, on recourt à son automatisation car la réalisation de textures homogènes de matériaux se révèle fastidieuse.

Déjà si difficiles à définir, les textures sont également d'une infinie variété. Cette variété est souvent conceptualisée comme un *continuum* de textures indexé par le caractère stochastique, ou au contraire régulier de leurs éléments constitutifs. Cette diversité est en effet si importante qu'aucune méthode existante ne peut se prétendre universelle, et la nôtre ne fera naturellement pas exception. Donc, nous nous consacrerons à une sous-catégorie d'échantillons jusqu'alors particulièrement mal capturés par les méthodes précédentes : celle des textures consistant en l'arrangement quelconque de formes individuellement discernables, observées depuis un point de vue frontal.

État de l'art sur la synthèse de textures par l'exemple

Le chapitre 2 est consacré à la présentation la plus complète possible de l'ensemble des techniques de synthèse par l'exemple de textures de type images *raster*. Les techniques de synthèse procédurale n'étant que brièvement présentées car n'entrant pas directement dans le cadre de ma thèse, il conviendra plutôt de se référer à des états de l'art entièrement consacrés à cette discipline [EMP+02]. Ce chapitre prend le parti d'explorer l'évolution du domaine de la synthèse de textures par l'exemple, en commençant par l'explication des fondements mathématiques de la modélisation de texture et des premières techniques de régression statistique sur modèles graphiques [ZWM96, PS00], puis en s'achevant sur l'exploration des nombreuses techniques approximantes communément utilisées aujourd'hui en informatique graphique. Le détail des différentes techniques non-paramétriques de synthèse est notamment organisé en fonction des primitives de base utilisées pour la création de leurs nouvelles textures, et présente ainsi dans l'ordre suivant les méthodes à base de recopie de pixels [EL99, WL00, KEBK05], de patchs [EF01, KSE+03], et enfin de propagation supervisée de pavages [NZvG03, LCT04].

Notre contribution en synthèse de textures *raster*[1]

Le chapitre 3 détaille notre approche pour la synthèse d'arrangements de formes *raster*. Au terme de la présentation de l'ensemble des approches et techniques préalablement décrites, il apparaît clairement qu'aucune de ces méthodes ne s'attèle au cas des textures composées de formes discernables. En effet, la plupart d'entre

[1] Une image de type *raster*, ou encore dite matricielle en français, est une image numérique représentée sous la forme d'un tableaux 2d de points de couleurs, ses pixels.

elles génèrent leurs sorties à partir de primitives n'évoluant pas à une échelle adaptée ou en imposant des contraintes et hypothèses fortes telles que le respect par les éléments de la texture d'une disposition suivant un pavage strict. Traiter les pixels de l'échantillon comme autant de couleurs indépendantes compromet grandement la préservation de structures spatialement larges au sein de la texture créée, et de nombreux mécanismes de contrôle à plus grande échelle, tels le recours à des voisinages hiérarchiques ou des approches basées sur des optimisations globales de fonctions de coût, sont nécessaires pour palier à cet écueil. La manipulation de patchs permet dans une certaine mesure de répondre à la difficulté posée par des textures dont les pixels présentent de fortes corrélations car représentant des formes identifiables, mais elle risque de créer des artéfacts visuels entre patchs aux zones de bordure lorsqu'ils seront transférés au résultat final. Également, une telle approche ne peut constituer une solution ultime car bien que pouvant incorporer des structures parcourant plusieurs pixels, ces patchs ne sont pas adaptés aux formes que la méthode se devrait de capturer afin de décrypter au mieux le texture d'entrée. En revanche, la solution à base d'extraction de structures de treillis s'efforce d'extraire des patchs en tenant compte de leur contenu et vérifie qu'ils sont en véritable adéquation avec l'échantillon. Cependant, l'hypothèse selon laquelle les éléments caractéristiques de la texture d'exemple suive un pavage réduit considérablement le champ d'action de ce type d'approches : la rigidité des placements supposés ainsi que l'absence de tout recouvrement entre les primitives de la texture réservent l'utilisation de telles méthodes à la synthèse et l'édition de textures strictement régulières.

Nous proposons donc de combler le vide laissé autour du cas de textures à base de formes planes sans *a priori* de disposition avec notre technique d'analyse de l'échantillon inspirée par les techniques de détection d'objets plus communément rencontrées en vision par ordinateur. En effet, dès que l'on regarde les textures que nous étudions, tout nous indique intuitivement la marche à suivre : il faut directement manipuler et réarranger les formes qui constituent l'échantillon et ainsi on peut le prolonger plus sûrement que par des ajouts de pixels, de patchs extraits aléatoirement, ou des tuiles d'un hypothétique pavage. Ainsi, nous employons un paradigme qui a fait ses preuves pour la recherche et reconnaissance d'objets au sein d'images afin de les extraire explicitement. Nous procédons par appariemment de voisinages visuellement similaires afin de détecter la présence de répétitions élémentaires au sein de l'échantillon. En effet, en l'absence complète de connaissance *a priori* en accord avec notre volonté d'automatisation, le caractère répétitif des motifs au sein de la texture est notre seul indice pour pouvoir en extraire la signature visuelle. Les voisinages sont donc appariés selon leur proximité en termes de distribution spatiale de directions de gradient encodée par le descripteur local *SIFT* (pour *Scale Invariant Feature Transform*). Cette étape de description est cruciale car elle permet de se placer dans un espace abstrait permettant de rassembler des voisinages visuellement proches tout en évitant la difficulté de traiter explicitement leurs eventuelles translation, rotation, ou dilatation géométrique. Dès lors, toute paire de voisinages définit de manière unique une transformation au sein de l'image (une similitude 2d dans notre cas). Il convient dès lors d'analyser ces mêmes transformations afin de parvenir à l'extraction de formes spatialement cohérentes. En effet, une accumulation locale de transformations comparables indiquera la présence d'un pan de l'image se dupliquant au sein de celle-ci et donc d'un motif à éventuellement extraire. Cette tâche revient à une recherche de modes au sein de la distribution de ces transformations entre voisinages, et donc un partitionnement au sein de l'espace de transformation. Nous utilisons à cette fin l'algorithme de partitionnement par *mean-shift* car il nous permet de détecter un nombre quelconque de classes de transformations. Une fois identifiées, ces transformations dominantes nous permettent de décider s'il convient ou non de s'attarder sur les points d'où elles émergent afin d'y trouver une éventuelle forme à extraire. Cette décision dépend directement de la cardinalité des classes associées à chacune de ces transformations candidates : plus le nombre de paires de voisinages –suivant approximativement une même transformation dans une zone localisée de l'image– est important, plus la probabilité d'être confronté à une forme caractéristique de la texture d'entrée se répétant est élévée. Si cela se justifie, nous procédons alors à l'extraction de cette forme par une expansion de région à partir des voisinages associés aux points de départ des transformations d'une même classe. Cette étape est assurée par l'addition gloutonne de nouveaux pixels à la bordure de la région en cours d'extraction, chacun d'entre eux devant présenter une distance en termes de *SIFT* suffisamment faible une fois projetés selon la transformation de la forme candidate. Au terme de la phase d'extraction de formes, il existe une grande redondance entre ces dernières car les transformations issues de

l'étape de partitionnement ne peuvent être que traitées indépendamment à ce stade de l'algorithme. Ainsi, un même motif sera extrait autant de fois qu'il se duplique au sein de l'image et donnera lieu à des formes alors indépendantes les unes des autres. Ses duplicata génèreront leurs propres formes qui ne seront alors pas encore mises en commun avec les précédentes. Pour remédier à ce problème et créer nos catégories de motifs multi-instanciés, nous étudions la position au sein de l'image de ces formes et encodons leur recouvrement au sein d'une graphe dont les sommets correspondent aux différentes formes alors obtenues, et les arêtes à l'ensemble des transformations susceptibles de générer un recouvrement entre ses deux sommets adjacents. En ne considérant que les arêtes définissant un recouvrement relatif important, il nous suffit d'extraire les composantes connexes de ce même graphe afin de réunir les différentes formes d'un même motif se répétant au sein de l'échantillon. Il est très important au moment de l'analyse du graphe en composantes connexes de s'assurer que tout cycle au sein de celles-ci correspond bien à une composition de transformations qui résulte en la fonction identité. Dans le cas contraire, des incohérences spatiales pourraient alors se produire. La dernière étape de notre analyse de textures consiste en la création d'une *carte d'instances* qui à chaque pixel de l'échantillon, associe l'identifiant de classe et d'instance de l'éventuel motif se situant à son emplacement.

La représentation de la texture d'exemple par sa *carte d'instances* contient l'intégralité de l'information que l'on aurait pu souhaiter extraire de l'échantillon pour produire de nouvelles textures. En effet, grâce à la connaissance de l'emplacement des différentes formes et des transformations qui les unissent, il nous est possible:

- de préserver ces mêmes formes grâce à leur manipulation explicite en cours de resynthèse;
- d'étudier leurs emplacements respectifs et ainsi s'efforcer de capturer un nouvel aspect de la signature visuel de l'échantillon;
- d'analyser les subtiles différences d'apparence entre instances d'un même motif afin de rendre plus naturelle l'apparence des résultats;
- et enfin de reconstruire dans une certaine mesure des motifs partiellement recouverts au sein de l'exemple en rassemblant de manière spatialement cohérente tout pixel apparaissant au moins deux fois dans l'ensemble de ses instances.

Notre contribution en synthèse de textures vectorielles

Au terme de notre analyse de textures *raster*, nous disposons donc d'un ensemble de formes complexes se répétant au sein de l'échantillon regroupées en plusieurs catégories. L'étape suivante est donc l'analyse de leurs placements relatifs afin de guider le processus de resynthèse et on peut la retrouver en détail au sein du chapitre 4. Les travaux de recherche qui y sont présentés ont été effectués en collaboration avec Thomas Hurtut[2] et se placent dans le contexte de l'analyse d'arrangements de formes vectorielles à base de courbes de type *spline*. Ce changement de format des données d'entrée est justifié par le besoin de traiter des données le moins bruitées possible afin d'en fournir une analyse la plus juste possible (un tel degré de précision n'étant pas en l'état systématiquement atteignable avec la méthode précédemment exposée). Malgré tout, la théorie soustendant notre formulation de la capture des positions de formes via modélisation statistique est parfaitement adaptée aux cas des textures vectorielles, comme *raster*.

Par contre, le changement de la nature de nos textures nous force à trouver de nouveaux moyens pour analyser de tels échantillons et regrouper leurs différents éléments vectoriels constitutifs en catégories. Nous pourrons alors disposer d'une information comparable à celle obtenue au terme de la précédente méthode consacrée aux entrées *raster*. Recueillir au sein d'une même catégorie d'apparence les éléments visuellement proches revient à nouveau à un partitionnement de données dans un espace de caractéristiques consciencieusement choisi. S'agissant d'éléments à base de courbes, nous nous inspirons des premières études conduites par Béla Julesz sur la discrimination préattentive de textures constituées de la collection

[2] http://www.math-info.univ-paris5.fr/~hurtut

de formes simples à base de traits [Jul75]. Figurent parmi les données favorisant la différentiation de telles textures, l'orientation, l'élongation, la surface desdites formes, ainsi que le nombre de coins et croisements qu'elles présentent. Nous nous servons donc de ces mêmes données, facilement évaluables sur des courbes paramétriques, afin de créer les vecteurs de caractéristiques de chaque élément et procéder à leur partitionnement en classes par une méthode *a contrario*.

Nous avons dès lors un ensemble de formes réparties en catégories d'apparence et représentées par leur centre de gravité, et notre objectif est donc la capture des règles implicites dictant leurs positions respectives les unes par rapport aux autres. Contrairement aux méthodes existantes qui transposent les techniques de synthèse de textures *raster* à base d'échantillonnage non-paramétrique en troquant le système de voisinage imposé par la grille des pixels par les relations d'adjacence entre éléments obtenues grâce à la triangulation de Delaunay de leur centre [BBT+06, IMIM08], nous décidons d'explorer la piste de la modélisation statistique. En plus de son contrôle global à l'échelle de l'intégralité de l'échantillon, une telle approche a l'avantage d'assurer la capture d'un spectre de distributions bien plus vaste. Bien que le choix d'un modèle statistique impose une connaissance *a priori* sur l'analyse des échantillons d'entrée, cette direction ne compromet pas notre volonté initiale d'automatisation (que ce soit pour l'analyse du placement de formes ou sa regénération). Mais la principale difficulté reste la gestion des difficultés que pose la taille réduite de nos données d'entrée (dessinées par l'utilisateur en amont de l'analyse) sur l'établissement de statistiques fiables et la pertinence de toute régression statistique. Nous choisissons pour cela une version simplifiée du processus ponctuel dit *Strauss hardcore* qui, bien que simple, permet la capture d'intéractions d'ordre 1 et 2 entre éléments au sein et entre catégories, et une estimation en forme close de ces paramètres. Une fois les degrés de liberté de notre modèle ajustés à notre échantillon, il est facile de créer de nouveaux arrangements de formes en utilisant la dérivée de Radon-Nickodym associée à notre modèle comme énergie à maximiser via échantillonnage de Monte Carlo par chaîne de Markov.

Conversion d'images couleur en niveaux de gris

La partie II de ma thèse change de domaine d'application, mais n'en reste pas moins intimement liée à la question de la création par l'exemple. En effet, bien que ne s'agissant pas de resynthèse, l'édition d'images constitue également une facette à part entière de la thématique abordée ici. Elle se distingue par le fait que la "marge de manœuvre" de la méthode est bien plus contrainte car il ne s'agit pas réellement de génération de nouveaux visuels mais plutôt de l'identification et de la mise en valeur de l'information existante.

Nous choisissons ici la question de la conversion d'images couleur en niveaux de gris. Par essence, la désaturation d'images confine à une réduction de dimensionalité de leur *gamut*, chaque pixel constitué de l'association de trois canaux de couleur potentiellement indépendants devant se voir attribuer une unique valeur de luminosité. Ainsi, la perte d'informations visuelles paraît inévitable et en effet, l'utilisation d'une formule de conversion irréfléchie aura tôt fait d'assigner une même valeur de gris à deux couleurs pourtant distinctes au sein de l'image d'entrée. Le rôle d'une désaturation "intelligente" est donc de prévenir et minimiser cette perte d'informations. Loin de constituer une thématique dépassée, cette problématique continue à être régulièrement abordée et cela malgré la démocratisation de périphériques de sortie couleur. Cependant, pour des considérations de coût d'impression ou d'esthétique (malgré la suprématie des appareils photos couleur depuis 1964, la photographie noir et blanc est une pratique artistique appréciée des artistes et du public car elle laisse plus de place à l'imagination et donne toute son importance aux jeux de lumière et à la composition), la question de la désaturation d'images continue de susciter beaucoup de recherche en traitement de l'image.

État de l'art sur la désaturation d'images

Le chapitre 5 propose un aperçu de l'ensemble des techniques précédant la publication de notre algorithme, mais également certaines méthodes lui succédant. La volonté de préserver l'information de nature purement chromatique après conversion pose la question de la définition de celle-ci : selon la nature de l'information

à préserver, l'objectif d'une méthode de conservation d'images en niveaux de gris diffèrera diamétralement. Si l'on considère que l'information à préserver réside dans les contrastes chromatiques de l'image d'entrée, alors la méthode visera à maintenir la "discrimabilité" entre couleurs d'origine au sein de l'image convertie. Si au contraire l'on cherche à préserver la sensation visuelle que suscite en termes de brillance les couleurs de l'image, alors la méthode s'évertuera à préserver l'exactitude perceptuelle de son résultat. Ces objectifs sont clairement conflictuels et il faut les équilibrer pour obtenir les meilleurs résultats possibles en termes d'exactitude mais également d'informativité.

En consultant les documents consacrés à la désaturation d'images adaptée à leur contenu chromatique, immédiatement, ce qui saute aux yeux est l'adhésion générale de la plupart des chercheurs à la préservation des contrastes plutôt qu'à la fidélité perceptuelle. Donc, sont apparues de nombreuses méthodes qui s'évertuaient à contraindre l'assignement de la valuer des pixels pour qu'ils continuent à assurer les mêmes contrastes que ceux de l'image d'origine après passage en niveaux de gris. Ces méthodes sont extrêmement coûteuses car elles font appel à des mécanismes d'optimisation globale impliquant la résolution de systèmes de contraintes entre paires de pixels [GOTG05]. Des approximations à base de décomposition hiérarchique [MMS06] ou des simplications n'incorporant que des contraintes entre paires de couleurs [RGW05] ont également été proposées, mais tout échantillonnage spatial ou en espace de couleurs dégrade la qualité des résultats. La complexité de ces méthodes est telle que cela les rend aléatoires ou imprévisibles et l'influence de leurs paramètres est difficile à estimer [GOTG05, GD07]. Cela est également vrai pour les méthodes à base de reconstruction d'images par la réintégration de leur champ corrigé de gradients [NCN07]. Enfin avant la publication de notre démarche, aucune technique ne pouvait être opérationnelle dans le traitement de séquences animées en raison de leurs temps de calculs prohibitifs ou de leur recours à des techniques d'échantillonnage à la cohérence temporelle difficile à obtenir.

Bien moins visibles que leurs consœurs visant au maintien de la discriminabilité des contrastes de l'images, les techniques attachées à la persistance des sensations de brillance causées par les couleurs de l'image originale n'étaient à notre connaissance qu'au nombre de deux : elles furent réalisées par la recherche de Raja Bala du groupe *Xerox Corporation* [BB03, BE04]. Mais ces dernières sont entravées par la nécessité de la discrétisation de la distribution des couleurs pour imposer un ordre absolu de luminosité entre couleurs [BB03], ou par des artéfacts résultant du recours à un filtre passe-haut qui renforce les arêtes chromatiques de l'image dans le résultat [BE04], contraintes qui rendent inopérante l'application de telles techniques à des images complexes au contenu fréquentiel disparate et aux distributions de couleurs complexes.

Notre contribution en désaturation d'images

En collaboration avec Kaleigh Smith[3], notre démarche pour la désaturation d'images est présentée au chapitre 6 et elle se démarque de l'ensemble de la littérature du domaine car elle se consacre principalement au respect des caractéristiques perceptuelles de nos résultats par rapport à l'image originale. Mais il convient surtout de préserver l'information chromatique et donc d'assurer la désambiguation entre couleurs perceptuellement distinctes qui se verraient assigner un même niveau de gris au terme d'une conversion hâtive. Notre approche est originale car elle choisit la simplicité maximale et elle se place ainsi en porte à faux vis-à-vis des autres méthodes à base d'optimisation globale et projection au sein d'espaces de couleurs adaptés au contenu de l'image d'entrée. Cette volonté s'explique non seulement par un soucis de clarté théorique, mais également par un soucis d'interactivité des temps de calculs. En effet, il serait peu judicieux, dans ce cas précis, de proposer une méthode entièrement automatique : car si elle est choisie en raison d'un projet artistique, l'utilisateur doit pouvoir contrôler l'influence visuelle de la technique de conversion pour éviter que l'image originale ne soit dénaturée. Pour des raisons relatives aux conditions de présentation (affichage, projection, résolution), naturellement inconnues de notre algorithme, des mécanismes de contrôle paraissent indispensables.

Il nous semble primordial d'assurer la fidélité perceptuelle de nos résultats par rapport à l'image originale. C'est une erreur de limiter le succès d'une technique de désaturation à sa capacité à restituer les contrastes de

[3] http://www.cs.mcgill.ca/~kaleigh

couleurs, quite à distordre la gamme dynamique de l'image en termes de luminosité ou inverser les polarités naturelles de ses gradients achromatiques. Peut-on véritablement considérer comme satisfaisante une image en niveaux de gris qui, certes, attribue une valeur de gris à chaque couleur distincte présente au sein de l'image d'entrée, mais qui perceptuellement semble émaner d'une scène qui n'a plus rien en commun avec l'original ? Cette contrainte d'acuïté perceptuelle limite les traitements que nous pouvons nous permettre, l'ensemble de nos éditions devant restaurer au mieux les contrastes chromatiques perdus tout en étant le moins intrusif possible et en conservant la gamme dynamique et la polarité des gradients de l'image couleur. Afin d'obtenir une influence maximale avec un minimum d'altérations, nous avons décidé de recourir exclusivement à des traitements travaillant de pair avec ceux du système visuel humain, en exploitant avant tout effets entoptiques et illusions d'optique. Dans ce but, nous avons dû nous inspirer de nombreux travaux et conclusions traitant de la perception appliquée – ces travaux ayant souvent été ignorés en traitement d'image. Le recours à cette connaissance extérieure revient donc à l'incorporation d'*a priori* dont l'utilité est directement liée à la complexité des protocoles expérimentaux, des modèles, ainsi que de la qualité des mesures statistiques que ces expériences en colorimétrie ont établis.

Notre algorithme se déroule en deux étapes, chacune d'elle devant assurer au mieux le respect des objectifs conflictuels identifiés plus haut. La première étape a pour rôle un assignement préliminaire global des valeurs de niveaux de gris et elle est censée garantir la qualité en termes de fidélité perceptuelle de nos résultats. La seconde étape, à base de manipulation locale de leurs contrastes, se doit de restituer les contrastes de nature exclusivement chromatiques perdus au terme de l'étape précédente. Toutes deux œuvrent en harmonie avec les mécanismes de notre système visuel et incorporent des phénomènes optiques précis afin de ne pas dénaturer la sensation que procure l'image d'origine. La phase d'attribution des valeurs de gris associe à chaque pixel de l'image colorée une valeur préliminaire de luminosité et prend en compte le phénomène dit d'Helmholtz-Kohlraush (à clarté (*lightness*) égale, une couleur plus saturée est perçue comme plus éclatante à une majorité d'observateurs). Il s'avère que la modélisation classique de la perception achromatique d'une couleur (sa brillance perçue, ou encore *perceived brightness*) par sa clarté n'est pas satisfaisante, car elle ne tient nullement compte de l'influence de la couleur, de sa teinte et saturation, sur celle-ci. Au terme de nos recherches, nous avons étudié trois principaux modèles de brillance perçue incluant de telles considérations : le modèle de Fairchild et Pirotta noté L^{**} [FP91], ainsi que les deux modèles de Nayatani L^*_{VAC} et L^*_{VCC} [Nay97]. En tenant compte de la finesse mathématique de ces différents modèles ainsi que de la nature de leur protocole expérimental qui conditionne directement leur domaine d'application, nous avons sélectionné le modèle L^*_{VAC} qui nous permet d'associer à chaque couleur la valeur de gris correspondant à sa brillance perçue qui est perceptuellement bien plus exacte que le recours direct à sa clarté L^* ou toute valeur uniquement dépendante de sa composante trichromatique Y. La seconde étape de notre technique est, en fait, une manipulation extrêmement localisée des contrastes de l'image achromatique précédemment obtenue. Du point de vue de l'information visuelle présente au sein de l'image originale, nous pensons qu'il est moins grave de perdre des contrastes chromatiques impliquant uniquement des couleurs distantes, et qu'il est primordial d'assurer la préservation de contrastes entre couleurs spatialement adjacentes. Ainsi, l'édition locale de contrastes paraît constituer une approche appropriée, d'autant plus que notre vision est particulièrement sensible aux contrastes et qu'il a été expérimentalement établi que notre sensation de qualité d'images est intimement liée à la netteté de ses contrastes [LGK06]. La force de l'influence des contrastes sur nos sensations visuelles trouve son explication dans l'hétérogénéité spatiale des réponses des cellules ganglionnaires de la rétine qui jouent un rôle essentiel dans notre perception du monde extérieur. Cette influence est si puissante qu'elle génère de nombreuses illusions d'optique à base de manipulation de contrastes très efficaces alors que les stimuli utilisés sont extrêmement simples. Nous tirons parti de l'une d'entre elles, l'illusion dite de Craik-O'Brien-Cornsweet qui se caractérise par une propagation de brillance illusoire de part et d'autre d'une perturbation curviligne de contraste à condition que celle-ci présente une discontinuité de pôlarité opposée et d'une décroissance exponentielle [Cra66, O'B59, Cor70]. Ainsi, une telle arête, que nous qualifions de contour de Cornsweet, traversant une zone de luminance constante créera l'illusion de correspondre à une démarcation entre deux zones de luminosité distincte et constante. Il faut noter que de tels contours sont très aisément introduits au sein d'images via l'utilisation du filtre du masque flou (*unsharp masking*) fréquemment utilisé pour améliorer le piqué de

photographies. La seconde phase de notre algorithme consiste donc en l'introduction de tels contours au sein de nos images achromatiques intermédiaires et dans le contrôle de la brillance illusoire qu'ils génèrent pour restituer les contrastes chromatiques locaux perdus. Nous utilisons à cette fin une version multi-échelle du masque flou que nous rendons adaptatif pour que la force de cette perturbation de contrastes soit directement proportionnelle à la force des contrastes chromatiques perdus. Cette dernière altération de nos résultats peut être interactivement contrôlée par l'utilisateur via un ensemble réduit et intuitif de paramètres qui permettent de limiter artéfacts et de spécifier son étendue spatiale.

Cette méthode, d'une étonnante simplicité, possède, cependant, de nombreuses propriétés intéressantes :

- la fidélité perceptuelle de ses résultats,
- la pertinence de l'ordre en termes de brillance qu'elle impose à l'ensemble des couleurs de l'image,
- sa désambiguation de couleurs isoluminantes,
- et enfin sa facilité de transposition aux cas de séquences animées.

Une étude indépendante conduite l'année suivant la publication de nos travaux, atteste la qualité de nos résultats, et place notre approche au second rang, après la méthode de Grundland *et al.*, et après confrontation avec les principales techniques disponibles alors [Čad08a].

Rendu au trait de géométries 3d par l'exemple

Avant ma conclusion, la partie III de ma dissertation s'intéresse à la ligne, au dessin au trait, et surtout au rendu de géométries 3d simulant ce genre d'illustrations. La question de la communication de formes potentiellement complexes au travers d'une sélection limitée de courbes savamment choisies est une problématique simple et complexe à la fois. L'expression de concepts à l'aide de dessins exécutés grâce à quelques lignes constitue en effet un mode de communication d'une étonnante efficacité, au point d'être correctement interprétée chez certains primates. Par contre, les modalités d'exécution mises en œuvre par nos mécanismes intellectuels pour réaliser, mais également déchiffrer ces mêmes dessins sont très complexes. Entièrement automatiser cette tâche semble hors de notre portée : pour préserver le caractère artistique et esthétique de telles représentations, nous avons décidé de privilégier une exploitation de l'information obtenue principalement par les choix interactifs entre l'utilisateur et l'outil informatique, et de mettre en arrière-plan toute velléité d'automatisation ou de recours à une connaissance *a priori* qui peuvent s'avérer réductrices.

État de l'art sur la création de dessins au trait de géométries 3d

L'état-de-l'art relatif aux thématiques de l'étude des processus cognitifs de nos aptitudes mentales à visualiser des formes complexes à partir d'un ensemble réduit de courbes, ainsi que de l'extraction de lignes pertinentes à la surface de maillages 3d est détaillé au sein du chapitre 7 du présent document.

Nous ignorons encore quelle est l'explication qui peut être proposée en ce qui concerne notre facilité à analyser, reconnaître et reconstruire des formes complexes à partir de croquis. La tolérance de nos inférences mentales face aux manque de fidélité ou de précision desdits croquis par rapport à la réalité géométrique qu'ils représentent impressionne également, et elle assure dès lors une grande liberté stylistique aux artistes. Actuellement encore, aucune conclusion définitive n'a été trouvée pour révéler les détails de notre aptitude à analyser et reconstruire les formes à partir de croquis même maladroitement esquissés. De nombreuses expériences en psycho-cognition isolent et analysent l'impact qu'ont certains signaux visuels sur notre compréhension de l'image rétinienne que nous renvoie notre environnement. Parmi eux, les indices de discontinuités de profondeur, d'illumination, ou encore la manière dont la lumière est réfléchie par la surface des objets jouent naturellement un rôle capital. Ainsi, il s'avère que notre perception de profondeur (*pictorial relief*) peut être véhiculée avec sensiblement autant de précision (bien que toujours prompte à erreur) via un

dessin composé simplement par les silhouettes, contours internes, ainsi que quelques lignes soigneusement choisies pour souligner les aspérités de leur surface [KvDK92].

Parmi les premières expérimentations en rendu expressif, on trouve la recherche de s'approcher de la notion de "réalisme fonctionnel" en opposition aux réalismes physique et photo-réaliste [Fer03]. En effet, il est surprenant de constater notre instinct à recourir à des représentations simplifiées (schémas ou des dessins au trait) plutôt qu'à des photographies pour désigner tâches à effectuer ou pour expliciter des formes complexes. En effet, les dessins au trait se prêtent remarquablement bien à l'abstraction de formes et rendent visuellement compréhensible le rendu de toute géométrie. La communauté des chercheurs en rendu non photo-réaliste a exploré de nombreuses possibilités pour permettre la génération automatique de rendus à base de trait à partir de maillages 3d et ont notamment dégagé un nombre impressionnant de règles pour la détection de lignes à leur surface, des règles qui sont donc d'autant de types de trait possibles pour la création du dessin final. Parmi les données spécifiques pour l'extraction de ces lignes, les données surfaciques de nature purement géométrique sont évidemment primordiales comme les normales ou la courbure, notamment pour la localisation de courbes telles que les crêtes et vallées, courbes d'une grande pertinence pour la décomposition et communication de formes [HRP+84], les "courbes de démarcation" (*demarcating curves*) [KST08], ou encore les "courbes de relief" [KST09]. Bien que soulignant des caractéristiques de surfaces convenables d'un point de vue purement descriptif, les lignes exclusivement liées à la recherche de comportements géométriques précis ne garantissent pas nécessairement l'obtention de dessins réussis. La prise en compte des conditions d'observation est tout aussi importante : silhouettes et contours sont en effet des lignes indispensables pour détacher tout objet de son environnement [Koe84], et l'inclusion de lignes dont l'évolution sur la surface paraît plus naturelle que les crêtes et vallées (qui, strictement attachées à la surface, semblent dessinées sur elle) est essentielle pour approcher le travail des artistes. Ainsi, ont été proposés les "contours suggestifs" (*suggestive contours and highlights*) [DFRS03, DR07] se basant sur la courbure radiale en chaque point de la surface pour étendre et anticiper naturellement les contours des formes, et les "crêtes apparentes" (*apparent ridges*) [JDA07] qui prennent en compte les effets de la projection perspective de la caméra sur notre perception de la courbure des surfaces. Enfin, de nouvelles lignes peuvent également être déduites de l'étude du comportement local de la fonction d'illumination sur la surface de l'objet à l'instar des "courbes photo-extrêmes" (*photic extremum curves*) [XHT+07], ou plus récemment des "lignes laplaciennes" (*Laplacian lines*) [ZHXC09].

Avancement de notre génération interactive de rendus au trait de géométries

Nous venons d'évoquer des techniques évoluant directement à la surface des objets 3d, mais il faut cependant préciser que des approches 2d se basant sur la projection à l'écran des informations de géométrie existent également. Néanmois, il est déjà possible d'évaluer l'abondance des différentes techniques disponibles, et de noter qu'une telle profusion de méthodes et d'approches (souvent difficilement comparables sur le plan théorique) est une véritable entrave à leur utilisation effective. À tel point, que de récentes recherches menées par Cole *et al.* ont récemment été publiées et se focalisent exclusivement sur la question de la validation comparative de ces différentes techniques, mais également de leur performance en termes de communication de forme ainsi que de la confrontation de leurs résultats avec les travaux d'artistes [CGL+08, CSD+09].

Les travaux présentés au chapitre 8 accompagnent cette problématique et tentent d'y répondre en affirmant que pour être véritablement efficace une méthode de génération de dessins au trait à partir de maillages 3d ne peut se permettre d'être automatique, et que le recours à toute connaissance *a priori* (prenant ici la forme de la recherche de motifs prédeterminés du comportement différentiel des surfaces) n'est nullement souhaitable. Au contraire, en plus de sa lisibilité, la dimension esthétique du rendu au trait joue un rôle déterminant, et il convient à l'utilisateur de guider le processus de création. Ainsi, nous proposons une approche par l'exemple, susceptible de remédier à l'hésitation produite par une offre pléthorique de solutions techniques en replaçant l'utilisateur au cœur de notre approche de création interactive.

Bien que très encourageants, les résultats détaillés ici restent une recherche en cours menée en étroite collaboration avec Amit Shesh[4]. Ils sont la preuve expérimentale de la validité de notre approche et de la mod-

[4] http://www.itk.ilstu.edu/faculty/ashesh

élisation de la solution proposée. En effet, afin de permettre à l'utilisateur de prendre part à l'élaboration du rendu, nous réduisons le problème de la localisation de lignes en accord avec le style de celui-ci à un problème de classification binaire : au vu des courbes déjà tracées ou approuvées par l'utilisateur, est-il judicieux d'accepter ou de rejeter une nouvelle courbe candidate ? Cet angle d'attaque n'est pas sans rappeler l'analyse *a posterio* des dessins proposés par les artistes prenant part à l'étude-utilisateur de Cole *et al.* [CGL+08] ou évidemment le système de sélection par l'exemple de Lum et Ma [LM05]. Notre principale contribution réside dans le caractère interactif de notre technique, ainsi que dans le fait que nous ne visons pas uniquement la sélection de lignes prédéterminées existantes, mais la génération de nouvelles lignes candidates et donc l'extraction de nouvelles courbes en accord avec la représentation computationnelle qu'a notre système du style de l'utilisateur.

Il convient dès lors de décrire au mieux le comportement de la géométrie que sont supposées rehausser les différentes courbes tracées à l'écran par l'utilisateur pour un point de vue fixe, et ainsi de nous placer dans un espace de caractéristiques adéquat pour procéder à l'étape de classification. La qualité de la signature géométrique des courbes est donc essentielle pour assurer une classification pertinente. Elle comprend ici la plupart des données géométriques communément étudiées pour le calcul des techniques déjà citées de détection automatique de lignes. Parmi elles, l'inclinaison et l'illumination diffuse de la surface ainsi que leurs dérivées secondes, la profondeur perspective par rapport à l'observateur, ainsi que les courbures principales, leurs directions associées, mais également les courbures radiale et apparente. Pour faciliter la mise en correspondance entre les courbes tracées à l'écran avec les données géométriques 3d, ces dernières sont projetées à l'écran, stockées et accédées via des *geometric buffers*[5] selon l'échantillonnage dicté par le chemin des courbes à l'écran.

Une fois le calcul de la signature géométrique de chaque courbe réalisé, il sera possible, alors, d'effectuer une classification binaire dans l'espace de caractéristiques correspondant. Cette dernière (prenant en considération le fait que telle ou telle courbe ait été dessinée, sélectionnée, ou au contraire précédemment supprimée par l'utilisateur) parviendra à inférer la pertinence de certaines régions de cet espace vis-à-vis du style de dessin de l'utilisateur. Nous avons recours à la technique d'apprentissage automatique *AdaBoost* ou d'*adaptive boosting* qui compte parmi ses nombreuses qualités, l'aptitude de créer des classifieurs dits forts (fortement corrélés avec la distribution des observations positives) à partir de combinaisons linéaires de classifieurs nettement plus faibles, sans tendance au surapprentissage [FS95]. Au terme de la phase de classification, il est alors possible d'obtenir une carte de prédiction, à partir d'un point de vue quelconque, en associant à chaque pixel de l'écran le score que donne le classifieur préalablement obtenu pour les données géométriques se projectant sur lui. À chaque pixel est alors assignée une valeur entre 0 et 1 qui peut intuitivement être interprêtée comme la probabilité qu'une ligne en adéquation avec le style de l'utilisateur passe par lui.

Cependant, l'évaluation de cette carte de probabilités reste coûteuse et pourrait compromettre notre objectif d'interactivité. Nous proposons donc une solution approchante qui consiste à échantillonner pour un nombre plus réduit de points la fonction de probabilité implicitement définie par le classifieur ajusté aux observations, et à obtenir de nouvelles courbes à partir de certaines arêtes de la triangulation de Delaunay de ce même ensemble de points au travers desquelles nous faisons passer des séquences de courbes de Bézier. Ces nouvelles lignes correspondront à l'anticipation par la machine du dessin de l'utilisateur et ce dernier, par ses corrections, éditions, et/ou ajouts, s'intègrera activement à notre système de rendu au trait par l'exemple.

Conclusions générales

Voici, les trois principales contributions de mon travail, liées à l'extrême variété de la création de contenus visuels en rendu expressif, évaluées au prisme de la synthèse de textures, de la désaturation intelligente d'images et à la génération interactive de dessins au trait de géométries. Ces différentes thématiques nous ont permis une large exploration du triangle des possibles sources d'information à la disposition de toute méthode dite

[5] Les *geometric buffers*, ou plus simplement *G-buffers*, sont des tableaux 2d dont chaque entrée correspond à un pixel de l'écran et archive les données géométriques associées à la surface se projetant à leur position.

par l'exemple (triangle dont les sommets correspondent à l'analyse "pure" de l'échantillon, l'emploi de modèles prédéterminés, et le recours au contrôle de l'utilisateur).

Les conclusions et discussions présentées au chapitre 9 sont de deux ordres. Évidemment, chaque étape des différents algorithmes proposés peut être avantageusement substituée par des procédures plus élaborées (via des techniques plus complexes de partitionnement de données ou encore le remplacement des approches gloutonnes par des optimisations globales d'énergie) faisant appel à des caractéristiques descriptives plus récentes. Mais des perspectives de plus haut niveau sont également envisageables, notamment entraînant le déplacement au sein du triangle d'information de nos différentes approches. En effet, chacune de nos contributions sont extrêmement "typées" si l'on considère leur mode d'intéraction et leur angle d'analyse des données d'entrée. Visant la complète automatisation, soit par le recours exclusif à l'analyse pure de l'échantillon (synthèse de textures *raster*) ou de modèles statistiques (synthèse de textures vectorielles), il serait *a posteriori* désirable de permettre à l'utilisateur de corriger facilement le processus de décomposition de l'échantillon en formes pour permettre les meilleurs résultats possibles. Notre algorithme de désaturation d'images, à présent au point d'équilibre entre les trois sources d'information, gagnerait à se diriger vers plus d'analyse pure du contenu des images qu'il traite, notamment pour palier à sa grande limitation de ne pouvoir réintroduire que les contrastes entre couleurs directement adjacentes. En raison de son état de recherche en cours, il est plus ardu de déterminer à quel changement de philosophie le dernier projet, entièrement tourné vers l'interaction homme-machine, devrait procéder. Actuellement, parvenir à générer de nouvelles lignes visuellement plus plaisantes est le plus important. Dans un second temps et au terme d'une étude-utilisateur, il sera intéressant de déterminer si le recours à la seule direction de l'utilisateur pour la création assistée de dessins au trait est véritablement judicieux.

La plus grande partie de mon travail se termine par une série de discussions qui sont en réalité autant d'interrogations sur le rôle de l'utilisateur, l'adéquation de la recherche en rendu expressif vis-à-vis de la communauté des artistes professionnels, et de l'extension naturelle de la création par l'exemple à la capture non-paramétrique de styles de dessin.

Présentation des annexes

Afin de compléter la lecture des parties contributives du document, il a été ajouté deux annexes qui doivent introduire l'ensemble des concepts nécessaires à la parfaite compréhension des techniques existantes mentionnées aux sein de ses différents états-de-l'art, ainsi que des mesures perceptuelles et quantités mathématiques employées dans notre démonstration.

Introduction aux concepts de colorimétrie et aux différents espaces de couleur

L'annexe A voudrait introduire l'intégralité des notions perceptuelles et colorimétriques indispensables pour comprendre le processus de formation de nos sensations visuelles liées aux couleurs ainsi que de leur modélisation. Les espaces de couleur se répartissent en deux grandes catégories distinctes que sont les espaces de couleur à base de mélange de couleurs primaires, et les espaces d'apparence de couleur.

Les premiers figurent parmi les premières tentatives de formulation mathématique de la mesure de l'ensemble des couleurs visibles par l'oeil humain, et ont été établis par l'appariement de couleurs par un ensemble d'observateurs. Cet appariement de couleurs est censé permettre la quantification des réponses chromatiques spectrales de l'oeil humain et est établi par ajustement de l'énergie radiante de sources lumineuses primaires monochromatiques jusqu'à ce que leur mélange produise la même sensation visuelle que la source lumineuse que l'on cherche à décrire. Les principaux espaces de couleurs de la Commission Internationale de l'Éclairage (CIE), notamment l'espace de couleurs CIE 1931 XYZ trouvent leur base théorique dans l'analyse des résultats ainsi collectés, et leur champ d'action est donc directement lié au protocole expérimental et à l'analyse statistique employés.

Alors que les espaces de couleurs additifs sont avant tout adaptés à la question de l'égalité perceptuelle entre deux couleurs lumineuses, les espaces d'apparence de couleurs se consacrent davantage à la description

perceptuelle de couleurs physiques (par opposition aux sources lumineuses) et à l'établissement de cartes de couleur. Ce dernier point est essentiel car les propriétés de telles cartes conditionnent la pertinence de l'utilisation de leur représentation associée des couleurs pour le traitement numérique des images. Il a en effet été expérimentalement établi que le diagramme chromatique de l'espace CIE 1931 XYZ ne pouvait être considéré comme une carte exploitable tant l'impact perceptuel de la perturbation d'une des composantes trichromatiques est corrélée de manière non prévisible à la position des couleurs au sein de cet espace. La question de l'uniformité perceptuelle des espaces de couleurs est une question primordiale pour que l'édition de couleurs en leur sein soit contrôlable en termes d'impact perceptuel, et lorsqu'elle est garantie, elle assure qu'une perturbation des composantes de couleur aura une influence directement proportionnelle à l'amplitude de la ladite perturbation. Principalement établis par l'observation simultanée de plusieurs plaquettes colorées pour ordonner et, avant tout, pour établir des distances perceptuellement significatives entre elles, les espaces d'apparence de couleurs sont dits relatifs car dépendants des conditions d'éclairage sous lequel de telles expériences sont conduites. Figurent parmi les principaux espaces visant à la garantie de l'uniformité perceptuelle de leurs axes les espaces de Munsell, CIE 1976 Luv et CIE 1976 Lab. L'espace Coloroid va plus loin en assurant l'uniformité esthétique de ses composantes. L'uniformité esthétique stipule que, lorsque considérées dans leur intégralité, des couleurs échantillonnées uniformement le long de chaque axe de l'espace sont perçues comme évoluant à pas constant.

Notions de base en géométrie différentielle

À l'attention des lecteurs moins familiers de la géométrie différentielle, l'annexe B présente brièvement et le plus pédagogiquement possible l'ensemble des notions nécessaires à l'appréhension des comportements surfaciques communément recherchés par la plupart des techniques précédentes pour localiser les courbes de leurs rendus au trait de géométries. L'ensemble de cette brève annexe est donc principalement consacrée au concept de courbure et de ses dérivées, véritablement ici au cœur de toutes les tractations. Introduite dans le cas de courbes 2d, la courbure normale en un point de la surface pour une direction donnée est ensuite naturellement étendue aux cas de surfaces plongées dans un espace 3d. La caractérisation locale du comportement de la surface autour d'un point par l'analyse de l'opérateur de forme est assurée par l'extraction des courbures principales, ainsi que par celle de leurs directions. Enfin, la structure d'ordre 3 correspondant aux dérivées directionnelles de la courbure normale assure la connaissance de l'évolution locale de celle-ci. Finalement, la transposition des formules mathématiques théoriques au cas pratique de leur évaluation à la surface des maillages 3d est également abordée, ainsi que les principales approches pour y parvenir : l'échantillonnage de courbes irradiant d'un point de la surface, l'approximation paramétrique de patchs de la surface, ou la moyenne de tenseurs de courbures autour des faces et sommets du maillage.

Bibliography

[ACOL00] Marc Alexa, Daniel Cohen-Or, and David Levin. As-rigid-as-possible shape interpolation. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 157–164, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 75

[AM99] Marc Alexa and Wolfgang Müller. The morphing space. In *Proceedings of WSCG 99, Plzen*, pages 329–336, 1999. 75

[App67] Arthur Appel. The notion of quantitative invisibility and the machine rendering of solids. In *Proceedings of the 1967 22nd national conference*, pages 387–393, New York, NY, USA, 1967. ACM. 149, 156

[Ash01] Michael Ashikhmin. Synthesizing natural textures. In *I3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 217–226, New York, NY, USA, 2001. ACM. 32, 33, 36, 40, 41, 59

[AT05] Gady Agam and Xiaojing Tang. A sampling framework for accurate curvature estimation in discrete surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 11(5):573–583, 2005. 234

[AWI+09] Zainab AlMeraj, Brian Wyvill, Tobias Isenberg, Amy A. Gooch, and Richard Guy. Automatically mimicking unique hand-drawn pencil lines. *Computers & Graphics*, In Press, Corrected Proof:–, 2009. 74, 75

[BA83] Peter J. Burt and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31,4:532–540, 1983. 109

[BA88] James R. Bergen and Edward H. Adelson. Early vision and texture perception. *Nature*, 333:363–364, 1988. 23

[BA06] William Baxter and Ken-ichi Anjyo. Latent doodle space. *Computer Graphics Forum (Proc. Eurographics)*, 25(3):477–485, 2006. 75, 76

[BAK97] A. G. Belyaev, E. V. Anoshkina, and T. L. Kunii. Ridges, ravines, and singularities. *Topological Modeling for Visualization*, 18:375–383, 1997. 142

[BB03] R. Bala and K. M. Braun. Color-to-grayscale conversion to maintain discriminability. In R. Eschbach and G. G. Marcu, editors, *Color Imaging IX: Processing, Hardcopy, and Applications. Edited by Eschbach, Reiner; Marcu, Gabriel G. Proceedings of the SPIE, Volume 5293, pp. 196-202 (2003).*, volume 5293 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 196–202, December 2003. 81, 93, 245

[BBA08] William Baxter, Pascal Barla, and Ken-ichi Anjyo. Rigid shape interpolation using normal equations. In *NPAR '08: Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, pages 59–64. ACM Press, June 2008. 75

[BBA09a] William Baxter, Pascal Barla, and Ken Anjyo. Compatible embedding for 2D shape animation. *IEEE Transactions Visualization and Computer Graphics*, 15:867–879, 2009. 75, 76

[BBA09b] William Baxter, Pascal Barla, and Ken Anjyo. N-way morphing for 2D animation. *Computer Animation and Virtual Worlds (proc. CASA 2009)*, 20:79–87, 2009. 76

[BBMT06] Pascal Barla, Simon Breslav, Lee Markosian, and Joëlle Thollot. Interactive hatching and stippling by example. Technical report, INRIA, 2006. 60, 72

[BBT+06] Pascal Barla, Simon Breslav, Joëlle Thollot, François Sillion, and Lee Markosian. Stroke pattern analysis and synthesis. In *Computer Graphics Forum (Proc. of Eurographics 2006)*, volume 25, pages 663–671, 2006. 45, 51, 52, 56, 59, 60, 69, 71, 72, 244

[BCD08] J.-M. Billiot, J.-F. Coeurjolly, and R. Drouilhet. Maximum pseudolikelihood estimator for exponential family models of marked gibbs point processes. *Electronic Journal of Statistics*, 2:234–264, 2008. 66

[BCGF10] Pierre Bénard, Forrester Cole, Aleksey Golovinskiy, and Adam Finkelstein. Self-Similar Texture for Coherent Line Stylization. In *NPAR 2010: 8th International Symposium on Non-photorealistic Animation and Rendering*, Annecy France, 06 2010. 156

[BD02] Stephen Brooks and Neil Dodgson. Self-similarity based texture editing. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 653–656, New York, NY, USA, 2002. ACM. 43

[BE04] Raja Bala and Reiner Eschbach. Spatial color-to-grayscale transform preserving chrominance edge information. In *Color Imaging Conference*, pages 82–86. IS&T - The Society for Imaging Science and Technology, 2004. 83, 121, 245

[Bes74] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36:192–236, 1974. 20

[Bes77] J. Besag. Some methods of statistical analysis for spatial data. *Bulletin of the International Statistical Institute*, 47(2):77–92, 1977. 24, 66

[BGG+06] James Bigler, James Guilkey, Christiaan P. Gribble, Charles D. Hansen, and Steven G. Parker. A case study: Visualizing material point method data. In *EuroVis*, pages 299–306, 2006. 136

[BHT00] Yingjun Bai, Steven J. Harrington, and Jean Taber. Improved algorithmic mapping of color to texture. In *Proceedings of SPIE*, volume 4300, pages 444–451, 2000. 86

[BK68] R. M. Boynton and P. K. Kaiser. Vision: the additivity law made to work for heterochromatic photometry with bipartite fields. *Science*, 161:366, 1968. 90

[BKR+05] Michael Burns, Janek Klawe, Szymon Rusinkiewicz, Adam Finkelstein, and Doug DeCarlo. Line drawings from volume data. *ACM Trans. Graph.*, 24(3):512–518, 2005. 152

[BKY97] Peter N. Belhumeur, David J. Kriegman, and Alan L. Yuille. The bas-relief ambiguity. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 1060, Washington, DC, USA, 1997. IEEE Computer Society. 128

[Bli78a] James F. Blinn. Simulation of wrinkled surfaces. *SIGGRAPH Comput. Graph.*, 12(3):286–292, 1978. 10

[Bli78b] James Frederick Blinn. *Computer display of curved surfaces*. PhD thesis, The University of Utah, 1978. 10

[BSS07] Meru Brunn, Mario Costa Sousa, and Faramarz F. Samavati. Capturing and re-using artistic styles with reverse subdivision-based multiresolution methods. *Int. J. Image Graphics*, 7(4):593–615, 2007. 61

[BSW08] Mikhail Belkin, Jian Sun, and Yusu Wang. Discrete laplace operator on meshed surfaces. In *SCG '08: Proceedings of the twenty-fourth annual symposium on Computational geometry*, pages 278–287, New York, NY, USA, 2008. ACM. 159, 168

[BVZ01] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001. 37, 56

[Čad08a] Martin Čadík. Perceptual evaluation of color-to-grayscale image conversions. *Comput. Graph. Forum*, 27(7):–, 2008. 78, 97, 117, 120, 121, 198, 247

[Čad08b] Martin Čadík. *Perceptually Based Image Quality Assessment and Image Transformations*. Ph.d. thesis, Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, January 2008. 97

[Cat74] Edwin E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. PhD thesis, Dept. of CS, U. of Utah, December 1974. 10, 133

[Cat75] Edwin E. Catmull. Computer display of curved surfaces. In *Proceedings of the IEEE Conference on Computer Graphics, Pattern Recognition, and Data Structure*, pages 11—17, May 1975. 9

[Cav05] Patrick Cavanagh. The artist as neuroscientist. *Nature*, 434:301–307, 2005. 99, 129, 148

[CB78a] T. Caelli and Julesz B. On perceptual analyzers underlying visual texture discrimination: part i. *Biol. Cybernetics*, 28:167–175, 1978. 18, 23

[CB78b] T. Caelli and Julesz B. On perceptual analyzers underlying visual texture discrimination: part ii. *Biol. Cybernetics*, 29:201–214, 1978. 18

[CF08] Forrester Cole and Adam Finkelstein. Partial visibility for stylized lines. In *NPAR 2008*, June 2008. 156, 162

[CGL+08] Forrester Cole, Aleksey Golovinskiy, Alex Limpaecher, Heather Stoddart Barros, Adam Finkelstein, Thomas Funkhouser, and Szymon Rusinkiewicz. Where do people draw lines? In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–11, New York, NY, USA, 2008. ACM. 131, 157, 158, 163, 165, 190, 199, 248, 249

[CHRW09] Ming Cui, Jiuxiang Hu, Anshuman Razdan, and Peter Wonka. Color-to-gray conversion using isomap. *The Visual Computer*, 2009. 83, 95

[CJTF98] Wagner Toledo Corrêa, Robert J. Jensen, Craig E. Thayer, and Adam Finkelstein. Texture mapping for cel animation. In *Proceedings of SIGGRAPH 98*, pages 435–446, July 1998. 139

[CM02] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, May 2002. 48

[Coo84] Robert L. Cook. Shade trees. *SIGGRAPH Comput. Graph.*, 18(3):223–231, 1984. 10

[Cor70] Tom N. Cornsweet. *Visual perception / Tom N. Cornsweet*. Academic Press, New York :, 1970. 103, 105, 246

[CP09] A.N.M. Imroz Choudhury and Steven G. Parker. Ray tracing npr-style feature lines. In *NPAR '09: Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering*, pages 5–14, New York, NY, USA, 2009. ACM. 136, 138

[Cra66] Kenneth James Williams. Craik. *The nature of psychology : a selection of papers, essays / and other writings. Ed. by S.L.Sherwood.* E.U.P., Camb. :, 1966. 103, 105, 246

[CSD+09] Forrester Cole, Kevin Sanik, Doug DeCarlo, Adam Finkelstein, Thomas Funkhouser, Szymon Rusinkiewicz, and Manish Singh. How well do line drawings depict shape? In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, pages 1–9, New York, NY, USA, 2009. ACM. 131, 157, 158, 164, 199, 248

[CST05] Paolo Cignoni, Roberto Scopigno, and Marco Tarini. A simple normal enhancement technique for interactive non-photorealistic renderings. *Computers & Graphics*, 29(1):125–133, 2005. 101, 102, 107

[Cur98] Cassidy J. Curtis. Loose and sketchy animation. In *SIGGRAPH '98: ACM SIGGRAPH 98 Electronic art and animation catalog*, page 145, New York, NY, USA, 1998. ACM. 138, 139

[CZM+10] Ming-Ming Cheng, Fang-Lue Zhang, Niloy J. Mitra, Xiaolei Huang, and Shi-Min Hu. Repfinder: Finding approximately repeated scene elements for image editing. *ACM Transactions on Graphics*, 29(3):to appear, 2010. 196, 197

[Dau85] J.G. Daugman. Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am.*, 2(7):1160–1169, 1985. 23

[DB66] J. E. Dowling and B. B. Boycott. Organization of the primate retina: Electron microscopy. In *Proceedings of the Royal Society B: Biological Sciences*, volume 166, pages 80–111, 1966. 208

[dB97] Jeremy S. de Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 361–368, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. 25, 26, 27

[dBV97] J. S. de Bonet and P. A. Viola. Rosetta: An image database retrieval system. In *Proceedings of the Image Understanding Workshop*, 1997. 25

[dBV98] J. S. de Bonet and P. Viola. Texture recognition using a non-parametric multi-scale statistical model. In *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 641, Washington, DC, USA, 1998. IEEE Computer Society. 25

[DCFB09] Mark S. Drew, David Connah, Graham D. Finlayson, and Marina Bloj. Improved colour to greyscale via integrability correction. *Human Vision and Electronic Imaging XIV*, 7240(1):72401B, 2009. 84

[Dec96] Philippe Decaudin. Cartoon looking rendering of 3D scenes. Research Report 2919, INRIA, June 1996. 135, 137, 138, 139

[DFR04] Doug DeCarlo, Adam Finkelstein, and Szymon Rusinkiewicz. Interactive rendering of suggestive contours with temporal coherence. In *Third International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 15–24, June 2004. 151, 168

[DFRS03] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. Suggestive contours for conveying shape. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 848–855, New York, NY, USA, 2003. ACM. 137, 139, 145, 146, 147, 151, 170, 177, 184, 248

[DJ68] E.M. Darling and R.D. Joseph. Pattern recognition from satellite altitudes. In *IEEE Transactions on Systems Science and Cybernetics*, 1968. 18

[DJP+92] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiway cuts (extended abstract). In *STOC '92: Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 241–251, New York, NY, USA, 1992. ACM. 49

[DMLG02] J.-M. Dischler, K. Maritaud, B. Lévy, and D. Ghazanfarpour. Texture particles. *Computer Graphics Forum*, 21:401–410, 2002. 34, 35, 60, 73

[DMM00] A. Desolneux, L. Moisan, and J-M Morel. Meaningful alignments. *International Journal of Computer Vision*, 40:7–23, 2000. 64

[DMM03] A. Desolneux, L. Moisan, and J.M. More. A grouping principle and four applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):508–513, 2003. 63

[DNGK97] Kristin J. Dana, Shree K. Nayar, Bram Van Ginneken, and Jan J. Koenderink. Reflectance and texture of real-world surfaces authors. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 151, Washington, DC, USA, 1997. IEEE Computer Society. 10

[dQB06] R.L. de Queiroz and K.M. Braun. Color to gray and back: color embedding into textured gray images. *Image Processing, IEEE Transactions on*, 15(6):1464 –1470, june 2006. 85

[DR07] Doug DeCarlo and Szymon Rusinkiewicz. Highlight lines for conveying shape. In *NPAR '07: Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, pages 63–70, New York, NY, USA, 2007. ACM. 137, 139, 168, 177, 248

[DZP08] Weiming Dong, Ning Zhou, and Jean-Claude Paul. Perspective-aware texture analysis and synthesis. *Vis. Comput.*, 24(7):515–523, 2008. 41

[EF01] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, New York, NY, USA, 2001. ACM. 34, 36, 37, 38, 41, 241

[EL99] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *ICCV (2)*, pages 1033–1038, 1999. 28, 29, 30, 33, 40, 45, 59, 241

[EMP+02] David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing and Modeling: A Procedural Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002. 16, 241

[Fer03] James A. Ferwerda. Three varieties of realism in computer graphics. In *Proceedings SPIE Human Vision and Electronic Imaging '03*, pages 290–297, 2003. 129, 130, 199, 248

[FFC82] Alain Fournier, Don Fussell, and Loren Carpenter. Computer rendering of stochastic models. *Commun. ACM*, 25(6):371–384, 1982. 16

[FP80] Olivier D. Faugeras and William K. Pratt. Decorrelation methods of texture feature extraction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-2(4):323–332, July 1980. 23

[FP91] Mark D. Fairchild and Elizabeth Pirrotta. Predicting the lightness of chromatic object colors using cielab. *Color Research & Application*, 16(6):385–393, 1991. 91, 95, 96, 97, 246

[FS95] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT '95: Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, London, UK, 1995. Springer-Verlag. 174, 175, 249

[FTA04] Roland W. Fleming, Antonio Torralba, and Edward H. Adelson. Specular reflections and the perception of shape. *J. Vis.*, 4(9):798–820, 9 2004. 129, 137, 154

[FTP03] William T. Freeman, Joshua B. Tenenbaum, and Egon C. Pasztor. Learning style translation for the lines of a drawing. *ACM Trans. Graph.*, 22(1):33–46, 2003. 61, 74

[Gar85] Geoffrey Y. Gardner. Visual simulation of clouds. *SIGGRAPH Comput. Graph.*, 19(3):297–304, 1985. 10, 15

[GBN96] S. Guillon, P. Baylou, and M. Najim. Robust nonlinear contrast enhancement filters. In *Image Processing, 1996. Proceedings., International Conference on*, volume 1, pages 757 –760 vol.1, sep 1996. 118

[GBNK98] Sébastien Guillon, Pierre Baylou, Mohamed Najim, and Naamen Keskes. Adaptive nonlinear filters for 2d and 3d image enhancement. *Signal Process.*, 67(3):237–254, 1998. 118

[GD95] D. Ghazanfarpour and J. M. Dischler. Spectral analysis for automatic 3-d texture generation. *Computers & Graphics*, 19(3):413 – 422, 1995. 26, 27

[GD07] Mark Grundland and Neil A. Dodgson. Decolorize: Fast, contrast enhancing, color to grayscale conversion. *Pattern Recognition*, 40(11):2891–2896, 2007. 78, 85, 114, 116, 120, 121, 245

[GDM69] S. L. Guth, N. V. Donley, and R. T. Marrocco. On luminance additivity and related topics. *Vision Res.*, 9:537–575, 1969. 90

[GI04] Jack Goldfeather and Victoria Interrante. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. Graph.*, 23(1):45–63, 2004. 235

[GIHL00] Ahna Girshick, Victoria Interrante, Steven Haker, and Todd Lemoine. Line direction matters: an argument for the use of principal directions in 3d line drawings. In *NPAR '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 43–52, New York, NY, USA, 2000. ACM. 142, 145, 168

[Gim96a] G. L. Gimel'farb. Texture modeling by multiple pairwise pixel interactions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(11):1110–1114, 1996. 22, 23

[Gim96b] G.L. Gimel'farb. Non-markov gibbs texture model with multiple pairwise pixel interactions. *Pattern Recognition, International Conference on*, 2:591, 1996. 23

[GIS01] Gabriele Gorla, Victoria Interrante, and Guillermo Sapiro. Growing fitted textures. SIGGRAPH 2001 Sketches and Applications, 2001. 30

[Glu05] J. Gluckman. Visually distinct patterns with matching subband statistics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(2):252–264, Feb. 2005. 23

[GM94] C.J. Geyer and J. Møller. Simulation procedures and likelihood inference for spatial point processes. *Scandinavian Journal of Statistics*, pages 359–373, 1994. 67

[GOTG05] Amy A. Gooch, Sven C. Olsen, Jack Tumblin, and Bruce Gooch. Color2gray: salience-preserving color removal. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 634–639, New York, NY, USA, 2005. ACM Press. 83, 97, 114, 116, 117, 121, 245

[Gre95] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995. 68

[GS81] D.D. Garber and A.A. Sawchuk. Texture simulation using a best-fit model. In *PRIP81*, pages 603–608, 1981. 28

[GSDM86] A. Gagalowicz and Song-Di-Ma. Model driven synthesis of natural textures for 3-d scenes. *Computers & Graphics*, 10(2):161–170, 1986. 13

[GSG+99] Bruce Gooch, Peter-Pike J. Sloan, Amy Gooch, Peter Shirley, and Richard Riesenfeld. Interactive technical illustration. In *I3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 31–38, New York, NY, USA, 1999. ACM. 149

[GU02] Jens Gravesen and Michael Ungstrup. Constructing invariant fairness measures for surfaces. *Advances in Computational Mathematics*, 17:67–88, 2002. 233

[GVH07] Todd Goodwin, Ian Vollick, and Aaron Hertzmann. Isophote distance: a shading approach to artistic stroke thickness. In *NPAR '07: Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, pages 53–62, New York, NY, USA, 2007. ACM. 192

[GZW01] G.-E. Guo, S.-C. Zhu, and Y. Wu. Visual learning by integrating descriptive and generative methods. *ICCV '01*, 1:370–377, 2001. 60, 73

[Har79] R.M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, May 1979. 19

[HB95] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238, New York, NY, USA, 1995. ACM. 25, 26, 27, 59

[HCV52] David Hilbert and S. Cohn-Vossen. *Geometry and the Imagination*. Chelsea Pub. Co., 1952. 141

[Her99] Aaron Hertzmann. Introduction to 3D Non-Photorealistic Rendering: Silhouettes and Outlines. In Stuart Green, editor, *ACM SIGGRAPH 99 Course Notes. Course on Non-Photorelistic Rendering*, chapter 7. ACM Press/ACM SIGGRAPH, New York, 1999. 135, 137

[HJO+01] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In Eugene Fiume, editor, *SIGGRAPH 2001, Computer Graphics Proceedings*, pages 327–340. ACM Press / ACM SIGGRAPH, 2001. 32, 36, 41, 42

[HLEL06] James H. Hays, Marius Leordeanu, Alexei A. Efros, and Yanxi Liu. Discovering texture regularity as a higher-order correspondence problem. In *9th European Conference on Computer Vision*, May 2006. 13, 14, 39

[HOCS02] Aaron Hertzmann, Nuria Oliver, Brian Curless, and Steven M. Seitz. Curve analogies. In *EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering*, pages 233–246, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association. 41, 61, 74

[HRP+84] D.D. Hoffman, Whitman Richards, Alex Pentl, John Rubin, and Joseph Scheuhammer. Parts of recognition. *Cognition*, 18:65–96, 1984. 142, 248

[HRV02] M. Hubert, P.J. Rousseeuw, and S. Verboven. A fast method for robust principal components with applications to chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 60(1-2):101–111, 2002. 64

[HS03] Aaron Hertzmann and Steven M. Seitz. Shape and materials by example: A photometric stereo approach. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:533, 2003. 12, 13

[HSD73] Robert M. Haralick, K. Shanmugam, and Its'hak Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, 3(6):610–621, 1973. 18, 20

[HZ00] Aaron Hertzmann and Denis Zorin. Illustrating smooth surfaces. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 517–526, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 130, 142, 150, 168, 177

[HZW+06] Jianwei Han, Kun Zhou, Li-Yi Wei, Minmin Gong, Hujun Bao, Xinming Zhang, and Baining Guo. Fast example-based surface texture synthesis via discrete optimization. *Vis. Comput.*, 22(9):918–925, 2006. 30, 40

[IFP95] Victoria Interrante, Henry Fuchs, and Stephen Pizer. Enhancing transparent skin surfaces with ridge and valley lines. In *VIS '95: Proceedings of the 6th conference on Visualization '95*, page 52, Washington, DC, USA, 1995. IEEE Computer Society. 142

[IMIM08] Takashi Ijiri, Radomír Mech, Takeo Igarashi, and Gavin Miller. An example-based procedural system for element arrangement. *Comput. Graph. Forum*, 27(2):429–436, 2008. 45, 51, 52, 56, 60, 61, 69, 70, 71, 72, 73, 244

[JDA07] Tilke Judd, Frédo Durand, and Edward Adelson. Apparent ridges for line drawing. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 19, New York, NY, USA, 2007. ACM. 153, 154, 168, 170, 177, 248

[JEGPO02] P.-M. Jodoin, E. Epstein, M. Granger-Piché, and V. Ostromoukhov. Hatching by example: a statistical approach. *NPAR '02*, pages 29–36, 2002. 60, 74

[JFK03] N. Jojic, B.J. Frey, and A. Kannan. Epitomic analysis of appearance and shape. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 34–41 vol.1, Oct. 2003. 196

[JGSF73] B. Julesz, E.N. Gilbert, L.A. Shepp, and H.L. Frisch. Inability of humans to discriminate between visual textures that agree in second-order statistics – revisited. *Perception*, 2:391–405, 1973. 18

[JGV78] B. Julesz, E. N. Gilbert, and J. D. Victor. Visual discrimination of textures with identical third-order statistics. *Biological Cybernetics*, 31:137–140, 1978. 18, 21

[JM91] J.L. Jensen and J. Møller. Pseudolikelihood for exponential family models of spatial point processes. *Annals of Applied Probability*, 1:445–461, 1991. 66

[JNLM05] Kyuman Jeong, Alex Ni, Seungyong Lee, and Lee Markosian. Detail control in line drawings of 3d meshes. In *Proceedings of Pacific Graphics 2005*, 2005. 152

[Jud51] D. B. Judd. Report of u.s. secretariat committee on colorimetry and artificial daylight. In *CIE Proceedings*, 1951. 90

[Jul62] B. Julesz. Visual pattern discrimination. *Information Theory, IRE Transactions on*, 8(2):84–92, February 1962. 17, 18, 20

[Jul75] B. Julesz. Experiments in the visual perception of texture. *Sci Am.*, 232:34–43, 1975. 18, 244

[Jul86] B. Julesz. Texton gradients: The texton theory revisited. *Biological Cybernetics*, 54(4):245–251, 1986. 59, 62

[KDCL96] Jan J. Koenderink, Andrea J. van Doorn, Chris Christou, and Joseph S. Lappin. Shape constancy in pictorial relief. In *ECCV '96: Proceedings of the International Workshop on Object Representation in Computer Vision II*, pages 151–164, London, UK, 1996. Springer-Verlag. 127, 128, 137, 158, 168

[KDMF03] Robert D. Kalnins, Philip L. Davidson, Lee Markosian, and Adam Finkelstein. Coherent stylized silhouettes. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 22(3):856–861, July 2003. 156

[KEBK05] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, volume 24, pages 795–802, New York, NY, USA, 2005. ACM. 30, 31, 40, 241

[KFCO+07] Johannes Kopf, Chi-Wing Fu, Daniel Cohen-Or, Oliver Deussen, Dani Lischinski, and Tien-Tsin Wong. Solid texture synthesis from 2d exemplars. *ACM Trans. Graph.*, 26(3):2, 2007. 30, 40, 41

[KJDL09] Yongjin Kim, Cheolhun Jang, Julien Demouth, and Seungyong Lee. Robust color-to-gray via nonlinear global mapping. *ACM Trans. Graph.*, 28(5):1–4, 2009. 86, 95, 117, 118

[KKLE06] Stephan Kopf, Thomas King, Fleming Lampi, and Wolfgang Effelsberg. Video color adaptation for mobile devices. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 963–964, New York, NY, USA, 2006. ACM. 86

[KMI+09] SungYe Kim, Ross Maciejewski, Tobias Isenberg, William M. Andrews, Wei Chen, Mario Costa Sousa, and David S. Ebert. Stippling by example. *In Proceedings of the 7th international symposium on Non-photorealistic animation and rendering (NPAR)*, pages to–appear, 2009. 200

[KMM+02] Robert D. Kalnins, Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Joseph C. Lee, Philip L. Davidson, Matthew Webb, John F. Hughes, and Adam Finkelstein. Wysiwyg npr: Drawing strokes directly on 3d models. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 21(3):755–762, July 2002. 61, 74, 164, 165

[KNS+09] Evangelos Kalogerakis, Derek Nowrouzezahrai, Patricio Simari, James McCrae, Aaron Hertzmann, and Karan Singh. Data-driven curvature for real-time line drawing of dynamic scenes. *ACM Transactions on Graphics*, 29(5), 2009. 156

[Koe84] J.J. Koenderink. What does the occluding contour tell us about solid shape? *Perception*, 13:321–330, 1984. 148, 184, 248

[Koe90] J. J. Koenderink. *Solid Shape*. MIT Press, Cambridge, MA, 1990. 141, 153, 168

[KOF08] Giovane R. Kuhn, Manuel M. Oliveira, and Leandro A. F. Fernandes. An improved contrast enhancing approach for color-to-grayscale mappings. *Vis. Comput.*, 24(7):505–514, 2008. 84

[Koh35] V. A. Kohlrausch. Zur photometrie farbiger lichter. *Das Licht*, 5:259–275, 1935. 88

[KS00] Craig S. Kaplan and David H. Salesin. Escherization. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 499–510, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 37, 200

[KSE+03] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: image and video synthesis using graph cuts. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, volume 22, pages 277–286, New York, NY, USA, 2003. ACM. 34, 36, 37, 38, 53, 241

[KST08] Michael Kolomenkin, Ilan Shimshoni, and Ayellet Tal. Demarcating curves for shape illustration. In *SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers*, pages 1–9, New York, NY, USA, 2008. ACM. 143, 145, 158, 168, 170, 177, 184, 248

[KST09] M. Kolomenkin, I. Shimshoni, and A. Tal. On edge detection on surfaces. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:2767–2774, 2009. 144, 145, 248

[KvDK92] JJ Koenderink, AJ van Doorn, and AM Kappers. Surface perception in pictures. *Perception & psychophysics*, 52:487–496, 1992. 127, 248

[KWR06] Anitha Kannan, John M. Winn, and Carsten Rother. Clustering appearance and shape by learning jigsaws. In *NIPS*, pages 657–664, 2006. 196

[LCD06] Thomas Luft, Carsten Colditz, and Oliver Deussen. Image enhancement by unsharp masking the depth buffer. *ACM Transactions on Graphics*, 25(3):1206–1213, jul 2006. 101, 102, 107

[LCT04] Yanxi Liu, Robert T. Collins, and Yanghai Tsin. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:354–371, 2004. 38, 39, 43, 54, 60, 196, 241

[LD05] Ares Lagae and Philip Dutré. A procedural object distribution function. *ACM Transactions on Graphics*, 24(4):1442–1461, October 2005. 16, 17

[LGK06] W. S. Lin, Y. L. Gai, and A. A. Kassim. Perceptual impact of edge sharpness in images. *IEE proceedings. Vision, image and signal processing*, 153:215–223, 2006. 99, 246

[LH05] Sylvain Lefebvre and Hugues Hoppe. Parallel controllable texture synthesis. *ACM Trans. Graph.*, 24(3):777–786, 2005. 31, 32, 42, 43

[LH06] Sylvain Lefebvre and Hugues Hoppe. Appearance-space texture synthesis. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, volume 25, pages 541–548, New York, NY, USA, 2006. ACM Press. 31, 32, 43, 59

[Lin98] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30:77–116, 1998. 47

[LKL06] Hyunjun Lee, Sungtae Kwon, and Seungyong Lee. Real-time pencil rendering. In *NPAR '06: Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pages 37–45, New York, NY, USA, 2006. ACM. 138, 139

[LL06] Wen-Chieh Lin and Yanxi Liu. Tracking dynamic near-regular textures under occlusion and rapid movements. In *9th European Conference on Computer Vision*, May 2006. 39

[LLDD09] Ares Lagae, Sylvain Lefebvre, George Drettakis, and Philip Dutré. Procedural noise using sparse gabor convolution. *ACM Transactions on Graphics (SIGGRAPH Conference Proceedings)*, 28(3), August 2009. 16

[LLH04] Yanxi Liu, Wen-Chieh Lin, and James Hays. Near-regular texture analysis and manipulation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, volume 23, pages 368–376, New York, NY, USA, 2004. ACM. 38, 43, 52, 60, 196

[LLX+01] Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum. Real-time texture synthesis by patch-based sampling. *ACM Trans. Graph.*, 20(3):127–150, 2001. 36

[LM05] Erik Lum and Kwan-Liu Ma. Expressive line selection by example. In *13th Pacific Conference on Computer Graphics and Applications (PG'05)*, 2005. 164, 165, 166, 190, 249

[LMLH07] Yunjin Lee, Lee Markosian, Seungyong Lee, and John F. Hughes. Line drawings via abstracted shading. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 18, New York, NY, USA, 2007. ACM. 137, 139

[LOG06] Sangwon Lee, Sven C. Olsen, and Bruce Gooch. Interactive 3d fluid jet painting. In *NPAR '06: Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pages 97–104, New York, NY, USA, 2006. ACM. 200

[Low99] D.G. Lowe. Object recognition from local scale-invariant features. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 2:1150–1157 vol.2, 1999. 46

[LP90] Y. H. Lee and S. Y. Park. A study of convex/concave edges and edge-enhancing operators based on the laplacian. In *IEEE Transactions on Circuits and Systems*, volume 37, pages 940 – 946, 1990. 118

[LP99] R. B. Lotto and D. Purves. The effects of color on brightness. *Nature Neuroscience*, pages 1010–1014, 1999. 105, 106

[LP00] R. Beau Lotto and Dale Purves. An empirical explanation of color contrast. *Proceedings of the National Academy of Sciences of the United States of America*, 97(23):12834–12839, 2000. 105

[LPRM02] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.*, 21(3):362–371, 2002. 12

[LTL05] Yanxi Liu, Yanghai Tsin, and Wen-Chieh Lin. The promise and perils of near-regular texture. *Int. J. Comput. Vision*, 62(1-2):145–159, 2005. 39

[LVLD09] Ares Lagae, Peter Vangorp, Toon Lenaerts, and Philip Dutré. Isotropic stochastic procedural textures by example. Report CW 546, Department of Computer Science, K.U.Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium, May 2009. 26, 27

[Max99] Nelson Max. Weights for computing vertex normals from facet normals. *J. Graph. Tools*, 4(2):1–6, 1999. 143, 235

[McC93] Scott McCloud. *Understanding Comics: The Invisible Art*. Dc Comics, 1993. 200

[MDSB03] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In Hans-Christian Hege and Konrad Polthier, editors, *Visualization and Mathematics III*, pages 35–57. Springer-Verlag, Heidelberg, 2003. 234

[MKG+97] Lee Markosian, Michael A. Kowalski, Daniel Goldstein, Samuel J. Trychin, John F. Hughes, and Lubomir D. Bourdev. Real-time nonphotorealistic rendering. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 415–420, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. 149, 152, 155, 156

[MLLY91] S.K. Mitra, H. Li, I.-S. Lin, and T.-H. Yu. A new class of nonlinear filters for image enhancement. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2525 – 2528, 1991. 118

[MMS06] Rafal Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. A perceptual framework for contrast processing of high dynamic range images. *ACM Trans. Appl. Percept.*, 3(3):286–308, 2006. 84, 245

[MS05] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005. 46

[MWGZ09] Chongyang Ma, Li-Yi Wei, Baining Guo, and Kun Zhou. Motion field texture synthesis. In *SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers*, pages 1–8, New York, NY, USA, 2009. ACM. 43

[NA03] Andrew Nealen and Marc Alexa. Hybrid texture synthesis. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering*, pages 97–105, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 36

[Nay97] Yoshinobu Nayatani. Simple estimation methods for the helmholtz-kohlrausch effect. *Color Research & Application*, 22(6):385–401, 1997. 92, 95, 96, 97, 246

[NCN07] Laszlo Neumann, Martin Cadik, and Antal Nemcsics. An efficient perception-based adaptive color to gray transformation. In *Proceedings of Computational Aesthetics 2007*, pages 73– 80, Banff, Canada, 2007. Eurographics Association. 84, 98, 109, 117, 121, 245

[NJLM06] Alex Ni, Kyuman Jeong, Seungyong Lee, and Lee Markosian. Multi-scale line drawings from 3d meshes. In *I3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 133–137, New York, NY, USA, 2006. ACM. 152

[NJLS05] Kyung Na, Moon Jung, Jongwan Lee, and Chang Geun Song. Redeeming valleys and ridges for line-drawing. In *PCM (1)*, pages 327–338, 2005. 145, 154

[NM00] J. D. Northrup and Lee Markosian. Artistic silhouettes: a hybrid approach. In *NPAR '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering*, pages 31–37, New York, NY, USA, 2000. ACM. 156

[NNN05] Laszlo Neumann, Antal Nemcsics, and Attila Neumann. Computational color harmony based on coloroid system. In L. Neumann, M. Sbert, B. Gooch, and W. Purgathofer, editors, *Computational Aesthetics in Graphics, Visualization and Imaging 2005*, pages 231–240, May 2005. 226

[NRH+92] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. Geometrical considerations and nomenclature for reflectance. In *Radiometry*, pages 94–145, , USA, 1992. Jones and Bartlett Publishers, Inc. 10

[NZvG03] A. Neubeck, A. Zalesny, and L. van Gool. Cut-primed smart copying. In *In Proc. 3rd Int. Workshop "Texture 2003", Nice, France, October 17,2003. Heriot-Watt Univ., pp. 119-124, 2003.*, 2003. 38, 241

[O'B59] Vivien O'Brien. Contrast by contour-enhancement. *American Journal of Psychology*, 72:299–300, 1959. 103, 105, 246

[OBM00] Manuel M. Oliveira, Gary Bishop, and David McAllister. Relief texture mapping. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 359–368, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 10

[OBS04] Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. Ridge-valley lines on meshes via implicit surface fitting. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 609–612, New York, NY, USA, 2004. ACM. 143

[PBCL09] Minwoo Park, Kyle Brocklehurst, Robert T. Collins, and Yanxi Liu. Deformed lattice detection in real-world images using mean-shift belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(10):1804–1816, 2009. 39

[Per85] Ken Perlin. An image synthesizer. *SIGGRAPH Comput. Graph.*, 19(3):287–296, 1985. 15, 17

[PFH00] Emil Praun, Adam Finkelstein, and Hugues Hoppe. Lapped textures. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 465–470, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 34, 35

[PL98] R. Paget and I.D. Longstaff. Texture synthesis via a noncausal nonparametric multiscale markov random field. *Image Processing, IEEE Transactions on*, 7(6):925–931, Jun 1998. 31

[PLC08] Minwoo Park, Yanxi Liu, and Robert Collins. Efficient mean shift belief propagation for vision tracking. In *Proceedings of CVPR 2008*, June 2008. 39

[PP93] Kris Popat and Rosalind W. Picard. Novel cluster-based probability model for texture synthesis, classification, and compression. In *In Visual Communications and Image Processing*, pages 756–768, 1993. 29

[PRM97] A. Polesel, G. Ramponi, and V. J. Mathews. Adaptive unsharp masking for contrast enhancement. In *in Proc. Int. Conf. Image Process*, pages 267–271, 1997. 118

[PS00] Javier Portilla and Eero P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70, 2000. 24, 25, 27, 241

[PSL99] Dale Purves, Amita Shimpi, and R. Beau Lotto. An Empirical Explanation of the Cornsweet Effect. *J. Neurosci.*, 19(19):8542–8551, 1999. 105, 118

[PWNL04] D. Purves, S. M. Williams, S. Nundy, and R. B. Lotto. Perceiving the intensity of light. *Psychological Review*, 111:142–158, 2004. 105

[Pér98] P. Pérez. Markov random fields and images. Technical Report 1196, Irisa, July 1998. 22

[Ras01] Ramesh Raskar. Hardware support for non-photorealistic rendering. In *HWWS '01: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 41–47, New York, NY, USA, 2001. ACM. 133, 134, 137, 138

[Rat85] Floyd Ratliff. The influence of contour on contrast: From cave painting to cambridge psychology. *Transactions of the American Philosophical Society*, 75:1–19, 1985. 103, 104, 105

[RBD06] Szymon Rusinkiewicz, Michael Burns, and Doug DeCarlo. Exaggerated shading for depicting shape and detail. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 25(3), July 2006. 101, 102

[RC99] Ramesh Raskar and Michael Cohen. Image precision silhouette edges. In *I3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 135–140, New York, NY, USA, 1999. ACM. 133, 137, 138

[RCcC98] Daniel L. Ruderman, Thomas W. Cronin, and Chuan chin Chiao. Statistics of cone responses to natural images: implications for visual coding. *Journal of the Optical Society of America A*, 15:2036–2045, 1998. 85

[RCWN06] Mark A. Ruzon, Scott Cohen, Gregg Wilensky, and Martin Newell. Fast and adaptive color-to-grayscale conversion. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches*, page 124, New York, NY, USA, 2006. ACM. 84

[RGW05] Karl Rasche, Robert Geist, and James Westall. Detail preserving reproduction of color images for monochromats and dichromats. *IEEE Comput. Graph. Appl.*, 25(3):22–30, 2005. 97, 114, 115, 116, 117, 121, 245

[Rip81] B. D. Ripley. *Spatial statistics*. Wiley New York, 1981. 65, 66

[RP98] Giovanni Ramponi and Andrea Polesel. A rational unsharp masking technique. *Journal of Electronic Imaging*, 7:333–338, 1998. 118

[RSI+08] Tobias Ritschel, Kaleigh Smith, Matthias Ihrke, Thorsten Grosch, Karol Myszkowski, and Hans-Peter Seidel. 3d unsharp masking for scene coherent enhancement. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–8, New York, NY, USA, 2008. ACM. 101, 102, 107, 118, 198

[RTF+04] Ramesh Raskar, Kar-Han Tan, Rogerio Feris, Jingyi Yu, and Matthew Turk. Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. *ACM Trans. Graph.*, 23(3):679–688, 2004. 134

[Rus04] Szymon Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *Symposium on 3D Data Processing, Visualization, and Transmission*, September 2004. 155, 170, 171, 235

[RvE92] J. Rossignac and M. van Emmerik. Hidden contours on a frame-buffer. In *Proceedings of the 7th Eurographics Workshop on Computer Graphics Hardware*, pages 188–204, Cambridge, UK, 1992. 133, 134, 137, 138

[SB87] Kent A. Stevens and Allen Brookes. Probing depth in monocular images. *Biological Cybernetics*, 56:355–366, 1987. 142

[SCA02] Cyril Soler, Marie-Paule Cani, and Alexis Angelidis. Hierarchical pattern mapping. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, volume 21, pages 673–680, New York, NY, USA, 2002. ACM Press. 36

[Sch90] Philip J. Schneider. An algorithm for automatically fitting digitized curves. In *Graphics gems*, pages 612–626. Academic Press Professional, Inc., San Diego, CA, USA, 1990. 169

[SF95] E.P. Simoncelli and W.T. Freeman. The steerable pyramid: a flexible architecture for multi-scale derivative computation. In *Image Processing, 1995. Proceedings., International Conference on*, volume 3, pages 444–447 vol.3, Oct 1995. 24

[SFAH92] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger. Shiftable multiscale transforms. *Information Theory, IEEE Transactions on*, 38(2):587–607, Mar 1992. 24

[SGG+00] Pedro V. Sander, Xianfeng Gu, Steven J. Gortler, Hugues Hoppe, and John Snyder. Silhouette clipping. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 327–334, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 150, 152

[SLTM08] Kaleigh Smith, Pierre-Edouard Landes, Joëlle Thollot, and Karol Myszkowski. Apparent greyscale: A simple and fast conversion to perceptually accurate images and video. *Computer Graphics Forum (Proceedings of Eurographics 2008)*, 27(2), apr 2008. url: http://www.mpi-inf.mpg.de/resources/ApparentGreyscale/. 121

[SS02] Thomas Strothotte and Stefan Schlechtweg. *Non-photorealistic computer graphics: modeling, rendering, and animation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002. 74, 167, 192

[SS08] Lindsay T. Sharpe and Andrew Stockman. *Visual Transduction and Non-Visual Light Perception*, chapter 15, pages 329–351. Humana Press, 2008. 209

[ST86] Ken Sagawa and Keishiro Takeichi. Spectral luminous efficiency functions in the mesopic range. *J. Opt. Soc. Am. A*, 3(1):71–75, 1986. 209

[ST90] Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-d shapes. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 197–206, New York, NY, USA, 1990. ACM. 134, 135, 137, 138, 168, 169

[Ste81] Kent A. Stevens. The visual interpretation of surface contours. *Artificial Intelligence*, 17:47–73, 1981. 142

[SW58] C. L. Sanders and G. Wyszecki. L/y ratios in terms of cie-chromaticity coordinates. *Journal of the Optical Society of America*, 48:389–392, 1958. 92

[SW64] C. L. Sanders and G. Wyszecki. Correlate for brightness in terms of cie color matching data. In *CIE Proceedings 15th Session*, 1964. 88, 90

[Tod85] J. T. Todd. Perception of structure from motion: is projective correspondence of moving elements a necessary condition? *Journal of experimental psychology. Human perception and performance*, 11:689–710, 1985. 148

[TOII08] Kenshi Takayama, Makoto Okabe, Takashi Ijiri, and Takeo Igarashi. Lapped solid textures: filling a model with anisotropic textures. *ACM Trans. Graph.*, 27(3):1–9, 2008. 34

[Tur52] Alan M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72, 1952. 16

[Tur91] Greg Turk. Generating textures on arbitrary surfaces using reaction-diffusion. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 289–298, New York, NY, USA, 1991. ACM. 16

[Tur01] Greg Turk. Texture synthesis on surfaces. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 347–354, New York, NY, USA, 2001. ACM. 29, 30

[TZL+02] Xin Tong, Jingdan Zhang, Ligang Liu, Xi Wang, Baining Guo, and Heung-Yeung Shum. Synthesis of bidirectional texture functions on arbitrary surfaces. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 665–672, New York, NY, USA, 2002. ACM. 30, 32, 33

[VBGS08] Romain Vergne, Pascal Barla, Xavier Granier, and Christophe Schlick. Apparent relief: a shape descriptor for stylized shading. In *NPAR '08: Proceedings of the 6th international symposium on Non-photorealistic animation and rendering*, pages 23–29. ACM, 2008. 102

[VPB+09] Romain Vergne, Romain Pacanowski, Pascal Barla, Xavier Granier, and Christophe Schlick. Light warping for enhanced surface depiction. *ACM Transaction on Graphics (Proceedings of SIGGRAPH 2009)*, 28(3), Aug 2009. 101, 102

[VPB+10] Romain Vergne, Romain Pacanowski, Pascal Barla, Xavier Granier, and Christophe Schlick. Radiance scaling for versatile surface enhancement. In *I3D '10: Proc. symposium on Interactive 3D graphics and games*. ACM, 2010. 129

[VW93] M. Visvalingam and J.D. Whyatt. Line generalisation by repeated elimination of points. *Cartographic Journal, The*, 30:46–51(6), 1993. 151

[WC83] C. Ware and W. B. Cowan. Specification of heterochromatic brightness matches - a conversion factor for calculating luminances of stimuli which are equal in brightness. In *National Research Council of Canada, Division of Physics, Pub. No. 26055*, 1983. 91

[WHCO08] Chung-Lin Wen, Chang-His Hsieh, Bing-Yu Chen, and Ming Ouhyoung. Example-based multiple local color transfer by strokes. *Computer Graphics Forum*, 27(7):1765–1772, 2008. Pacific Graphics 2008 Conference Proceedings. 197, 198

[WHZ+08] Li-Yi Wei, Jianwei Han, Kun Zhou, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Inverse texture synthesis. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–9, New York, NY, USA, 2008. ACM. 196

[WK91] Andrew Witkin and Michael Kass. Reaction-diffusion textures. *SIGGRAPH Comput. Graph.*, 25(4):299–308, 1991. 17

[WL00] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 28, 29, 33, 36, 40, 52, 53, 59, 241

[WL01] Li-Yi Wei and Marc Levoy. Texture synthesis over arbitrary manifold surfaces. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 355–360, New York, NY, USA, 2001. ACM. 29, 30, 42, 45

[WL02] Li-Yi Wei and Marc Levoy. Order-independent texture synthesis. Technical Report TR-2002-01, Computer Science Department, Stanford University, 2002. 31

[WLKT09] Li-Yi Wei, Sylvain Lefebvre, Vivek Kwatra, and Greg Turk. State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR*. Eurographics Association, 2009. 9

[WMP98] S. Mark Williams, Allison N. McCoy, and Dale Purves. An empirical explanation of brightness. *Proceedings of the National Academy of Sciences of the United States of America*, 95(22):13301–13306, 1998. 105

[WO53] Hans Wallach and D. N. O'Connell. The kinetic depth effect. *Journal of Experimental Psychology*, 45:205–217, 1953. 148

[Wor96] Steven Worley. A cellular texture basis function. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 291–294, New York, NY, USA, 1996. ACM. 17

[WS00] Günther Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae, 2nd Edition*. Wiley Series in Pure and Applied Optics, 2000. 88

[WWOH08] Huamin Wang, Yonatan Wexler, Eyal Ofek, and Hugues Hoppe. Factoring repeated content within and among images. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–10, New York, NY, USA, 2008. ACM. 196

[WY04] Qing Wu and Yizhou Yu. Feature matching and deformation for texture synthesis. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, volume 23, pages 364–367, New York, NY, USA, 2004. ACM Press. 35, 59

[Wys67] G. Wyszecki. Correlate for brightness in terms of cie chromaticity coordinates and luminous reflectance. In *Journal of the Optical Society of America*, 1967. 88

[XCOJ+09] Kai Xu, Daniel Cohne-Or, Tao Ju, Ligang Liu, Hao Zhang, Shizhe Zhou, and Yueshan Xiong. Feature-aligned shape texturing. *ACM Transactions on Graphics, (Proceedings SIGGRAPH Asia 2009)*, 28(5):1–7, 2009. 40

[XGS00] Ying-Qing Xu, Baining Guo, and Harry Shum. Chaos mosaic: Fast and memory efficient texture synthesis. Technical Report MSR-TR-2000-23, Microsoft Research, 2000. 34, 35

[XHT+07] Xuexiang Xie, Ying He, Feng Tian, Hock-Soon Seah, Xianfeng Gu, and Hong Qin. An effective illustrative visualization framework based on photic extremum lines (pels). *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1328–1335, 2007. 154, 164, 165, 168, 248

[XZWB05] Dong Xu, Hongxin Zhang, Qing Wang, and Hujun Bao. Poisson shape interpolation. In *SPM '05: Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 267–274, New York, NY, USA, 2005. ACM. 76

[YBS05] Shin Yoshizawa, Alexander Belyaev, and Hans-Peter Seidel. Fast and robust detection of crest lines on meshes. In *Proc. of ACM Symposium on Solid and Physical Modeling*, pages 227–232. Press, 2005. 143

[YHBZ01] Lexing Ying, Aaron Hertzmann, Henning Biermann, and Denis Zorin. Texture and shape synthesis on surfaces. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 301–312, London, UK, 2001. Springer-Verlag. 30

[ZBA+07] Remo Ziegler, Simon Bucheli, Lukas Ahrenberg, Marcus Magnor, and Markus Gross. A bidirectional light field - hologram transform. In Danny Cohen Or and Pavel Slavik, editors, *Computer Graphics Forum*, volume 26, pages 435–446, Prauge, Czech Republic, September 2007. Eurographics Association, Blackwell Publishing. 78

[ZDDM06] Wei Zhang, Hongli Deng, Thomas G. Dietterich, and Eric N. Mortensen. A hierarchical object recognition system based on multi-scale principal curvature regions. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 778–782, Washington, DC, USA, 2006. IEEE Computer Society. 78

[ZG03] Steve Zelinka and Michael Garland. Interactive texture synthesis on surfaces using jump maps. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering*, pages 90–96, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 30, 33, 42

[ZG04] Steve Zelinka and Michael Garland. Jump map-based interactive texture synthesis. *ACM Trans. Graph.*, 23(4):930–962, 2004. 30, 33, 42

[ZHXC09] Long Zhang, Ying He, Xuexiang Xie, and Wei Chen. Laplacian lines for real-time shape illustration. In *I3D '09: Proceedings of the 2009 symposium on Interactive 3D graphics and games*, pages 129–136, New York, NY, USA, 2009. ACM. 155, 157, 159, 168, 248

[ZWM96] Song Chun Zhu, Yingnian Wu, and David Mumford. Frame: Filters, random fields, and minimax entropy–towards a unified theory for texture modeling. In *CVPR '96: Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, page 686, Washington, DC, USA, 1996. IEEE Computer Society. 24, 27, 241

[ZWM97] Song Chun Zhu, Ying Nian Wu, and David Mumford. Minimax entropy principle and its application to texture modeling. *Neural Comput.*, 9(8):1627–1660, 1997. 24, 27

[ZZ10] Mingtian Zhao and Song-Chun Zhu. Sisley the abstract painter. In *NPAR '10: Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering*, pages 99–107, New York, NY, USA, 2010. ACM. 200

Information Extraction for Editing and By-Example Synthesis in Expressive Rendering

Abstract Processing graphical data, either for its editing or the synthesis of new content, demands a good balance between the different sources of information one may exploit. Unlike "procedural" techniques, synthesis by example stands out thanks to its extreme ease-of-use : indeed, tasks such as identification, analysis and reproduction of the distinguishing features of the user-provided examples are left to the method itself. Such approaches, along with today's intricate editing methods have greatly favored the production of compelling graphical content at a wide scale, and henceforth facilitated the adoption of computer-assisted tools by artists. But in order to meet with success, they also have to be highly controllable without being a mere extension of the artist's hand. We explore here such concerns in the context of expressive rendering and study the interactions, may they be collaborative or competitive, between the different sources of information at the core of such processes. In our opinion, there are three main sources of information: the automatic analysis of the inputs before processing; the use of prior knowledge through predetermined models; and users' explicit intervention. Through a clever combination of these sources, we propose new expressive synthesis techniques which satisfy the aforementioned usability. More than photographic realism, expressive rendering strives for the fulfillment of less easily quantifiable goals such as the intelligibility or the aesthetic value of its results. The subjectivity behind the assessment of such criteria thus forces us to attach much importance to the careful choice of the source of information to favor; the required amount of user intervention (without being detrimental to the method's theoretical value); and the possible resort to prior models (without endangering its generality). Three main synthesis instances are studied in this document: texture generation, image de-colorization, and artistic line rendering. The great disparity of inputs (raster and vector textures, complex images, 3d meshes), terms of synthesis (imitation, conversion, depiction) and objectives (preservation of a texture's visual signature, plausible restitution of chromatic contrasts, creation of drawings in accordance with users' styles) gives rise to distinct balances between those sources of information and requires the consideration of various modes of user interaction.

Keywords example-based texture synthesis, grey-scale image conversion, line drawing generation, pattern recognition, applied perception, machine learning

Extraction d'information pour l'ï¿½dition et la synthï¿½se par l'exemple en rendu expressif

Résumé Le traitement de données graphiques, soit en vue de leur édition ou de la synthèse de nouveaux contenus, requiert un juste équilibre entre les sources d'information que l'on peut exploiter. Contrairement aux techniques "procédurales", l'approche par l'exemple se distingue par sa grande simplicité d'utilisation : reviennent en effet à l'algorithme de synthèse l'identification, analyse et reproduction des éléments caractéristiques des exemples fournis en entrée par l'utilisateur. Ce mode de création de même que les techniques approfondies d'édition ont grandement contribué à la facilitation de la production à grande échelle de contenus graphiques convaincants et ainsi participé à l'adoption par la communauté des artistes des outils proposés par le support numérique. Mais pour être ainsi exploitées, celles-ci doivent également être hautement contrôlables tout en évitant l'écueil de n'être que le simple prolongement de la main de l'artiste. Nous explorons ici cette thématique dans le cadre de la création de rendus dits expressifs et étudions les interactions (collaboratives ou concurrentielles) entre les différentes sources d'information au cœur de ce processus. Ces dernières sont à notre sens au nombre de trois : l'analyse automatique des données d'entrée avant rendu ou traitement ; l'utilisation de modèles a priori en vue de leur compréhension ; et enfin le contrôle explicite par l'utilisateur. En les combinant au plus juste, nous proposons des techniques nouvelles dans divers domaines de la synthèse en rendu expressif. Au delà du réalisme photographique, le rendu expressif se caractérise par sa poursuite de critères plus difficilement quantifiables tels la facilité de compréhension ou le caractère artistique de ses résultats. La subjectivité de tels objectifs nous force donc ici plus qu'ailleurs à estimer avec soin les sources d'information à privilégier, le niveau d'implication à accorder à l'utilisateur (sans que ce choix ne s'opère au détriment de la qualité théorique de la méthode), ainsi que le possible recours à des modèles d'analyse (sans en compromettre la généralité). Trois principales instances de synthèse sont ici détaillées : la génération de textures, la désaturation d'images, et la représentation de maillages par le dessin au trait. La grande variété des données d'entrée (textures matricielles ou vectorielles, images complexes, géométries 3d), des modalités de synthèse (imitation, conversion, représentation alternative) et d'objectifs (reproduction de la signature visuelle d'une texture, restitution crédible de contrastes chromatiques, génération de dessins conformes au style de l'utilisateur) permettent l'examen de divers équilibres entre ces sources d'information et l'exploration de degrés plus ou moins élevés d'interaction avec l'utilisateur.

Mots-clés synthèse de textures par l'exemple, conversion d'images en niveaux de gris, génération de dessins au trait, reconnaissance de motifs, perception appliquée, apprentissage artificiel