Intuitive modeling of 3D objects from outline drawing. Application tothe design of virtual animals.

Even Entem

Communauté
UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Informatique**

Arrêté ministériel : 25 mai 2016

Présentée par

Even ENTEM

Thèse dirigée par **Marie-Paule CANI**
et codirigée par **Loïc BARTHE**

préparée au sein **du Laboratoire Jean Kuntzmann (LJK), de l'Institut de Recherche en Informatique de Toulouse (IRIT)**
dans **l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Interprétation et modélisation 3D automatique à partir de dessins au trait de formes organiques

Automatic interpretation and 3D modeling from line drawings of organic shapes

Thèse soutenue publiquement le **26 octobre 2018**,
devant le jury composé de :

Marie-Paule CANI
Professeur, Grenoble INP, Directeur de thèse
Loïc BARTHE
Professeur, Université Toulouse III Paul Sabatier, Co-Directeur de thèse
Yotam GINGOLD
Associate Professor, George Mason University - USA, Rapporteur
Raphaëlle CHAINE
Professeur, Université Lyon 1, Rapporteur
Pierre BÉNARD
Professeur Associé, Université de Bordeaux, Examinateur
Georges-Pierre BONNEAU
Professeur, Université Grenoble-Alpes, Président

Abstract

Drawing is the most common way to communicate about shapes. Thus, using sketching as a tool in the process of modeling 3D content is an attractive approach. However in the world of machines, drawings are still difficult to interpret as shape depictions. This has been the challenge tackled by many different research works since leveraging the little we know about perception is non trivial.

My thesis focuses on pushing the limits of what can be inferred from single drawings of smooth shapes without any help from the user. In a first attempt we chose to select a category of shape namely animals and other creatures for which prior knowledge helps to solve the problem. Then we proposed to generalize parts of the solution to tackle the case of free form organic shapes. This manuscript thus presents the respective solutions we developed. The first one is able to infer plausible 3D models of animals from a single side-view sketch using anatomic principles to both interpret the drawing's elements and infer depth offsets between these elements. The second is an approach to decompose depictions of smooth shapes with non trivial cusp points into a set of structural parts' silhouettes ordered in depth, which can be used for editing and animation purposes. Many related ideas were explored on the way, and the ones presented in this manuscript leaves me confident about the future of this field of research.

Résumé

Le dessin est la manière la plus courante de communiquer sur les formes. Ainsi, l'utilisation de l'esquisse comme outil dans le processus de modélisation de contenus 3D est une approche attrayante. Cependant, dans le monde des machines, les dessins sont encore difficiles à interpréter comme des représentations de formes 3D. Ce défi a été relevé par de nombreux travaux de recherche, car tirer parti du peu de connaissances que nous possédons sur la perception n'est pas trivial.

Ma thèse se concentre sur les limites de ce qui peut être inféré à partir de dessins uniques de formes lisses sans aucune aide de l'utilisateur. Dans un premier temps, nous avons choisi une catégorie de forme, à savoir les animaux et autres créatures pour lesquels une connaissance préalable aide à résoudre le problème. Ensuite, nous avons proposé de généraliser certaines parties de la solution pour aborder le cas des formes organiques libres. Ce manuscrit présente donc les solutions respectives que nous avons développées. La première permet de déduire des modèles 3D plausibles d'animaux à partir d'une seule esquisse de vue latérale en utilisant des principes anatomiques pour interpréter les éléments du dessin et déduire des décalages de profondeur entre les éléments. La seconde est une approche qui consiste à décomposer les représentations de formes lisses avec des points de recouvrement non triviaux en un ensemble de silhouettes de parties structurelles ordonnées en profondeur, qui peuvent être utilisées à des fins d'édition et d'animation. Beaucoup d'idées connexes ont été explorées en parallèle, et celles présentées dans ce manuscrit me donnent confiance en l'avenir de ce domaine de recherche.

Remerciements

Je remercie mille fois mes encadrants : Marie-Paule CANI et Loïc BARTHE, pour l'aide précieuse qu'ils m'ont apportés tout au long de cette aventure que ce soit sur le plan scientifique que personnel.

Je remercie Frédéric CORDIER, Cédric ZANNI, Michiel van de PANNE, John HUGHES et Amal Dev PARAKKAT pour leurs conseils et l'aide apportée lors des soumissions.

Je remercie les membres de mon jury : Raphaëlle CHAINE, Yotam GINGOLD, Pierre BÉNARD et Georges-Pierre BONNEAU, pour leur gentillesse lors de l'organisation de cette fin de thèse.

Je salue et souhaite réussite et bonheur à toutes les personnes qui me connaissent au sein de l'IRIT de Toulouse et de l'Inria de Grenoble.

Je remercie les collègues avec qui j'ai eu des discussions d'une qualité difficile à égaler en dehors de cet environnement de recherche et avec qui j'ai aussi pu décompresser en dehors du travail : Florian CANEZIN, Grégoire NIETO, Geoffrey GUINGO, Estelle (les deux) et Anahid.

Je remercie mes voisins et amis de Toulouse, notamment Dr. Vincent COURJAULT-RADÉ et Dr. Nathan MINOIS qui ont supportés mes sautes d'humeur lors de mes overdoses de caféine. Profitez, je ne sais pas si tout le monde prendra la peine d'écrire Dr. en vous citant !

Je remercie Coralie qui a partagé ma vie de doctorant pendant 2 ans.

Je remercie ma famille qui a su m'apporter le soutien nécessaire malgré l'éloignement.

Encore un grand merci à tous ceux que j'ai cité ou qui se reconnaissent.

Contents

1 Introduction **1**

1.1 Motivations 1
1.2 Goals 5
1.3 Contributions 7
1.4 Publications 8
1.5 Organization of this manuscript 9

2 Introduction (FR) **11**

3 Background and Related Work **21**

3.1 Terminology of Sketches 21
3.2 Perceptual psychology 23
3.2.1 Hoffman's rules 25
3.3 Shape interpretation 26
3.3.1 Sketch simplification 26
3.3.2 Interpretation from contours 28
3.3.3 Interpretation from silhouette 34
3.4 Sketch-based modeling 36
3.4.1 Sketch-based 3D modeling techniques 36
3.5 Modeling depth-augmented 2D sketches 40
3.6 Conclusion 42

4 Modeling 3D animals from a side-view sketch **43**

4.1 Introduction 43
4.2 Overview 45
4.3 Structural parts identification and completion 47

4.3.1 Representing the sketch as a set of cycles . . . 47
4.3.2 Contour completion . . . 49
4.4 Depth hierarchy of structural parts . . . 51
4.4.1 Body and symmetrical parts . . . 51
4.4.2 Graph construction . . . 52
4.5 3D Reconstruction . . . 52
4.5.1 Medial-axis computation & simplification . . . 52
4.5.2 Surface reconstruction of the shape components . . . 54
4.5.3 Placement & embedding of the shape components . . . 54
4.6 Results and discussion . . . 55
4.7 Future Works . . . 59
4.8 Conclusion . . . 60

5 Structuring and Layering Contour Drawings of Organic Shapes 61
5.1 Introduction . . . 61
5.2 Overview . . . 63
5.2.1 Terminology and assumptions . . . 64
5.2.2 Processing pipeline . . . 65
5.3 Aesthetic and efficient contour completion . . . 68
5.3.1 Scale-Invariant MVC . . . 68
5.3.2 Efficient implementation . . . 69
5.4 Extraction of Salient Junctions within a Part . . . 70
5.4.1 Salient Junctions . . . 70
5.4.2 Radial Variation Metric . . . 71
5.4.3 Salience junction detection . . . 74
5.5 Recursive part decomposition . . . 75
5.5.1 Prioritizing salient junctions . . . 75
5.5.2 Processing complex suggestive contours . . . 76
5.5.3 Part decomposition method . . . 77
5.5.4 Contour / Suggestive contour (C,SF) closure . . . 78
5.5.5 Contour / Contour (C,C) closure . . . 80
5.6 Results and discussion . . . 80
5.7 Future Works . . . 84
5.8 Conclusion . . . 84

6 Conclusion 89
6.1 Future Directions . . . 90

7 Conclusion (FR) 95

Bibliography 97

CHAPTER 1

Introduction

Sketch-based modeling techniques aim at providing intuitive solutions using 2D sketching as input for the difficult task of 3D modeling. This research field is an answer to the demand of an ever growing industry and communities of digital world enthusiasts. Many industries which used to build physical models in various design processes have replaced them with digital solutions and now look for more productivity. The current industrial pipelines are often using a refinement cycle in which many 3D concept models are simply abandonned and directors' demands are subject to misunderstandings. The communities of digital world enthusiasts including the "Do it yourself" (DIY) movement, 3D-printers owners, and independent game developers, are rapidly limited by the complexity of standard 3D modeling software and the small number of free 3D models shared on the Web. This new demands can be solved by novel 3D modeling techniques oriented toward intuitiveness and robustness, such as those based on sketching. Let us take a step back to better motivate the need for these new techniques.

1.1 Motivations

Nowadays 3D content is part of our daily lives. We find computer graphics images in movies, commercials, video games, and even on business cards. More recently, virtual reality arose and is on its way to make our dreams look like real experiences. Where photorealism was a limit, we now have rendering techniques that are sometimes considered better looking than reality is. In essence, virtual worlds could give far more freedom for creation and imagination than reality provides. Images have always been a strong communication medium, and this is even more true since we can show things that do not exist but in our minds.

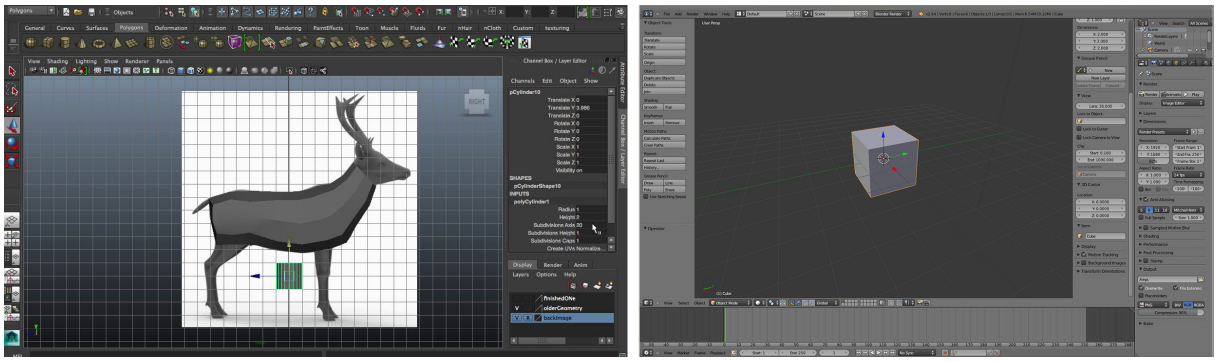


Figure 1.1 – The complex interfaces of standard modeling software: Autodesk Maya (left, commercial), Blender (right, open source). We can note a slightly simpler interface in Blender but most tools are also available and hidden in menus.

However there is a main problem, a bottleneck to this technological leap, which is the fact that the required 3D models are difficult to produce. Thus the quantity of 3D contents that is currently available and reusable is still too small to provide the variety of shape that would satisfy most requests. The fact is most 3D models are currently designed by highly trained and skilled artists who work on current well-known modeling tools such as Autodesk Maya, Autodesk 3ds Max, Blender, ZBrush, and MudBox. This content is often kept in the private sector and will eventually never be shared. While the free content is often of poor quality and the content for sale is not worth the value for non commercial use.

The main reason to this difficulty to produce 3D models is that standard 3D modeling software are hard to use and unintuitive. They require a user to constantly keep a mental image of the geometry he is working on in order to manipulate the surface components or vertices as he intends to. Manipulating a polymesh requires the understanding of a range of specific tools that are not yet intuitive and often hidden in between a huge quantity of other highly specific tools as seen in Figure 1.1.

Usually, the first thing 3D modelers do is to start from simple shapes like cylinders and cubes and deform them to match a set of concept arts aligned in different views. This involves splitting triangles where needed, filling gaps, extruding faces, and all of this while maintaining a reasoned density of triangles. Then, even if this model looks nice in a static position, changing its pose is another challenge and usually induces artifacts if the topology of the mesh has not been designed accordingly. Thinking of this kind of problems in advance is a matter of experience and far from being intuitive. Although the learning curve on standard software is manageable to some extent for quick learners, the steep end remains a source of despair. Sculpting software are a step toward more intuitive modeling since the intent behind sculpting brushes is more clear than manipulating vertices in side views. Unfortunately, the user still needs specific skills to handle common problems such as polygon density artifacts, which can appear quickly. This issue could be solved with

the use of implicit surfaces which represent a good alternative to meshes but also bring a number of disadvantages. For instance these surfaces have relatively long rendering times, embeds a complex underlying tree of blending operators, and are not trivially texturable. Until robust hybrid representations develops, scultping solutions will not scale well to animation purposes nor allow users to retroactively modify their creation. For now the different actors of the 3D content creation world still face the constraints of current software.

On one side, the communication and entertainment industries managed to use the third dimension to provide high quality and expressive contents through specific production pipelines. One of their early phase of artistic production is drawing concept arts. These are early sketches of ideas following an artistic goal, and are used to explore the design possibilities. To this end, multiple views of one concept are often required for a better understanding of volumes prior to creating coarse 3D models. A noticeable innefficiency is that most of these drawings and coarse models are discarded and never reused because they do not fit the artistic director's vision. We believe that the concept art phase remains a time consuming task. It could be improved by techniques leading to fast 3D concept models which are more adapted to rapidly verify that ideas are consistent with the artistic direction.

On the other side, the general public recently expressed their frustration at facing the complexity of 3D content creation. This social phenomenon became really noticeable with the arrival of ready-to-use and user-friendly game engines as well as affordable 3D printers which brought the need for 3D models to the public and the desire to make their own. It has been encouraging researchers to develop smart modeling tools in order to improve their accessibility. Inspiring from the standard industrial pipeline where 2D concept arts are made before 3D models, a range of approaches called sketch-based modeling techniques have been proposed. The idea is to consider a user with basic drawing skills and let her use sketching to drive the 3D modeling process. These approaches consider different types of input in order to manage ambiguities and obtain the desired fidelity of the output model with respect to the sketch. They can be designed for either single-view or multi-view sketches, for either industrial design or free-form shapes, and are either automatic or require user assistance.

Drawings as communication media

Line drawings were and are still commonly used to depict concepts and shapes and thus is an interesting input medium to study. Drawing is a visual language that develops as early as the age of 3, in parallel with spoken language. However, opposed to a spoken language which children develop by hearing people speaking in their surroundings, drawing is developed based on what they see and the emerging structural nature of their thoughts.

Since we all mostly see the same things we tend to develop the same visual language as any other human, making line drawing a reasonnably universal language. We discuss why it is only "reasonnably" universal and why the drawing process avoids any trivial hypothesis to be formulated without knowing how humans actually perceive drawings in Section 3.2.

This language nonetheless is practiced at different levels in the population. But although realistic drawing is not mastered by most adults, we can still draw structured depictions of shapes called schematic line drawings or sketches. Moreover, with the critical thinking necessary to select important features, we are often able to produce humanly unambiguous drawings, avoiding misinterpretations from whoever looks at it. We emphasize that even if there are disparities in the interpretation of features in a complex line drawing of free-form 3D shape, the intended structure is often properly carried. Notably, these disparities can be alleviated with a form of prior-knowledge associated with the depicted shape.

While understanding a drawing is easy for humans, it is still a complex task to even define what "understanding" means in terms of computer algorithms. This difficulty appears because most of the fundamentals of this visual language are not properly rationalizable, it remains an highly instinctive and intuitive way to communicate. Even for a spoken language, recent algorithms cannot properly tackle more complexity than short stories with no meaning hidden "between the lines". Our thoughts are made of words or simple shapes, but while there is a well defined set of symbols, a dictionary and grammar that rule the building and understanding of sentences, there is no such thing for complex drawings. Moreover, the intent is often contextualized and requires knowledge of this context to be understood. Shape recognition from pictures is an active field in Computer Vision, but the use of deep neural networks, while supporting the theory of a collection of intermediate concepts driving our perception, does not trivially suggest a solution for the computational understanding of pictures and even less line drawings. In fact, these solutions can be tricked to recognize objects in designed noise or non sense images. It let us suppose that we are still far from achieving a real resilience and consistency without any semantic model with induction and deduction capabilities to cope with unresolved ambiguities. Some kind of intermediate subconscious representations enables new concepts to be understood, and visualized. We are not aware of intermediate processes ourselves as simple experiments prove that we most often perceive a whole concept before we realize its sub-components. For example in Figure 1.2 one can see a horse and its rider before one realizes where are the legs, arms and other components. In other words we do not check that there are four legs to know that it is a horse, we magically know it in an instant.

This ability of the human brain is not understood well enough yet in order to be used as a model for sketch-based modeling techniques. We are still far from being able to actually mimic the way we mentally see 3D shapes from 2D sketches. However some theories

Figure 1.2 – Perceptual experiment in which we become aware of the concept (horse and its rider) before we realize the visual sub-components. Splats are placed where the original contours showed maxima of curvature. This example shows that it is not the mental listing of sub-components that enables us deduce the nature of the whole entity.

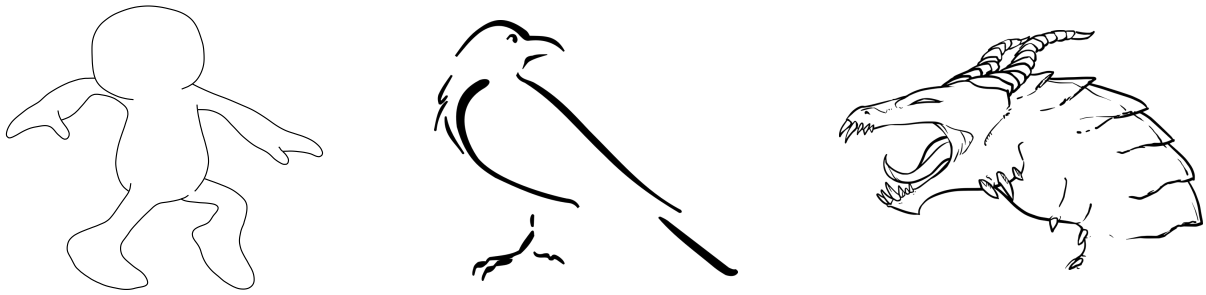


Figure 1.3 – Some examples of complex line drawings. The character drawing (left) is from *Modeling Character Canvases from Cartoon Drawings* [Bessmeltsev et al., 2015].

identified some basic visual principles that have shown to be consistent across participants to different perceptual experiments. Most automatic algorithms in sketch-based modeling use these principles, starting with the well-known Gestalt Principles [Koffka, 1935]. Applying these in the general case is non trivial as one can judge by looking at Figure 1.3.

1.2 Goals

This thesis not only addresses the interpretation of complex line drawings, but also their use to create 3D models. The under-constrained nature of the 2D to 3D goal as well as the current knowledge on human perception, puts our work between hard and soft sciences. In this regard, I will present as well as I can the intuitions and supporting facts behind some aspects of the techniques that have sometimes been blindly judged as ad-hoc solutions instead of reasonable educated guesses.

We oriented our work based on an analysis of the state of the art from Computer Vision to Computer Graphics, while aiming for sketch-based 3D modeling of animals and other creatures. Sketch-based modeling approaches can be grouped into different categories by identifying their specific constraints:

1. **Type of algorithm:** Iterative or User-assisted / Automatic
2. **Number of inputs:** Single view / Multiple views
3. **Viewpoint:** Orthogonal / Free
4. **Type of the depicted object:** 2D inter-woven surfaces / Computer Aided Design solids / Smooth manifold-solids / Conceptual characters (Cartoons)
5. **Prior knowledge:** Generic / Specific / Database

Concerning the type of algorithm, we noted that iterative or user-assisted algorithm have flourished during the last decade and already provide interesting solutions, and this for all input types. With this in mind, we chose to stay as close as possible to fully automatic solutions while considering user-assistance for specific problems. This kind of algorithm involves many fields at its roots such as sketch segmentation, hidden contour completion, and 3D inference from 2D cues, which are difficult to combine in a single algorithm.

Drawing multiple views of a smooth shape is not an intuitive task, thus we chose to work on single side-views that often are the most explicit depictions of a creature's structure. To avoid any unnecessary and non-trivial additional challenge we considered clean line drawings with no hatching, and no oversketched strokes since these can be cleaned by other techniques.

Challenges

With these research goals, the major challenge of this thesis revealed to be the gap in terms of literature between the automatic completion of drawings and the 3D inference. On one side there are techniques designed to complete the occluded contours of anterior surfaces or shapes with simple cusps. On the other side there are techniques infering 3D from drawings that either use user annotations or only work for a restricted domain of smooth shapes. Connecting both into a single approach has been tackled for cusps by [Karpenko and Hughes, 2006] but still requires a user annotated-input to handle holes, and process cusps locally. Without user assistance, there is a trade-off to choose between restricting the domain and generalizing prior knowledge. We chose to start with category-specific shapes (animals), then we tried to reduce the constraints on the drawing and generalize the prior knowledge in order to tackle general smooth solids.

Another identified challenge is that general line drawings embed a lot of ambiguities that are hard to visually detect at first sight and are non-trivial to identify in labeled contour graph diagrams.

1.3 Contributions

Our research has led to different contributions toward fully automatic sketch-based modeling of general smooth shapes. These contributions are part of two automatic techniques, one focused on 3D modeling of animals using prior knowledge and the second on a 2.5D representation of more complex drawings with internal chains of cusp-ended contours and internal strokes. We now briefly present the contributions in their order of appearance in this thesis.

Modeling 3D animals from a side-view sketch (Chapter 4)

This worked focused on 3D inference from drawings of animals with structural symmetry. The input is restricted to the kind of side-view drawings one can find in children books and can easily draw from a picture. Prior knowledge on animals led us to differ from the previous work on cusps [Karpenko and Hughes, 2006] and to look for structural contours instead of only hidden contours. Then we lift this structure into a 3D shape.

Structural analysis of line drawings of animals

The main part of this contribution is the global optimization of an energy related to contour completion, tackling the problem of noisy junctions of contours that are non-trivially avoidable. We also proposed a solution to detect partially occluded background parts which are possibly structural symmetries with a twin part in the foreground (e.g.: legs and ears) based on simple assumptions about the anatomy of animals. This contribution is presented in Section 4.3.

Generation of a 3D shape from a structured drawing of animal

To this end we propose to use Implicit Surfaces, enabling users to easily articulate the resulting animal into new poses. The skeleton is computed using a new simplification algorithm on the topological skeleton of each body part (e.g.: head, eye, legs). The resulting surfaces and their respective skeletons are then assembled in hierarchical order from the body to the extremeties. We infer depths using an anatomical rule that fits most vertebrates and is presented in Section 4.5.

Structuring and Layering Contour Drawings of Organic Shapes (Chapter 5)

We proposed in this work to reduce the constraints on the drawing topology as well as to use more general prior knowledge on drawings of smooth manifold-solids. However we only achieved a 2.5D structural representation adapted to 2D animation but not directly usable for 3D inference.

An aesthetic contour completion algorithm with non explicit end points on contours. This is a requirement for our way of structuring drawings into sets of closed regions with smooth contour completion. We inspired from drawing methods that use an underlying structure of potato-shaped primitives to draw creatures (as seen in Figure 5.2). We present our solution and discuss different variants that we explored in Section 5.3.

A radial variation metric to identify structural connection zones based on the difference of distances to contours from neighbor point of views located inside the 2D shape. This delta represents a speed of transition and is in fact the first derivative of our metric. Since shape segmentation relies on the latter, we directly use our delta function for such purpose. Although this speed is made independent from the scale of the drawing, its numerical nature still implies a parameter for precision. We discuss solutions to spare the user from setting this parameter in Section 5.4.2. We present this metric as well as how we use it to identify structural parts in Section 5.4.

A recursive algorithm for structural decomposition of a complex drawing into structural parts, using the junction zones identified by the latter contribution. The depth hierarchy is kept to enable easy manipulation of the resulting set of parts. We discuss the advantages and limitations to this approach in Section 5.5.

1.4 Publications

1. Entem, E., Barthe, L., Cani, M.-P., Cordier, F., and Van De Panne, M. (2015). Modeling 3D animals from a side-view sketch. *Computers and Graphics*, 46:221–230. Shape Modeling International 2014

2. Cordier, F., Gingold, Y., Entem, E., Cani, M.-P., and Singh, K. (2016). Sketch-based Modeling. In *Eurographics 2016*, number T7, Lisbonne, Portugal. The Eurographics Association, The Eurographics Association

3. Entem, E., Barthe, L., Cani, M.-P., and van de Panne, M. (2016). From drawing to animation-ready vector graphics. In *ACM SIGGRAPH 2016 Posters*, SIGGRAPH '16, pages 52:1–52:2, New York, NY, USA. ACM
   An earlier version of this poster has been presented at Loisirs et IMAges 2015 in Lyon.

4. Entem, E., Parakkat, A. D., Cani, M.-P., and Barthe, L. (2018). Structuring and layering contour drawings of organic shapes. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*, Expressive '18, pages 4:1–4:14, New York, NY, USA. ACM
   Video material of this paper is available on the web:

http://www.youtube.com/watch?v=MrZuQ5g4Vmo

1.5 Organization of this manuscript

I propose to start this thesis manuscript with a presentation of the relevant background knowledge as well as the related work in Chapter 3. Then I present our contributions in the two projects which led to international publications in Chapters 4 and 5. Finally I will conclude and present some work in progress and future works in Chapter 6.

CHAPITRE 2

Introduction (FR)

Les techniques de modélisation 3D par ésquisse sont développée dans l'objectif de fournir des outils intuitifs pour la création de contenu 3D via une interface 2D. L'existence de ce domaine de recherche est une réponse à un besoin, celui de pouvoir créer du contenu plus facilement et rapidement. Ce besoin est partagé par l'industrie croissante du cinéma d'animation et du jeux-vidéo ainsi que des communautés d'amateurs voulant concrétiser leur imaginaire. Je pense notamment au mouvement "Do it yourself" (DIY), aux possesseurs d'imprimantes 3D et aux développeurs de jeux-vidéo indépendants. Dans une autre mesure, les industries qui précédemment utilisaient des maquettes physiques dans leur processus de design se sont pour la plupart converties au virtuel et recherche toujours plus de productivité : l'industrie automobile, l'immobilier et d'autres. Les *pipelines* industriels actuels définissent souvent un système dans lequel les modèles 3D sont revus et améliorés par itérations successives. En revanche les directeurs artistiques ne sont pas toujours capables de manipuler eux-mêmes les outils et donc certains projets sont simplement abandonnés. Du côté des amateurs, la complexité des logiciels de modélisation 3D courants représente un facteur très limitant. Les bibliothèques de modèles libres de droits sont très petites et il est rare d'y trouver son bonheur. Ces nouvelles demande en terme de création de contenu pourrait être résolu grâce à des solutions de modélisation 3D conçus en privilégiant l'intuitivité pour l'utilisateur. Lorsque l'on veut exprimer une forme on pense souvent en premier à prendre une feuille et un crayon, et ce constat est la première motivation des techniques de modélisation par esquisse.

Motivations

De nos jours le contenu 3D fait partie intégrante de nos vies. On retrouve des images de synthèses dans les films, les publicités, les jeux-vidéos et même sur les cartes de visite. Plus récemment la réalité virtuelle est apparue pour le grand public et finira peut-être par être présente dans tous les foyers. Là où le photo-réalisme était une limite nous avons maintenant des rendus qui paraissent étrangement plus naturels et beaux que la réalité elle-même. Par nature, les mondes virtuels pourraient donner beaucoup plus de liberté pour la création et l'imagination que la réalité ne l'offre. Les images ont toujours été un moyen de communication puissant, et cela est encore plus vrai depuis que nous pouvons montrer des choses qui n'existent pas ailleurs que dans notre esprit.

Cependant, il existe un problème majeur, un obstacle à ce saut technologique, à savoir que les modèles 3D requis sont difficiles à produire. Ainsi, la quantité actuellement disponible et réutilisable est encore trop faible pour fournir la variété de formes qui satisferait la plupart des demandes. Le fait est que la plupart des modèles 3D sont actuellement créés par des artistes hautement qualifiés et qui travaillent sur des outils de modélisation complexes bien connus tels qu'Autodesk Maya, Autodesk 3ds Max, Blender, ZBrush et MudBox. Ce contenu est souvent conservé dans le secteur privé et ne sera jamais partagé. Malheureusement le contenu gratuit est donc souvent de mauvaise qualité et les modèles payants sont trop onéreux pour la plupart des usages non commerciaux.

La principale raison de la difficulté à produire des modèles 3D est que les logiciels de modélisation 3D standards sont très peu intuitifs. Ils exigent de l'utilisateur qu'il garde constamment une image mentale de la géométrie sur laquelle il travaille afin de manipuler les composants de surface ou les sommets comme il le souhaite. Manipuler un maillage 3D nécessite la compréhension d'une gamme d'outils spécifiques souvent cachés entre une grande quantité d'autres outils hautement spécifiques, comme le montre la Figure 1.1.

Habituellement, la première étape pour un artiste 3D consiste à commencer avec des formes simples telles que des cylindres et des cubes et ensuite les déformer pour les adapter à un ensemble de dessins appelés *concept arts* alignés dans différentes vues. Cela implique de séparer des triangles là où cela est nécessaire, de combler les vides, d'extruder les faces et tout cela tout en maintenant une densité raisonnée de triangles. Ainsi, même si un modèle a l'air correct dans une position définie, changer sa posture pour l'animer est un autre défi et induit généralement des artefacts si la topologie du maillage n'a pas été conçue en conséquence. Penser à ce type de problèmes à l'avance est une question d'expérience et loin d'être intuitive. Bien que la courbe d'apprentissage des logiciels standards soit dans une certaine mesure gérable pour les plus talentueux, sa fin abrupte demeure une source de désespoir. Les logiciels de *scultping* (sculpture) sont une étape vers une modélisation plus intuitive, car le comportement des outils est plus clair que la manipulation des sommets dans des vues latérales. Malheureusement, un utilisateur qualifié est toujours nécessaire pour gérer certains problèmes qui peuvent apparaître rapidement tels qu'une densité de

polygones non adaptée. Ce problème pourrait être résolu par l'utilisation de surfaces implicites, qui représentent une bonne alternative aux maillages, mais présentent également des inconvénients. Par exemple, ces surfaces ont des temps de rendu relativement longs, incorporent une arborescence sous-jacente complexe d'opérateurs de fusion et ne peuvent pas trivialement être associé à une texture. Tant que des représentations hybrides robustes ne seront pas développées, les solutions de *scultping* ne seront pas adaptées aux besoins de l'animation et ne permettront pas aux utilisateurs de modifier rétroactivement leur création. Pour l'instant, les différents acteurs du monde de la création de contenu 3D sont toujours confrontés aux contraintes des logiciels actuels.

D'un côté les industries de la communication et du divertissement ont réussi à utiliser la troisième dimension pour fournir des contenus de haute qualité et expressifs à travers des pipelines de production spécifiques. L'une des premières phases de la production artistique dans ces industries consiste à dessiner des concepts artistiques. Ce sont des esquisses qui suivent un objectif artistique et sont utilisées pour explorer les possibilités de conception. À cette fin, plusieurs vues d'un concept sont souvent nécessaires pour mieux comprendre les volumes avant de créer des modèles 3D bruts. Une inefficacité notable est que la plupart de ces dessins et modèles initiaux sont rejetés et ne sont jamais réutilisés car ils ne correspondent pas à la vision du directeur artistique. La phase conceptuelle reste ainsi une tâche chronophage. Elle pourrait être améliorée par des techniques conduisant à des modèles 3D rapides, plus adaptés à la vérification rapide d'idées.

Le grand public, lui, a récemment exprimé sa frustration face à la complexité de la création de contenu 3D. Ce phénomène social s'est fait sentir avec l'arrivée de moteurs de jeux prêts à l'emploi et conviviaux, ainsi que d'imprimantes 3D abordables, qui ont amené le public à avoir besoin de modèles 3D et à créer les leurs. Ce phénomène encourage les chercheurs à développer des outils de modélisation intelligents pour améliorer leur accessibilité. S'inspirant du pipeline industriel standard où les *concept arts* sont élaborés avant les modèles 3D, diverses approches appelées techniques de modélisation par esquisse ont été proposées. L'idée est de considérer un utilisateur avec des compétences de base en dessin et de le laisser utiliser le dessin pour piloter le processus de modélisation 3D. Les techniques que l'on retrouve dans la litérature scientifique prennent en compte différents types d'intéractions afin de gérer les ambiguïtés et d'obtenir la fidélité souhaitée du modèle de sortie par rapport à l'idée qu'en avait l'utilisateur. Elles peuvent aussi être conçus pour des esquisses à vue unique ou à vues multiples, adaptées à des formes spécifiques ou aux formes libres, et sont automatiques ou nécessitent une assistance utilisateur.

Le dessin, un moyen de communiquer

Les dessins au trait étaient et sont encore couramment utilisés pour décrire des concepts et des formes et constituent donc un moyen de communication intéressant à étudier. Le

dessin est un langage visuel qui se développe dès l'âge de 3 ans, parallèlement à la langue parlée. Cependant, contrairement à une langue parlée qu'un enfant développe en entendant la ou les langues parlées utilisées dans son environnement, le dessin est développé en fonction de ce qu'il voit et de la nature structurelle émergente de ses pensées. Comme nous voyons tous la plupart du temps les mêmes choses, nous avons tendance à développer le même langage visuel que n'importe quel autre humain, faisant du dessin au trait un langage raisonnablement universel. Nous discutons dans la Section 3.2 de la raison pour laquelle il est seulement «raisonnablement» universel et pourquoi le processus de dessin prévient tout formalisme trivial dans l'impossibilité de savoir comment les humains les perçoivent réellement.

Notons que la partie "écriture" de ce langage est pratiquée à différents niveaux dans la population contrairement à sa "lecture". En revanche, bien que la plupart des adultes ne maîtrisent pas le dessin réaliste, nous pouvons toujours dessiner des représentations structurées de formes appelées schémas ou croquis. De plus, nous avons à l'âge adulte la pensée critique nécessaire pour sélectionner les caractéristiques importantes et nous sommes ainsi souvent en mesure de produire des dessins sans ambiguïté. Cela évite les erreurs d'interprétation de quiconque les regarde. Je pense que même s'il existe des disparités dans l'interprétation des caractéristiques d'un dessin au trait complexe d'une forme 3D libre, la structure voulue est souvent correctement transportée. Dans certains cas les disparités peuvent être atténuées par une forme de connaissance préalable associée à la forme représentée. Il ne nous suffit par exemple de deux grandes oreilles et deux dents pour identifier un lapin et rendre cohérent le reste du dessin.

Bien que la compréhension d'un dessin soit facile pour les humains, il est encore complexe de définir ce que signifie «comprendre» en termes d'algorithmes informatiques. Cette difficulté apparaît au vu de la plupart des fondamentaux de ce langage visuel qui ne sont pas correctement rationalisables. C'est une manière très instinctive et intuitive de communiquer, et donc inconsciente et très difficile à formaliser. Même dans le cas d'une langue parlée, les algorithmes récents ne peuvent pas traiter correctement beaucoup plus que des histoires courtes sans signification cachée "entre les lignes". Nos pensées sont faites de mots ou de formes simples, mais s'il existe un ensemble bien défini de symboles, un dictionnaire et une grammaire qui régissent la construction et la compréhension des phrases, les dessins complexes ne sont pas de ce type. De plus, l'intention est souvent contextualisée et nécessite une connaissance de ce contexte pour être comprise. La reconnaissance des formes à partir d'images est un domaine actif de la vision par ordinateur, mais l'utilisation de réseaux neuronaux profonds, tout en soutenant la théorie des concepts intermédiaires qui animent notre perception, ne suggère pas encore l'existence d'une solution simple et élégante pour la compréhension informatique des images et encore moins des dessins au trait. En effet, ces algorithmes reconnaîssent par erreur des objets dans des images de bruit ou dénuées de sens. Cela laisse supposer que nous sommes encore loin d'un modèle

sémantique intermédiaire et d'une technique capable de rivaliser l'homme avec des capacités d'induction et de déduction pour à résoudre les ambiguïtés. Je pense qu'il existe une représentation subconsciente intermédiaire qui permet de comprendre et de visualiser de nouveaux concepts. Malheureusement ce processus intermédiaire nous échappe comme le fait que nous percevons le plus souvent un concept complet avant de réaliser ses sous-composants le suggère. Par exemple, dans la Figure 1.2 on peut voir un cheval et son cavalier avant de réaliser où sont les jambes, les bras et les autres parties. En d'autres termes, nous ne vérifions pas qu'il y a quatre pattes pour savoir que c'est un cheval, nous le savons comme par magie en un instant.

La capacité de perception du cerveau humain n'est pas encore suffisamment comprise pour être utilisée comme modèle pour les techniques de modélisation par esquisse. Cependant, certaines théories ont identifié certains principes visuels de base qui se sont révélés cohérents entre les participants dans différentes expériences perceptuelles. La plupart des algorithmes automatiques utilisés dans la modélisation par esquisse utilisent ces principes, à commencer par les principes de Gestalt [Koffka, 1935]. Les appliquer dans le cas général n'est pas trivial comme on peut en juger à la vue de la Figure 1.3.

Objectifs

Cette thèse traite non seulement l'interprétation de dessins au trait complexes, mais aussi leur utilisation pour création de modèles 3D. La nature sous-contrainte de l'inférence de contenu 3D à partir de d'esquisse 2D, ainsi que les connaissances actuelles sur la perception humaine, placent notre travail entre les sciences dures et les sciences dites "molles". À cet égard, je présente dans ce manuscrit les intuitions et les faits à l'appui des solutions techniques proposées qui ont quelque fois été jugées péjorativement comme ad hoc.

Nous avons défini l'orientation de mon travail à partir d'une analyse de l'état de l'art de la vision par ordinateur à l'infographie 3D, tout en visant dans un premier temps un système de modélisation 3D à partir d'esquisses d'animaux et d'autres créatures. Les approches pour la modélisation par esquisse peuvent être regroupées en différentes catégories aux contraintes spécifiques :

1. **Type d'algorithme :** Interactif / Automatique
2. **Nombre d'esquisses nécessaire :** Un seul point de vue / Plusieurs points de vue
3. **Nature des plans de vue :** Orthogonal / Libre / Sagittal
4. **Type d'objet dessiné :** Surfaces 2D entrelacées / Objet de design ou industriel / Solide manifold lisse / Personnages conceptuels (cartoon, BD)
5. **Connaissance *a priori* :** Générique / Spécifique / Base de donnée

En ce qui concerne le type d'algorithme, nous avons noté que les algorithmes interactifs aussi appelé itératifs ou assistés par l'utilisateur ont prospérés au cours de la dernière décennie et déjà fourni des solutions intéressantes pour tous types d'objets. Sur ce constat, nous avons choisi de rester au plus près des solutions entièrement automatiques tout en se réservant la possibilité de demander l'assistance de l'utilisateur pour des problèmes spécifiques. Ce type d'algorithme implique de nombreux champs tels que la segmentation des esquisses, la complétion des contours cachés et l'inférence 3D à partir d'éléments 2D, difficiles à combiner dans un seul algorithme.

Dessiner des vues multiples d'une forme lisse n'est pas une tâche intuitive, nous avons donc choisi de travailler dans un premier temps sur des vues latérales uniques qui représentent souvent le plus explicitement possible la structure d'une créature. Pour s'épargner un défi supplémentaire inutile et non trivial, nous avons considéré des dessins avec des courbes nettes et sans hachures. Les dessins avec des traits hésitants, des croisillons peuvent être préalablement nettoyés par d'autres techniques pour satisfaire nos contraintes.

Défis

Au vu de ces objectifs de recherche, le défi majeur de cette thèse s'est révélé être la lacune en termes de littérature entre la complétion automatique des dessins et l'inférence 3D. D'un côté, il existe des techniques conçues pour compléter les contours cachés des surfaces antérieures ou des formes simples. De l'autre côté, il existe des techniques déduisant la 3D à partir de dessins qui utilisent des annotations utilisateur ou ne fonctionnent que pour un domaine restreint de formes lisses. Connecter les deux en une seule approche a été abordé par [Karpenko and Hughes, 2006] mais nécessite toujours une entrée annotée par l'utilisateur pour désambiguïser les trous et ne sais traiter qu'une catégorie particulière de formes libres, sans pour autant pouvoir formaliser cette dernière. Sans l'aide de l'utilisateur, il faut choisir entre restreindre le domaine et généraliser les connaissances antérieures. Nous avons choisi de commencer avec des formes spécifiques à une catégorie (animaux). Puis nous avons essayé de réduire les contraintes sur le dessin et de généraliser les connaissances antérieures sur les solides lisses. Cela a abouti à une vision différente mais peut-être complémentaire de la technique proposée par [Karpenko and Hughes, 2006]. Un autre défi identifié réside dans le fait que les dessins au trait en général intègrent de nombreuses ambiguïtés difficiles à détecter visuellement ou consciemment et non triviales à identifier dans un graphe de contours.

Contributions

Nos recherches ont conduit à différentes contributions vers une future modélisation entièrement automatique basée sur des esquisses de formes lisses générales. Ces contributions font partie de deux techniques automatiques, l'une centrée sur la modélisation 3D d'animaux utilisant des connaissances préalables et la seconde sur une représentation 2.5D de dessins plus complexes avec des chaînes internes de contours partiellement visibles et des traits internes ambigus. Ci-dessous une brève présentation des travaux et des contributions respectives dans leur ordre d'apparition dans ce manuscrit.

Modeling 3D animals from a side-view sketch (Chapitre 4)

Ce travail porte sur l'inférence 3D à partir de dessins d'animaux avec symétrie structurelle. L'entrée utilisateur se limite aux dessins que l'on peut trouver dans les livres pour enfants et peut facilement être dessiné à partir d'une image. Des connaissances préalables sur les animaux nous ont amenés à différer des travaux antérieurs et à rechercher des contours structurels au lieu de contours cachés. Ensuite seulement nous inférons un modèle 3D à partir de cette structure.

Analyse structurelle de dessins d'animaux vus de côté

L'essentiel de cette contribution est l'optimisation globale d'une énergie liée à la complétion des contours qui prend en compte les jonctions de contours bruitées. Nous avons également proposé une solution pour identifier les parties en arrière plan partiellement cachées ayant une partie jumelle au premier plan. Cette symétrie structurelle est identifiée grâce à des hypothèses simples qui approximent l'anatomie animale. Cette contribution est présentée dans la Section 4.3.

Génération de modèle 3D à partir de dessins structurés d'animaux

À cette fin, nous proposons d'utiliser des surfaces implicites, permettant aux utilisateurs d'articuler facilement l'animal résultant dans de nouvelles poses. Chaque animal est construit comme une combinaison des surfaces implicites générées indépendamments pour chaque partie du corps (par exemple : la tête, les yeux, les jambes). Au squelette topologique de chacune de ces dernières on applique un algorithme de simplification afin d'obtenir le squelette d'une surface implicite. Les surfaces résultantes sont ensuite assemblées et mélangées dans un ordre hiérarchique, du corps aux extrémités. Nous déduisons les profondeurs à l'aide d'une hypothèse anatomique présentée dans la Section 4.5 qui convient à la plupart des vertébrés.

Structuring and Layering Contour Drawings of Organic Shapes (Chapitre 5)

Nous proposons dans cet article de réduire les contraintes sur la topologie du dessin, ainsi que d'utiliser des connaissances *a priori* plus générales sur les dessins de solides lisses que nous appelons aussi *formes organiques*. Cependant, nous n'avons atteint qu'une représentation structurelle 2.5D adaptée à l'animation 2D mais non trivialement utilisable pour de l'inférence 3D.

Un algorithme rapide pour de la complétion esthétique de contours structurels avec sommets non explicites. Il est adapté à notre approche de la complétion qui vise à structurer le dessin en un ensemble de contours fermés correspondant à des primitives lisses et imaginaires composant la forme organique dessinée. Nous nous sommes inspirés des techniques de dessin qui utilisent une structure sous-jacente de "patatoïdes" pour dessiner des créatures (c.f. Figure 5.2). Nous présentons notre algorithme de calcul de courbes de complétion dans la Section 5.3. Leur utilisation est présentée en Section 5.5.

Une métrique de variation radiale permettant d'identifier des jointures structurelles basée sur les différences de distance visibilité à l'intérieur d'une région entre deux points voisins. La dérivée de cette mesure correspond à la vitesse de transition entre deux cavités ou parties structurelles saillantes. Même si cette mesure est invariante par échelle lorsqu'il n'y a pas de discontinuité de visibilité, sa nature numérique implique au minimum un paramètre de précision. Nous discutons des solutions possibles afin d'épargner l'utilisateur du contrôle manuel des paramètres de notre métrique dans la Section 5.4.2. L'utilisation de cette dernière pour l'identification de parties structurelles est présentée Section 5.4.

Un algorithme récursif pour la décomposition structurelle d'un dessin complexe en parties structurelles. Cet algorithme utilise les zones de jointures identifiées grâce à la contribution ci-dessus. La hiérarchie en profondeur est conservée entre chaque itération afin de proposer un format de sorties avec calques adapté à de l'animation 2D simplifiée. Je présente cet algorithme ainsi que ses avantages et ses limitations dans la Section 5.5.

Publications

Merci d'accéder à la liste des publications en Section 1.4.

Organisation de ce manuscrit

Je débute ce manuscrit par une présentation générale du contexte du problème étudié afin d'introduire certaines notions importantes. Cette présentation complétée d'un état de l'art du domaine forme le Chapitre 3. Je présente ensuite mes deux travaux publiés ainsi que leurs extensions et expérimentations non publiées dans les Chapitres 4 & 5. Je conclus ce manuscrit avec un petit résumé et une présentation de certaines expérimentations non publiées et travaux futurs au Chapitre 6 et plus succintement en version française au Chapitre 7.

CHAPTER 3

Background and Related Work

Sketch-based modeling approaches have been inspired by the common use of drawings to represent 3D shapes as well as our ability to perceive 3D shapes from drawings. It may not be obvious, but this last sentence already stated the main problem of sketches: one draws what one perceives as being what one wants to draw. It means that drawings may only correlate with a meaning that is embedded in our visual process. Thus not only the depth information is missing in drawings but also some important elements compared to a mathematical projection of a 3D shape (as shown in Figure 3.3 (right)). Perception involves subconscious processes and it makes many tasks difficult to translate into computer algorithms: perceiving composition of objects, extrapolating hidden parts, interpolating missing parts, perceiving the depiction of an object as similar to the object itself, resolving ambiguties, etc. Many sketch-based modeling approaches avoid most of these problems by being interactive and only allowing non ambiguous and constrained intermediate input. Since our goal is to develop automatic approaches, it is a part of our problem.

Therefore, after defining the terminology relative to drawings used in the rest of the manuscript, this chapter first presents some key principles of perception discovered in the domain of perceptual psychology. We then explain how these principles are used, first for shape interpretation, then for modeling augmented 2D content (such as layered 2D and *bas-reliefs*) from sketches, and finally for 3D modeling from sketches.

3.1 Terminology of Sketches

Let us consider "black on white" line drawings without hatching strokes nor shading, multiple terminologies are introduced :

Regions of a drawing are connected components of the white space of the drawing.

Internal silhouette: The internal silhouette of a 3D object is the locus of all 2D points that belong to its projection, similar to a shadow.

External silhouette / Silhouette contour: boundary points of a shape's internal silhouette.

Contours including silhouette contours, are connected subsets of the locus of depth discontinuities due to occlusions (i.e. where the normal to the shape is orthogonal to the viewpoint).

Crease lines/curves correspond to valleys and ridges of a shape. They can either be sharp (e.g. in polyhedral solids) or soft (e.g in smooth manifold solids).

Suggestive contours have been introduced by [DeCarlo et al., 2003]. They correspond to extensions of contours where a true contour would appear with a minimal change in viewpoint (Figure 3.2). For the sake of simplicity, this name is used in the rest of this manuscript for branch-less contours with at least an open suggestive end.

Hair represent 1D elements.

Texture contours are connected subsets of the locus of color/shading discontinuities of the underlying surface.

Cartoon-specific elements are over-simplified 1D representations of 2D objects, movements or intents.

T, Y, and X-junctions are points where more than two incident curves meet. Their respective names describe the local shape of the junction (see Figure 3.1)

Figure 3.1 – The common curve junctions found in line drawings.

Although the term "line" is used interchangeably with "curve" when talking about drawings with artists, the use of "line" is reserved for straight lines in this manuscript.

Inner closed contours and some configurations of connected contours are challenging to interpret. The ambiguity resides in the local indistinguishability between the two sides of a contour (see Figure 3.9). To solve this ambiguity, the notion of "oriented contour" is introduced. There are two possible and opposite orientations per contour, and an oriented contour is arbitrarily defined in this manuscript to be the boundary of its right side as seen from any of its oriented local tangents. In other words, for one moving in the orientation of the contour, the right-hand side is closer to the viewpoint than the left-hand side. The

Figure 3.2 – (left) Contours of David's head. (right) Suggestive contours, as defined and rendered in [DeCarlo et al., 2003]. The latter helps to convey shape and is similar to what one could find in a drawing, except that these contours are mathematically generated.

Figure 3.3 – (left) One tends to perceive an object not only when looking at the object itself but also when looking at a depiction or image of this object. This image is an adaptation of *La Trahison des images*, René Magritte. (right) One also tends to perceive more detailed objects than the drawing actually depicts.

term "half-curve" is used in this manuscript to more generally represent an oriented curve that is a boundary of the 2D region of the plane that is on its right-hand side. A pair of opposite half-curves are created for each curve in the initial planar graphs of drawings, when the contour orientations are not yet known.

3.2 Perceptual psychology

As one can experience when looking at Figure 3.3 (left), a sketch can automatically pop the idea of the depicted 3D shape in anyone's mind. This is what makes the human perceptual system a long standing wonder for many communities: neuroscience, psychology, cognitive science and computer vision. But as other brain systems, it unveils some of its secrets when it is fooled or not working as predicted. Thanks to optical illusions and awareness of some feelings toward particular patterns, researchers were able to theorize some key psychological components.

Gestalt psychologists in the first half of the 20th century recognized key psychological components of the visual system, often called the Gestalt principles of perception [Koffka, 1935]. These were mainly developed around a consensual principle called the law of *Prägnanz* which states that one perceives and interprets complex images as the simplest form(s) possible. They found that the information that is treated consciously tends to be ordered, regular and simpler than it actually is. Subconscious processes automatically try to build hierarchies, relations and groups to form simple ideas. That is what everybody is experiencing in the everyday life, otherwise consciousness would be overwhelmed by raw stimuli. Indeed the brain also tends to generalize facts or minimize problems because the complexity of our environment is difficult to apprehend in its globality. In an attempt to formalize their findings about perception, Gestalt psychologists designed over the years a list of laws. The most commonly known laws are often referred to as the Gestalt Principles in the literature (illustrated in Figure 3.4) are:

Law of proximity: One perceives objects that are close to each other as a whole, a visual group. Hence one also perceives these at approximately the same depth in the monoscopic scenario.

Law of similarity: Perception tends to group similar objects together.

Law of closure: Visual gaps tend to be filled if it produces a coherent whole. It can close a silhouette contour or connect partially occluded parts.

Law of symmetry: Symmetry is not perceived as accidental. It can be experienced while programming: pairs of brackets are perceptually forming a block. Two overlaping squares with an axis of symmetry are not perceived as 3 contiguous polygons.

Law of continuity: One does not perceive continuities as coincidences. Therefore, aligned hidden contours are interpolated to build background shapes and 2 crossing lines are not perceived as 4 lines meeting at a point.

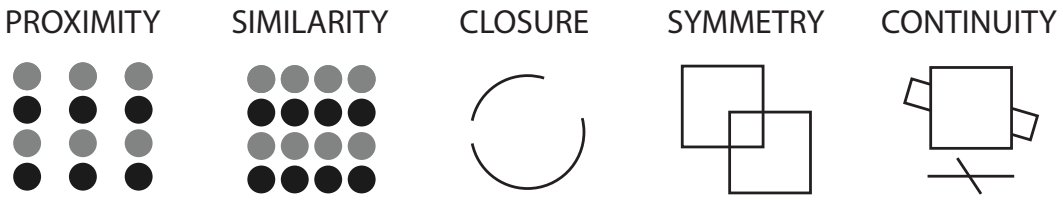


Figure 3.4 – The Gestalt Principles, core rules of the visual system (see Section 3.2).

A more complex and specific principle is also involved in the perceptual process but in a lower measure: the past experiences principle. It states that one tends to perceive elements according to past experiences, whether it is colors or shapes. Particularly, prior knowledge of an object helps to perceive it properly but also seems to affect the way one would depict it.

Quantifying or weighting the relations between the different laws of perception in a single solution is really difficult as these weights are influenced by this *a priori* knowledge. A recent deep learning approach to group discrete graphical patterns [Lun et al., 2017] suggests that machine learning might be a good solution to structure scenes as a human being would.

The most commonly known bias related to prior knowledge is the human ability to differentiate human faces. It is known among artists that drawing a portrait from a photograph is easier upside-down. This practice is studied in [Edwards, 2001] and shows that our recognition process overrides our appreciation of pure geometry. It is to be understood here that our perception influences our depiction. It correlates with studies [Willats, 1997] which claim that people actually depict their *idea* of a shape instead of reconstructing it from a photographic memory. Simple form of these ideas can be seen in drawings of young children where characters are stick figures with smiley heads.

If we consider the drawing process, it means that for each stroke, the perception of the current state will affect the rest of the depiction. It is an important factor in the design of a sketch-based interface for 3D modeling in the case of iterative techniques. The stroke effects in the software must be relevant of the user intents, more than the stroke itself.

3.2.1 Hoffman's rules

Another set of visual processing rules similar to the Gestalt Principles and also oriented toward shape understanding has been proposed by [Hoffman, 1998]. He claims that one constructs what one sees, and that the perception of a shape from contours is ruled by local cues, from which one infers local features and structure, and finally infer a coherent global shape.

Human perception mechanisms also heavily rely on statistics so that uncommon and aberrant shapes are subconsciously discarded from possible interpretations. For instance, in our daily visual experience, end points of curves that are coincident in one view have close to a hundred percent chance to remain this way in nearby views. Our brain uses this as an assumption when we are given a single viewpoint and no depth information such as in a drawing or a far enough scene. Thus we instantaneously perceive curves meeting in 2D as if they were meeting in 3D whether or not it's actually the case. This is called the rule of **generic views**. It is also obvious that one considers smoothly continuous contours as continuous in depth, and do not consider discrete accidental views. When it is actually accidental, perception is tricked, and the trick is called an optical illusion as in (see Figure 3.5). Similarly, perceptual processes tend to assume and conserve parallelism, colinearity, proximity, straightness, and smoothness when inferring 3D from 2D.

Hoffmann defines other rules such as "a line is a rim" but are admitted to be statistically weaker assumptions. It implies that some local cues can be misleading and that designing a resilient inference algorithm is not trivial.

Figure 3.5 – Anamorphic painting with space discontinuities. The text looks like it is overlayed to the photography while it is actually painted in the scene. It is an accidental view since only this view point hides the spatial discontinuities and create the illusion of a planar text. By Edmundo Saez [Creative Commons BY-SA 4.0 (http://creativecommons.org/licenses/by-sa/4.0)].

Similarly to Hoffmann's rules, the various approaches found in Computer Vision literature give a lot of importance to contours. Essential information can be derived from connections as well as maxima of curvature along contours as discussed in the following section.

3.3 Shape interpretation

Interpretation of drawings in Computer Vision has usually been studied with two different angles: contours and regions. Oversketched contours and other real world drawing specifities have not been considered by most shape interpretation algorithms in the literature, since translating raw strokes into perceived contours is not trivial and small gaps can easily lead to misidentification of perceived regions. It can be noted that recent advances have been made in sketch embelishment techniques, providing ways to simplify sketchy contours. With real world usability in mind, this section now presents such techniques that could complement and prefix shape interpretation algorithms.

3.3.1 Sketch simplification

Digital sketches and drawings are usually represented either in rasterized form or in vector graphics form. Since vector graphics allow for easier computation of the topology as well as provide better defined junctions of curves, it reprensents a more convenient input for shape interpretation algorithms. However, most of the drawings openly shared on the Web are in rasterized form and are not trivially convertible into vector graphics. To support our choice of input as well as to introduce applications of perceptual principles,

two recent techniques (among others) which tackle the problem of vectorization of raster drawings are now presented.

Topology-Driven Vectorization of Clean Line Drawings [Noris et al., 2013]

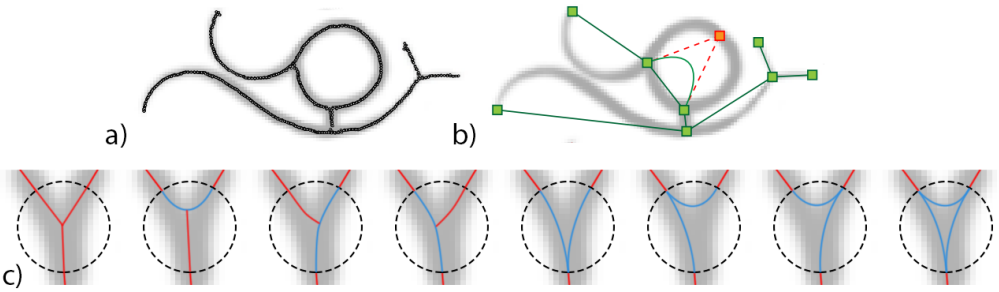


Figure 3.6 – Vectorization steps of [Noris et al., 2013]. (a) Pixel clustering. (b) Topology extraction. (c) Possible solutions at stroke junctions. The continuity principle favoring continuities between incident strokes, the solution is chosen based on incident curvatures estimates as well as curvatures of the interpolating lines.

Noris et al. proposed to consider raster drawings with relatively crips and distinct strokes that have been traced in single hand movements. The problem here is that strokes are radially softened and thus creates ambiguities at stroke junctions. The first part of their algorithm is a clustering of pixels. Then they extract the topology these clusters form (see Figure 3.6 (a, b)). Finally they use this topology and the principle of continuity to disambiguate junctions using the curvatures of incident strokes (see Figure 3.6 (c)).

Closure-aware sketch simplification [Liu et al., 2015]

In the case of sketchy drawings the main problems are oversketching and accidental gaps. The main Gestalt Principles involved can be identified, namely: proximity, continuity, and closure (see Figure 3.7).

Liu et al. proposed to combine these principles in a closure-aware sketch simplification algorithm. Although applying the proximity and continuity rules appear relatively easy, tackling the principle of closure at the same time is non trivial. Since closure depends on regions and that their contours are yet to be built from raw strokes, we understand that we face a problem of reciprocal influence. It is solved using cyclic refinement by alternating between a process that merges non user-intended regions into perceptual regions and a process that groups strokes using the closure principle with the updated regions. this method gives interesting results (see Figure 3.8) that are close to having the clean and informative contour topology most shape interpretation techniques require, including ours.

Since the methods we just presented can be used to convert clean line raster drawings into vector graphics drawings, we will only consider the latter as inputs in the remain-

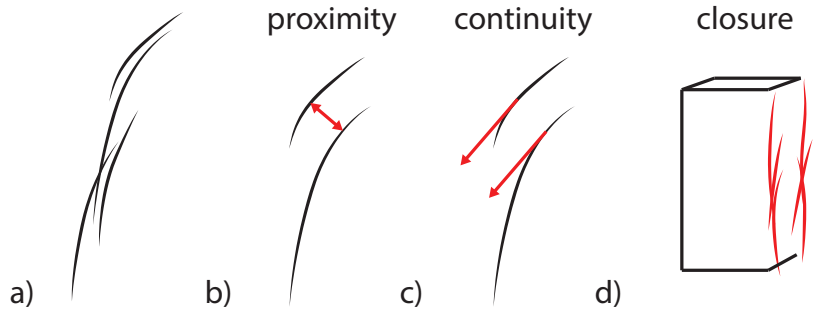


Figure 3.7 – Gestalt Principles involved in the perception of oversketching and accidental gaps. (a) An oversketched stroke. (b, c) Strokes are perceputally grouped depending on proximity and continuity. (d) The principle of closure here constrains the grouping of contours in order to close visible faces.

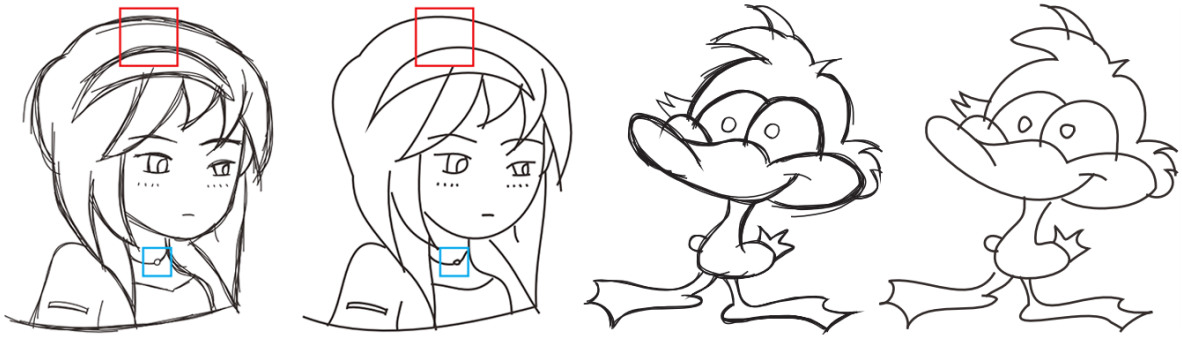


Figure 3.8 – Two results of sketch simplification by [Liu et al., 2015].

der of this manuscript. Thus the notion of "stroke" mostly disappear and is replaced by "curve".

It is now interesting to focus on the main problem that is identifying the meaning of curves. Let's first look at an obvious ambiguity regarding open end curves located inside closed regions, such as the mouth of the girl in Figure 3.8. These curves can either be suggestive contours, hairs, texture curves, or cartoon-specific strokes. As of now, dealing with all this diversity in a single drawing without asking the user to annotate such curves is still an open problem. To alleviate this kind of ambiguity, the first studies in computer vision considered restricted domains, as in the line drawing interpretation algorithms that is now presented.

3.3.2 Interpretation from contours

Curves in a drawing can have different meanings such as those presented in Section 3.1. The ambiguity can be mitigated by restricting the input domain to drawings with contours only. However, as one can see in Figure 3.9, the occluding side of a contour in the

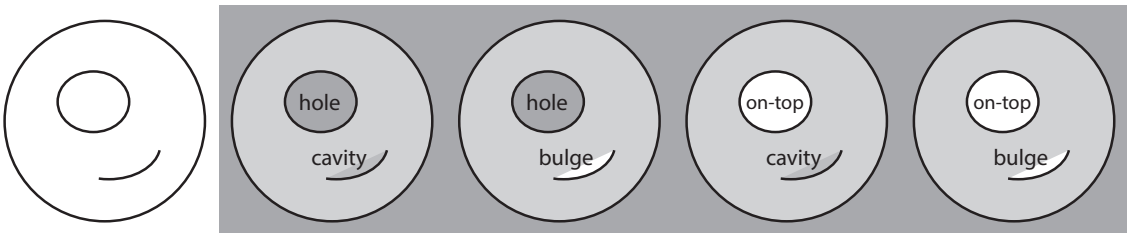


Figure 3.9 – Ambiguity of orientation (occluding side) of inner contours (supposed to be contours). From left to right, the input contour drawing followed by the 4 possible interpretations.

case of drawings of smooth solids is not trivial to deduce and can even be unresolvable without prior knowledge and semantics about the depicted shape. These are not sufficiently restricted domains to begin the study of partially occluded parts completion and depth inference. But there exist classes of input that alleviate this ambiguity by providing geometric cues and constraints, namely polyhedral manifold-solids, and forward facing faces of solids (termed *anterior surfaces*) that are the subject of pioneer works in shape interpretation.

Polyhedral manifold-solids

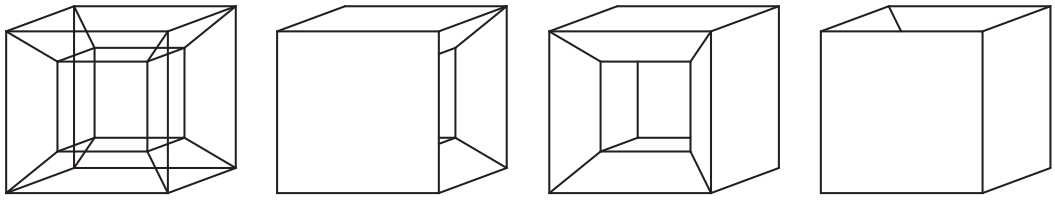


Figure 3.10 – Ambiguity of the represented solid in a wireframe drawing of a polyhedral solid (left) with the three possible interpretations (right).

Polyhedral manifold solids show the advantage of giving useful geometric cues and constraints to start with. Three different ways of drawing such solids are commonly found in Computer Aided Design: wireframe, solid, and solid with dashed lines for occluded parts of edges. However, the latter becomes equivalent to wireframe beyond the foreground faces' edges as occlusions can happen between dashed-line edges themselves. In the wireframe case the missing occluder/occluded disambiguation allow for different sets of faces to define coherent solids (as shown in Figure 3.10). This ambiguity cannot be lifted without arbitrary constraints or domain-specific knowledge as in approaches presented later in Section 3.4.1.

Thus, the pioneer work on shape interpretation focuses on representations of this kind of shapes, with a solid aspect, in which the elements can be well-defined. Those drawings

are exclusively composed of lines with no open end. These lines may either represent contour edges or crease edges but are all boundaries of at least one and at least partially visible face. The other assumption is that the considered polyhedra are manifold-solids, thus all vertices have at least three incident edges. Visible junctions of lines in 2D are however not always 3D vertices and can represent the intersection of an edge being occluded by a contour that is closer to the viewer.

From these considerations, Guzmán first proposed a solution to identify visual groups of polyhedral regions forming independant polyhedral blocks or solids (see [Guzmán, 1968]). Although it works well with scenes composed of isolated convex trihedral polyhedra, it required ad-hoc solutions to extend it to scenes with occlusions and concavities. Later, [Huffman, 1971] and [Clowes, 1971], proposed to interpret the lines in drawings of polyhedra in regard to the matter they locally bound, providing more meaningful information. To this end, they defined four different labels for lines: "+" for a convex edge, "-" for a concave edge, and "<" or ">" for an occluding edge and its orientation. The orientation of a contour edge is defined as the direction in which one would walk along the edge in order to have the occluding matter on its right side.

Obviously the labeling of a line is unique and cannot be different at its two ends. Hence when identifying the possible combinations of labels for incident edges to each vertex the problem becomes constrained by global consistency. Althought the approach in [Guzmán, 1968] looks for consistent membership of edge sides to independant solids, these labeling techniques do not provide any way to check for the planarity of faces. As a result, nonsense labelings that are not consistent with polyhedra can be found among the resulting solutions (as shown in Figure 3.11). To tackle this problem, the additional consistency on faces was later studied by [Macworth, 1973] but it is specific to polyhedra and does not scale to the family of shape studied in this thesis.

The labeling scheme has also been extended by [Kanade, 1980] to handle non-manifold polyhedra, referred to as "origamis". This extension confirmed the difficulty of defining what are "legal line drawings" when considering more and more possible labelings at junctions. With respect to the number of junctions N, as the quantity of line junction interpretations M rises, the labeling complexity explodes exponentially $O(M^N)$ if locally impossible cases are not directly excluded from the possibilities. This not only induces longer computational times but also leaves more possible solutions due to insufficient or imprecise constraints. Accidental views remain the worst case scenario for the labeling scheme approaches since additional ambiguities can only amplify the under-constrained nature of the problem.

Although these approaches provide a good insight of the line interpretation problem,

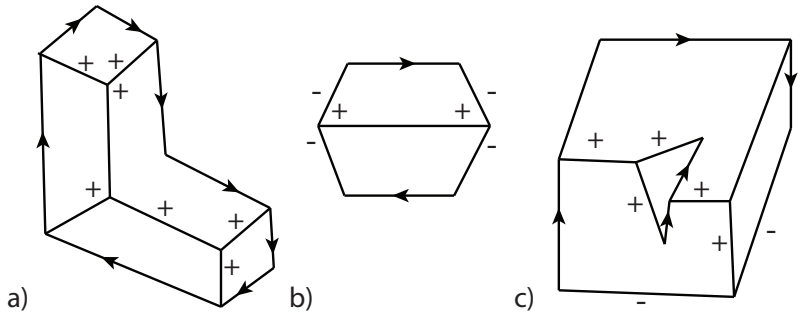


Figure 3.11 – Nonsense labelings identified as coherent by Huffman's scheme [Huffman, 1971]. Labels for lines are: "+" for a convex edge, "-" for a concave edge, and "<" or ">" for occluding edges and their orientation.

drawings of smooth manifold solids unfortunately shows even less constraints with non-straight curves, no faces but a single non-planar surface, and ambiguous open end curves. We now propose to present some previous works on a category of drawings that is closer to drawings of general smooth manifold-solids, that is "anterior surfaces".

Anterior surfaces

In the case of smooth manifold-solids, the occluded parts are often as important as the visible ones. Thus instead of only interpreting the visible curves and regions, [Nitzberg and Mumford, 1990] as well as [Williams and Hanson, 1996] studied a superset of visible surfaces referred to as "anterior surfaces". These surfaces are defined as the locus of surface points in a scene where the normal has a positive component in the viewing direction. When a scene is only composed of such surfaces, it is called an "anterior scene". We note that considering the anterior surfaces of a smooth manifold-solid removes the notion of back surfaces and thickness which are only useful for 3D depth reconstruction but are not relevant to define a simple depth ordering.

An anterior surface is a reasonable consistent whole to aim for when trying to infer high-reliefs from the visible surfaces of smooth manifold-solid. It can be approached as a completion problem where Gestalt principles constraint the solution.

This is what Williams proposed in the context of illusory surfaces (see [Williams and Hanson, 1996]). Note that only the underlying computational theory and surfaces with non-illusory contours are discussed in this chapter. His idea is to tackle the problem with a labeling scheme as for the previous works on polyhedra, and explore the space of possible completions of visible contours looking for valid anterior surfaces with respect to this labeling scheme. This is called a figural completion.

The ambiguities to alleviate for inputs with only visible contours of anterior surfaces

are of three forms:

shape ambiguity: Any kind of shape feature can be occluded with no cue left. Thus the inference of occluded contours has not a single solution. Usually, least energy curves are used.

unit ambiguity: Contour fragments can be interpolated in different ways forming different units (see Figure 3.12 (b)).

depth ambiguity: Occluded overlaps of units have not definite preferred order (see Figure 3.12 (c)). Also, some contours may have different possible orientations, either forming a unit in the form of an independent surface boundary or a hole in an existing unit (see Figure 3.9 (c)).

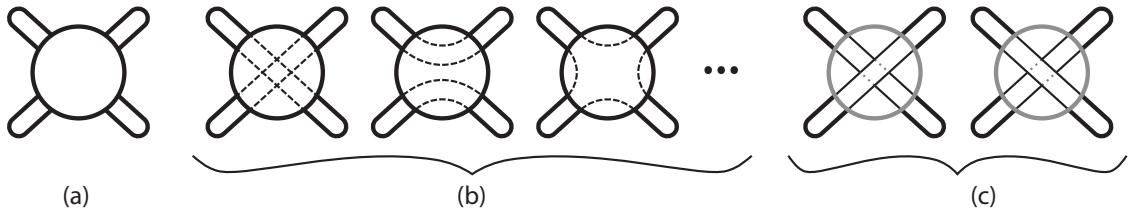


Figure 3.12 – **(a)** Visible contours input. **(b)** Some plausible solution of unit organizations. It is a *unit ambiguity*. **(c)** The two possible solutions of depth organizations for the occluded overlap of units in the first figure in (b). It is a *depth ambiguity*. The gray circle is the boundary of the disk in (a) that is the occluder of the overlap in the middle.

Williams et al. made the assumption that *shape ambiguity* is independent from the two others. That is why his figural completion is divided into two stages respective to the latter groups of ambiguities. Furthermore Williams et al. chose to restrict the problem to generic views where contours are not tangentially overlapping but can only form non ambiguous crossings in order to avoid unnecessary complexity. In other words, his visible contours inputs may only contain T-junctions but no open end curves as the Kanizsa's *X's with Bars* [Kanizsa, 1979] (see Figure 3.13 (a)).

The first stage of Williams' algorithm is computing for each T-junctions a small set plausible occluded contours connecting this junction to others. In [Witkin and Tenenbaum, 1983], the authors supports the use of least energy interpolating curves using the incident tangents at the junctions. The main argument is that the existence and validity of such curve is a reliable indicator of a non-accidental relationship between junctions. We emphasize that it over-constrains the solution compared to diffusion-based techniques as in [Geiger et al., 1998] and [Sýkora et al., 2014].

The second stage deals concurrently with both *unit* and *depth* organization. Since an anterior surface has a single smoothly continuous closed contour in 2D similar to a

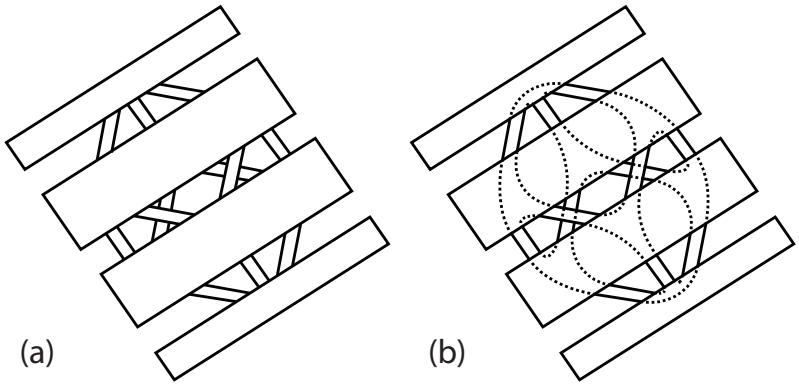


Figure 3.13 – (a) Kanizsa's *X's with Bars* test figure [Kanizsa, 1979]. (b) Figural completion achieved by Williams' technique (see [Williams and Hanson, 1996]). Contour orientations are not shown for readability.

knot, Williams proposed to study the projection of anterior surfaces as a knot-diagram (as shown in Figure 3.14 (a)). This diagram is composed of labelled curves that are the pieces of the surface boundary knot cut at all the intersections in 2D. Labels are composed of an orientation similarly to oriented contours and a depth index equal to the number of surfaces lying between the viewer and the contour. Since all crossings are T-junctions the scheme is composed of two rules to respect two possible configurations of incident labels (as shown in Figure 3.14 (b, c)). Valid labelings are searched using an integer linear program. Since only depth inequalities can be determined with this algorithm, the resulting representation is considered being in between 2D and 3D, and is usually called a 2.5D representation.

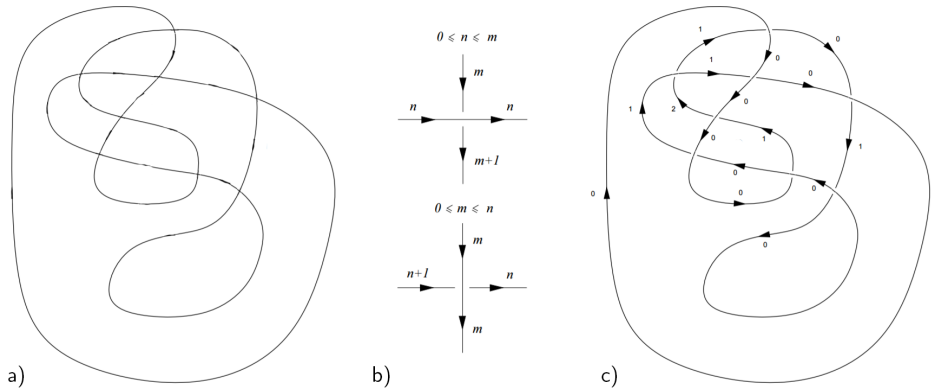


Figure 3.14 – **(a)** A knot-diagram of an anterior surface. **(b)** The set of rules, called a labeling scheme, used in [Williams and Hanson, 1996] and [Williams, 1997]. **(c)** The scheme in (b) applied to (a).

Combining those two stages with preference criteria, Williams achieved results for

inputs without cusps but with geometrical cues such as the one shown in Figure 3.13 (b). Later, he proves that the labeling extensionto smooth-manifold solids in [Huffman, 1971] is sufficient to distinguish impossible objects (see [Williams, 1997]).

For smooth manifold solids the generation of plausible completed shape candidates remains a very difficult problem. Thus, to this day it exists no general algorithm of figural completion for these objects. However we can interpret drawn shapes in an other way that is looking at whole silhouettes.

3.3.3 Interpretation from silhouette

The internal silhouette of a drawn shape brings the advantage that it is a single element with a unique contour. Even if this contour may be composed of fragments of shape contours boundaries belonging to distinct components of the shape, we emphasize that most objects can be recognized from only their internal silhouette. That is because we recognize a shape by both its main component and its collection of features. We note that most shape features are oriented outward in living beings as well as in designed objects. This may be explained by their common necessity of being useable or able to do and use things so they have manipulators, positioners and handles. A tree grows in branches to position its leaves in the sun. On the other hand, features which are oriented inward are mostly holes or cavities as in containers, cheese, filters, tubes. It is also important to point out that these shape features are rarely constrained to lye on a plane, thus we can often see at least one feature from every point of view.

Since the visible contours of some features may not be part of an external silhouette, A silhouette alone does not provide enough information to properly infer a higher dimension representation in many cases since contours of some shape features may not be part of its boundary. Therefore studied silhouettes can be divided into two categories with different usage: silhouettes of complex shapes with visible features, and silhouettes of primitive shapes including potato-like shapes also called blobs. The latter category can be used in iterative 3D sketch-based modeling as presented in Section 3.4.1.

On the other hand the first category is often approached as a shape recognition problem first. This recognition can build upon a structural analysis of the silhouette either using intermediate representations. Only then a matching model from a database can be used and may be ajusted to fit the silhouette to infer a satisfying 3D shape (see [Kraevoy et al., 2009]).

Medial-Axis Transform

The Medial Axis Transform, also called MAT or just Medial-Axis, was introduced by [Blum, 1967] as a loss-less representation of silhouettes. It is often displayed as an inner skeleton of the shape but each point is actually the center of a disk inscribed in

the silhouette. Thus the radii are kept along with skeleton points. A way of computing the skeleton is to see it as a voronoi diagram of a polyline silhouette (see Voronoi1908). Different improvements and extensions have been developped upon this work such as approximations of the equivalent structure for 3D shapes. We now emphasize that noise on the boundary of the silhouette leads to an over-branched skeleton. This problem can be alleviated by The Scale-Axis Transform introduce by [Giesen et al., 2009] to obtain clean skeletons as in Figure 3.15.

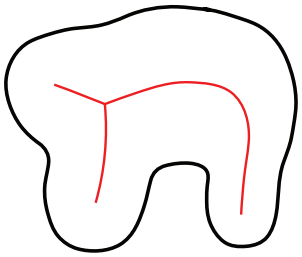


Figure 3.15 – A Medial-Axis skeleton (in red) of an external silhouette (in black) pruned using the Scale-Axis Transform [Giesen et al., 2009].

A similar structure but with a discontinuous skeleton called Smoothed Local Symmetries or SLS was introduced by [Brady and Asada, 1984] (see Figure 3.16 (b)).

Segmentation and decomposition

The main goal of segmentation of a silhouette is to divide it into a descriptor structure of smaller meaningful sub-parts such as salient features. Pioneering work on 2D shape structuring via segmentation use the local axis and radii of Medial-Axis points as well as their derivatives to compute a structure of salient branches (see [Blum and Nagel, 1978]). Inspired from this work, a recent technique by [Leonard et al., 2016] proposes a shape structure with multi-level decomposition and identification of similarities.

Other early works on 2D shape segmentation and understanding use the maxima of negative curvature on the contour to identify part boundaries as in [Richards et al., 1987, Latecki and Lakämper, 1999] (see Figure 3.16 (a)). However the meaningfulness of curvature maxima alone is difficult to quantify since it is scale dependent.

While focusing on the inside of the silhouette seemed obvious [Liu et al., 2014] used the complement of the silhouette or dual-space to measure salience and infer a segmentation for complex shapes.

A segmentation technique introduced by [Zeng et al., 2008] combines both skeleton and contour information in order to tackle the problem of noise in contours that could locally be interpreted as small features. Analyzing both contours and regions concurrently is in my opinion an evident requirement when trying to tackle complex drawings. The importance of contour features is relative to the local size of the region and the unit and

depth organizations are partially given by contours. While my published research handles noise in a different manner, I discuss some experimental work that uses this complementarity to deal with noise in Section 5.7.

Many other silhouette decomposition techniques exist, thus I refer the reader to [Yang et al., 2008] and [Shamir, 2008] for surveys.

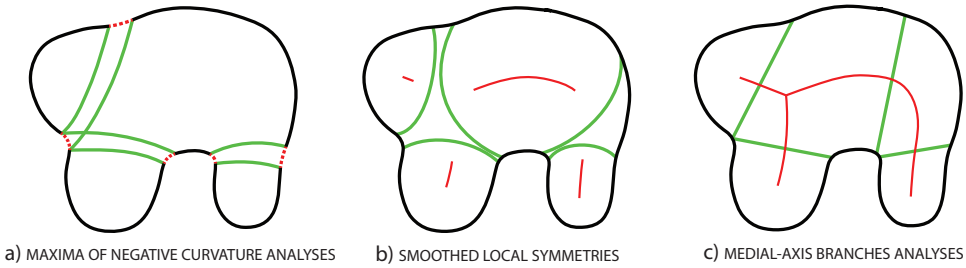


Figure 3.16 – Illustration of structures generated by different silhouette decomposition techniques (see Section 3.3.3).

3.4 Sketch-based modeling

Line drawings may not contain enough information to fully describe what an artist has in mind. However it is arguably the best compromise between intuitiveness and precision when using a 2D medium for the creation of 3D digital content. That is why a number of techniques called sketch-based modeling techniques aim to use sketches in the 3D modeling process. This section follows the chronological order of my interests in this field during my thesis. First I present the different approaches for sketch-based 3D modeling. Then a range of techniques for 2.5D modeling that try to handle more complex inputs while only achieving results that are not complete 3D models.

3.4.1 Sketch-based 3D modeling techniques

Most 2D drawings and sketches only represent the visible contours and the silhouette of an object. Converting them automatically into 3D models therefore requires resolving indeterminacies and inferring for a large amount of missing data. This strategy is used by a category of methods called direct single-view methods that usually require different sets of hypotheses or/and *a priori* knowledge of the shape being modeled. Other sketch-based modeling techniques avoid these hypotheses using either user interaction, multiple views, or a database of predefined shapes (see [Cook and Agah, 2009] and [Olsen et al., 2009] for detailed surveys).

Iterative methods

A way of approaching the sketch-based modeling field is to not consider sketching as an input to a 3D inference problem but as a solution to fast 3D modeling of specific shapes. Iterative methods enable the user to build general models part by part, by iteratively adding new shape components from different viewpoints as in the interactive Teddy system (see [Igarashi et al., 1999]). These methods make the hypothesis that the final shape is a combination of parts that all have planar silhouettes from a given viewpoint, and can therefore be inflated from closed planar contours. This is supported by our prior discussion on how artists pre-sketch organic shapes with potato-shaped loops. Creatures belong to this category, but theory aside these methods still require practice and time in order to achieve a convincing result. However it brings the advantage of superior edition capabilities as each element can be represented as an independant unit that can be later re-edited or removed. The simplicity of these units also allows for the use of intuitive manipulators or annotations as used by [Gingold et al., 2009] in a two-panel application. These panels are: a single viewpoint 2D sketching and editing space and a viewport to the iteratively built 3D model (see Figure 3.17). Other iterative methods specific to non design shapes have been developed such as techniques based on implicit surfaces as in [Schmidt et al., 2007] and [Bernhardt et al., 2008]. The ability of a collection of blobs or masses to represent an organic shape shows through the above techniques.

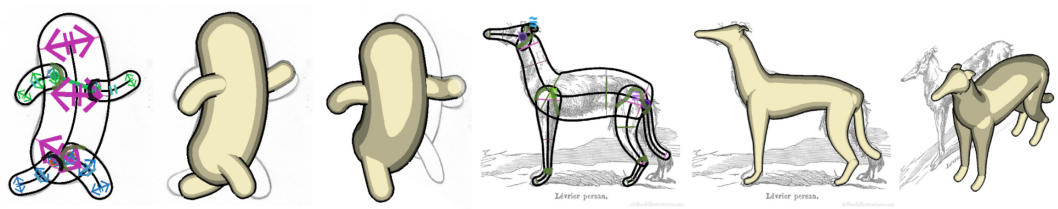


Figure 3.17 – Results from *Structured Annotations for 2D-to-3D Modeling* by [Gingold et al., 2009].

Some techniques give the user different predefined patterns that he can draw and that have different meanings or usage. It often comes with a specific creation process that the user must follow. For instance in [Wither et al., 2009] the user is asked to draw trees in a specific way. Then he must draw quite-straight lines for trunks, and cloud-shaped closed curves for the foliage silhouettes. The user then refines the style of one branch until the leaf level, and this is finally propagated to other branches. Pre-defined styles can also be applied to match specific silhouettes. The method proposed by [Ijiri et al., 2006] for flowering plants uses a similar strategy but allows the user to also model the lower level components that are stems, leaves, petals, flowers. In [Turquin et al., 2007] the user draws a dress in two steps: general aspect first then folds as ridges or valleys parameterized with

orthogonally intersecting u-shaped strokes.

Direct methods

Some object-specific methods also exist without user interaction, such as for the modeling of blood vessels in [Pihuit et al., 2010]. In this work Pihuit proposes to use a specific color to draw contour lines and hatching. Everything that is not white is considered the silhouette of a network of tubular elements. From this assumption a skeleton of this network is computed. Then hatching and Y-junctions are identified in the contour network and used as cues as to how the different segments should be curved in depth. It aims to respect both the smoothness of curvatures and the rule of proximity that says that we tend to perceive the shape as taking no more space in depth than it takes in the two visible dimensions.

We saw that object-specific methods create drawing conventions for the artist to use. But some techniques uses the already existing conventions of certain types of sketches such as design drawings. In this category [Xu et al., 2014] proposes to identify the different geometric regularities given by descriptive curves found in design drawings to infer a plausible 3D curve network from which a surface is interpolated. These cross section curve regularities are planarity, orthogonality, parallelism and symmetry and are cued by crossings, tangents and local symmetries.

Symmetry can also be a global assumption of the method as in [Cordier et al., 2011] who take orthogonal projections of perfectly symmetric shapes as input for 3D reconstruction.

Line drawings of smooth-manifold solids do contain only a few cues as presented in Section 3.3.2. As 2D figural completion is already non trivial in this case, 3D inference is even more difficult. Nonetheless some techniques tackle this problem by making a reasonable number of assumptions. For instance, our algorithm presented in Section 4 uses *a priori* knowledge on the anatomy of animals to drive the unit organization. Amongst previous works, a technique proposed by [Karpenko and Hughes, 2006] uses minimal user input on visible contours drawings of smooth shapes. Their algorithm takes as input a visible contour drawing of a smooth-manifold-solid with cusps and annotated contours' orientation. The authors propose a figural completion method designed for cusp-ended contours incident to T-junctions (see Figure 3.18 (a,b)). Inspiring from [Williams and Hanson, 1996] they compute for each pair of endpoints a likelihood for this pair to be the endpoints of an hidden contour. The figural completion consists in a greedy search amongst the set of distinct pairs for one that completes the input so that a valid Huffman's labeling exists. In practice only the 10 most likely next pairs are used for the search in the space of completion sets. This likelihood is a sum of two energies: one is inversely

proportional to the curvature of the interpolating curve and the other is an heuristic on the angle between the tangents at endpoints. Once a likely set is filled, it is tested for veridicity with the labeling step.

Finally the resulting valid labeling scheme is used to compute a paneling of regions with both front facing regions and back facing regions. A set of coherence rules on stacks of regions, edges and points are used to stitch the triangulations of the panels at paired edges as well as positioning them at different depths. A relaxation is then used to smoothen the resulting manifold mesh to obtain the final result (see Figure 3.18 (c,d)).

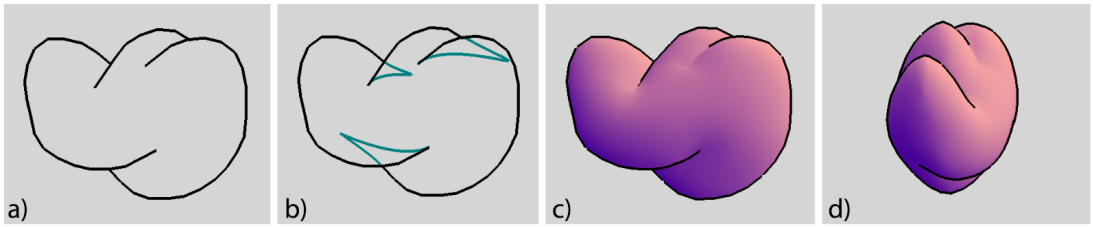


Figure 3.18 – (a) Input drawing. (b,c,d) Figural completion handling cusp points and final results using *SmoothSketch* [Karpenko and Hughes, 2006].

My research presented in Section 4 and Section 5 are both inspired from this work but are built upon the idea of structural part completion instead of figural completion of actual contours.

Other approaches

Approaches that use shape recognition generally try to match the user sketch with silhouettes, parts of silhouettes, or specific descriptors of predefined 3D models. These matched 3D models can then be deformed or posed to better match the depicted shape. This method was successfully applied to organic shapes such as humans or animals [Kraevoy et al., 2009], as well as to technical models [Fonseca et al., 2009]. However, they require a template example of the given general class of animal which imposes a restriction on the family of sketches that can be handled. We did not consider this approach in our first work in order to allow the user to imagine any animal shape without further restrictions on the number of limbs, horns, etc.

Multi-view methods such as in [Rivers et al., 2010a] generate 3D shapes from two or three sketches drawn from different and preferentially orthogonal viewpoints. They require the ability to draw consistent views of the shape to be reconstructed. Therefore they are more adapted to the depiction of man-made objects than to the one of organic shapes. Alternatively, a number of man-made objects can be easily built from a network of 3D curves, which can themselves be reconstructed from perspective sketches [Schmidt et al., 2009, Xu et al., 2014]. However, these methods require more user input than a

single contour sketch, and are generally difficult to apply to the modeling of the shapes this thesis focuses on, namely smooth free form organic shapes.

3.5 Modeling depth-augmented 2D sketches

Vector graphics drawings are often considered 2D, but their layering capability may already be used as a discrete third dimension if surfaces are not interwoven. Since using the term "3D" would be confusing, representations with only partial or approximated depth information are often referred to as being in 2.5D, something lying in between 2D and 3D. As seen in Section 3.3.2, some sketch interpretation techniques can infer these 2.5D representations from 2D inputs and other techniques use interactive solutions. This section presents a few relevant works on both sketch-based 2.5D modeling techniques and approaches to the manipulation of 2.5D models.

Local layering [McCann and Pollard, 2009]

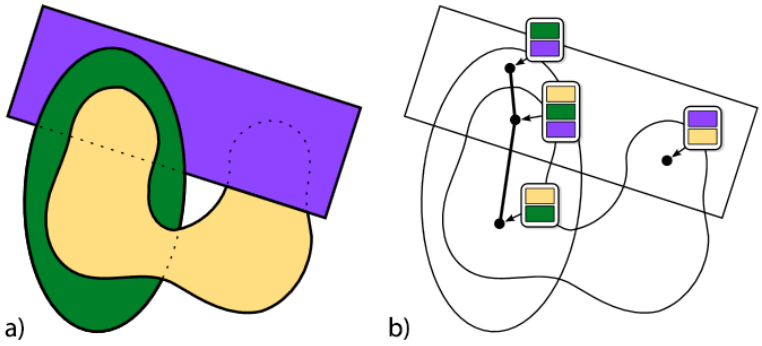


Figure 3.19 – (a) Example of three 2D objects interwoven and locally layered. Solid and dotted lines respectively represent visible and hidden contours. (b) Graph of local layerings. The depth orders of objects on both sides of each boundary are kept valid during editing.

This work proposes a method to manipulate interwoven surfaces while conserving the veridicity of local depth ordering. To this end a graph similar to the paneling used by [Huffman, 1971] and [Williams, 1997] is created. Each edge of this graph corresponds to a contour segment between two regions of the drawing. Each vertex corresponds to a region and a depth-ordered stack of surface chunks lying in this region (see Figure 3.19). Every manipulation of a surface unit that would create an edge where the two corresponding stacks do not present compatible depth orderings are prevented. A dedicated reordering tool satisfying the same constraint is also proposed.

2.5D Cartoon Hair Modeling and Manipulation [Yeh et al., 2015]

Figure 3.20 – Pipeline for the automatic animation of cartoon hair proposed by [Yeh et al., 2015].

In this research, the authors propose to extract 2D hair strands from drawn cartoon hair and animate them using a skeleton based representation. It builds upon the same intuitions of sub-part extraction via a combination of segmentation and contour completion. However the completion here is easier since hidden contours are quite straight and the fact that each strand should end with a sharp point is a strong constraint.

Ink-and-Ray [Sýkora et al., 2014]

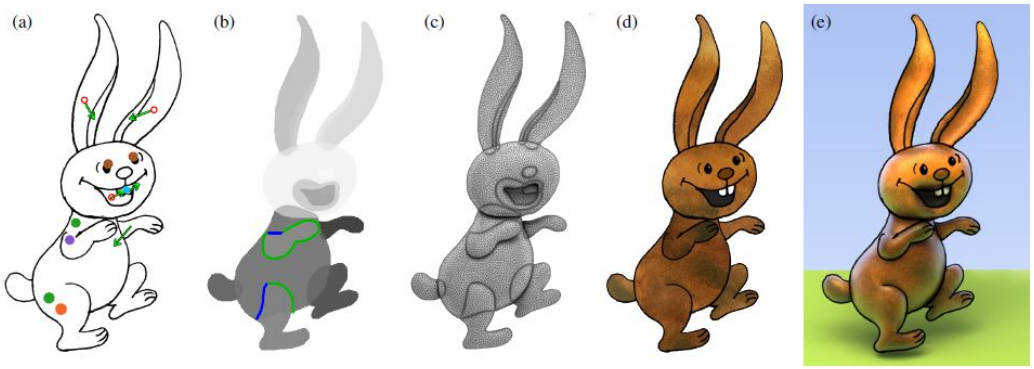


Figure 3.21 – Pipeline for the user-assisted inference of high-reliefs for cartoon drawings by [Sýkora et al., 2014].

D. Sykora et al. proposed an algorithm for the inference of an high-relief model from an annotated drawing building upon their own previous work in [Sýkora et al., 2010]. The segmentation and the depth orderings are however automatically guessed and the user annotations are only used to fix possible misidentifications. In order to complete the hidden contours they use a default rule that is: each region is an independant shape unit unless the user specify it otherwise. The completion method used is based on diffusion curves, where probabilities of pixels to belong to a unit are diffused from the open ends of unit contours. Then each unit is completed with every pixel that has a propability of belonging to it greater than 0.5. This solution is particularly interesting considering the

reasonably small amount of fixes that the user has to do to produce convincing results (see Figure 3.21).

A more recent technique for the interactive modeling of high-relief meshes as been proposed by [Yeh et al., 2017]. But in my opinion, in requires an excessive amount of user annotations.

Other works

I would like to cite [Rivers et al., 2010b] for their 2.5D cartoon drawings that provide a convincing cartoonish rotation of characters around the vertical axis. Such rotation is defined expressively more than geometrically, yet they succeeded to tackle this problem using an interpolation between elements drawn in orthogonal views and a way to specify at which angle individual elements should disappear completly.

Another approach to the animation of drawings is proposed by [Dalstein et al., 2014] with a 2.5D representation called *Vector Graphics Complexes* adapted to the edition of topology changes in time using key frames.

3.6 Conclusion

In this chapter I presented a quick overview of human perception as well as some works related to sketch interpretation and sketch-based modeling in vision and graphics. I gave the key elements to understand why having a machine automatically interpret and build structured information from a single sketch remains a challenge in the case of organic shapes. The next chapter explore our first contributions to this field, starting with a sub-category of sketches where prior knowledge eases interpretation and modeling.

CHAPTER 4

Modeling 3D animals from a side-view sketch

4.1 Introduction

With the spread of 3D virtual environments and of 3D printing technologies, many practitioners would like to author their own 3D shapes. Among them, animal models – including imaginary and fantastic ones – are an important category. Being able to easily create and then animate animals would be an important step for generating more lively virtual worlds. Animals are also among the models that the general public, especially children, would typically like to sculpt and print.

There is currently no fast and easy method for creating 3D models of animals. Unfortunately, getting data from 3D scans is much more difficult for animals than for humans, beginning with the obvious challenge of requiring an animal to stand still. In addition, such reconstructions are also limited to existing animals. Standard 3D modeling software, such as Autodesk's Maya or Blender, as well as digital sculpting software such as Pixologic-Zbrush, can be used for creating animals, but their complexity limits their use to experienced or passionate users. The use of 3D sculpting is possible, but is still difficult: many people are not adept at sculpting animals using real clay and in this case, they are not likely to perform much better in a virtual setting, even with an investment of time in the mastery of digital sculpting interfaces. Sketch-based modeling systems, which only require users to sketch contours in 2D, are probably the most intuitive and accessible class of methods. However, despite these advantages, they either require users to iteratively draw complex shapes part by part, using different viewpoints, or, alternatively, they require an existing data-base of 3D models.

Our work belongs to the category of sketch-based modeling methods and is the first to explore the creation of a 3D animal model from a single, side-view sketch. We are

motivated by the belief that many users are capable of drawing a single sketch that depicts the contour lines and the internal silhouettes of an animal, such as shown in the top-left of Figure 4.1. If needed, users can use a background drawing or a photo as a guide. The process of inferring 3D geometry from the 2D sketch necessitates the use of relevant assumptions in order to be tractable, and in our context of modeling animal forms, we shall assume smoothness of the resulting shape as well as the presence of structural symmetries. Several further moderate constraints include: (a) restricting ourself to non self-overlapping limbs in the sketch; (b) requiring the user to draw both contours for pairs of symmetric limbs; and (c) ignoring the reconstruction of repetitive details scattered on the surface, such as scales. With these assumptions and constraints in place, the method we develop is capable of automatically converting an input vectorized sketch into a 3D model. The total processing time is less than one second, effectively enabling one to create new animal models in only the time required to sketch them.

Note that in this work, we only tackle the creation of the volumetric shape parts of an animal and that we do not consider the surface components that should be used for ears or scales. The ears we reconstruct are also therefore interpreted as volumes. We are not able to reconstruct large flat parts such as wings. In addition, we reconstruct limbs in a symmetric fashion, even when they were drawn in arbitrary postures in the input sketch. The construction of symmetric 3D models is usually desirable, as it ensures that left and right limbs have identical dimensions. To achieve a desired non-symmetric posture, the 3D model can be deformed, either by using an animation skeleton and the associated skinning weights, or by directly articulating the implicit surface's skeleton.

Our processing pipeline for creating 3D animals from a sketch is summarized in Section 4.2. It consists of three main steps, which also correspond to our three technical contributions:

1. the identification of the animal's foreground structural parts in the sketch, with completion of the parts that are not explicitly bounded, such as the top of the legs (see the teaser figure)

2. the generation of a hierarchical graph of depths for the structural parts, using the complete results and based on an algorithm for detecting the portions of the sketch that correspond to symmetrical parts of the animal;

3. 3D reconstruction based on a specific choice of implicit surface, scale invariant integral surfaces, which enables us to accurately reconstruct shape parts from their medial axis in the 2D sketch, and to seamlessly blend them into a single animal model.

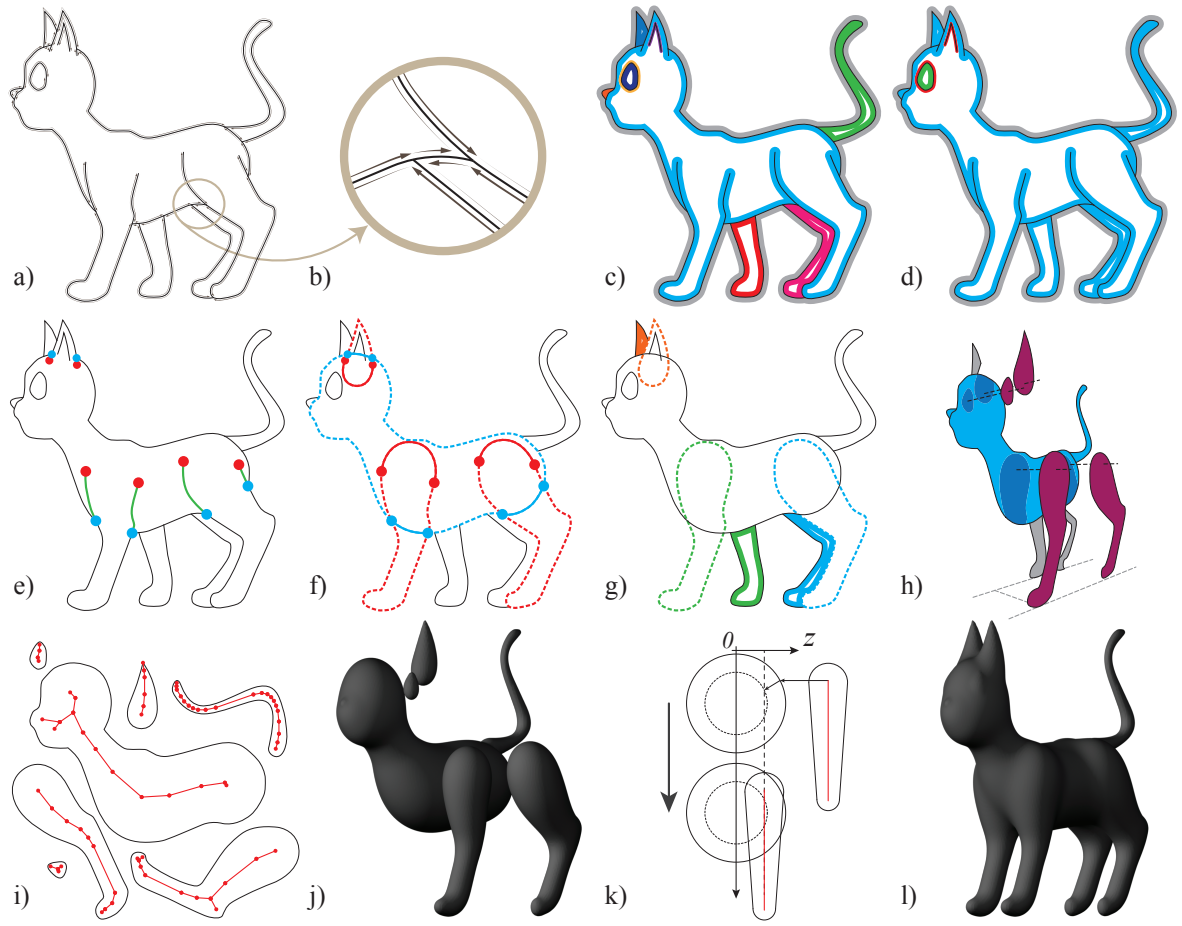
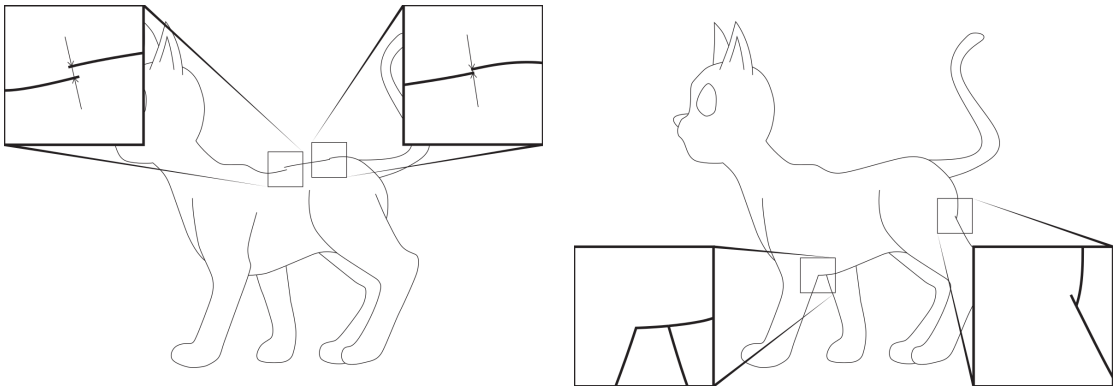


Figure 4.1 – Overview of algorithmic stages: a) Half-edge graph; b) Detailed view of the resulting bounded curves; c) Cycles extracted from the graph, each one depicted in a different color; d) Classification of cycles: the different colors now represent the cycle type; e) Detection of suggestive contours; f) Completion of contours around shape parts; g) Detection of the main body and of pairs of symmetrical parts (displayed using the same color); h) Extracting a depth ordering; i) Skeletons extracted from the medial axes of shape parts; j) Reconstructed shape parts; k) Front view: inferring depth information for shape parts; l) Final result after implicit blending.

4.2 Overview

Our method builds a 3D model from a single 2D sketch. This involves the following steps, which are also summarized in Figure 4.1.

We start from a sagittal-view vectorized sketch of an animal in the (x, y) plane. The hand-drawn input sketch is transformed into a set of parametric curves using a vectorization algorithm such as the one proposed by [Noris et al., 2013]. These algorithms are relatively robust to noise and provide the smooth curves we seek. What the user must really pay attention to is the correctness of the input hand-drawn sketch topology,



(a) (b)

Figure 4.2 – Two cases of failure: (a) Discontinuous sketch: a large discontinuity, as the one on the left, prevents the detection of the main body part in the drawing. In that case our algorithm fails from its first steps. A small gap in the sketch, as on the left, is filtered and considered as a connection during our first step. The reconstruction will perform adequately. (b) Incomplete sketch: on the left, a structural part is missing in the sketch, here, the left inner-body silhouette of the front foreground leg. Our algorithm will not be able to detect the presence of a limb and the reconstruction of the legs will fail. On the right, both inner-body silhouettes of the rear foreground leg are drawn and the rear legs will therefore be adequately reconstructed.

i.e., the provision of continuous closed contours that do not have spurious crossings with other contours. Figure 4.2 illustrates two failure cases, the first, in 4.2(a), is due to an incorrectly closed contour and the second, in 4.2(b), is due to an incomplete input.

The goal of the first step, detailed in Section 4.3, is to identify the strokes that correspond to the same structural part, such as an eye, leg, or body, and to infer any missing curves, i.e., perform contour completion, if the resulting contour is open. This is done in the following way: We build a counter-clock-wise oriented half-edge graph whose edges are in general bounded parametric curves (see Figure 4.1(a,b)). In this graph, we iteratively follow successive half-edges, and store cycles in a list (Figure 4.1(c)). Each cycle is classified as being either an *outer-sketch*, *island*, *border* or *other* as explained in Section 4.3.1 and illustrated in Figure 4.1(d).

We then define as *inner-edges* those edges where both half-edges belong to the same cycle (the green edges in Figure 4.1(e)). These lists of inner-edges represent *suggestive curves* where two shape parts merge (such as a leg merging with the body). This implies the presence of hidden silhouettes or cusps [Karpenko and Hughes, 2006]. Each suggestive curve has an open endpoint and a closed endpoing, which are depicted in red and blue, respectively, in Figure 4.1(e). The next step is to pair these suggestive curves and connect their extremities in order to infer new cycles representing the different structural parts of the animal (Figure 4.1(f)). This is the topic of Section 4.3.2.

The subsequent step, presented in Section 4.4, is the computation of a depth hierarchy for structural parts, based on the structural symmetry hypothesis. We first identify the main body part located in the saggital plane, i.e., the torso, and detect structural symmetries around it such as pairs of ears or pairs of front legs, cf. Section 4.4.1 and Figure 4.1(g)). We then use the pieces of information we already gathered, e.g., island cycles, structural parts, chest, and symmetries, to define a depth hierarchy between structural parts, as detailed in Section 4.4.2 and Figure 4.1(h).

This leads us to the last step, detailed in Section 4.5 which consists of the 3D reconstruction of the animal model. For each pair of limbs, only the one in the foreground of the drawing is considered during the reconstruction steps. The 3D reconstruction of the background limbs are added at the end of the process by symmetry. We choose implicit surfaces to represent shape parts, in order to benefit from their blending capabilities. We first compute the medial axis [Fortune, 1986] of each structural part (Figure 4.1(i)). Medial axes are then filtered and specifically simplified in order to be be used as skeletons for 3D implicit surface modeling (Section 4.5.1 and Figure 4.1(j). Among the variety of existing implicit models, we use scale invariant integral surfaces (SCALIS) [Zanni et al., 2013] to accurately reconstruct shape parts (Section 4.5.2). Figure 4.1(j)) shows the 3D reconstruction of each structural shape part in isolation. Structural parts without symmetries are placed in the sagittal plane of the model. The depth of the foreground structural parts is computed using the thickness of their 3D reconstruction and the thickness of the part to which it is connected in the sagittal plane (Section 4.5.3 and Figure 4.1(k)). Symmetric background parts are then added to provide the final 3D model (Figure 4.1(l)). The final shape is obtained using a simple sum to blend the fields from the different implicit primitives.

The next sections give a detailed presentation of our solutions at each stage of this process.

4.3 Structural parts identification and completion

4.3.1 Representing the sketch as a set of cycles

The first step is to decompose the curves of the sketch into a set of cycles. This decomposition is a preliminary step towards identifying the different parts of the animal drawn in the sketch. In particular, this step is required for the hidden contour computation and the contour closure, as described in Section 4.4.

The input data is a hand-drawn sketch composed of a set of non-crossing curves that are connected to each other (see Figure 4.1(a)). Each of these curves is represented by a pair of directed half-edges [1] of opposite direction (see Figure 4.1(b)). The curves of the

1. Note that each graph edge geometrically correspond to a full curve of the sketch.

sketch are then represented using a graph whose edges are the half-edges of the curves and whose nodes are the points at which the curves are connected to each other. Next, we decompose this graph into a set of cycles.

Formally, a cycle is defined as a closed sequence of connected half-edges; each half-edge shares its endpoints with at least two other half-edges. We require the half-edges that compose a cycle not to cross each other; we also require the cycles to be topologically equivalent to a disk. Therefore, a cycle divides the sketch into an "interior" region and an "exterior" region. In order to define the interior of a cycle, cycles and their associated half-edges are given a direction which is either clockwise or counter-clockwise; the interior of the region bounded by a cycle lies on the left side of the directed half-edges that compose this cycle.

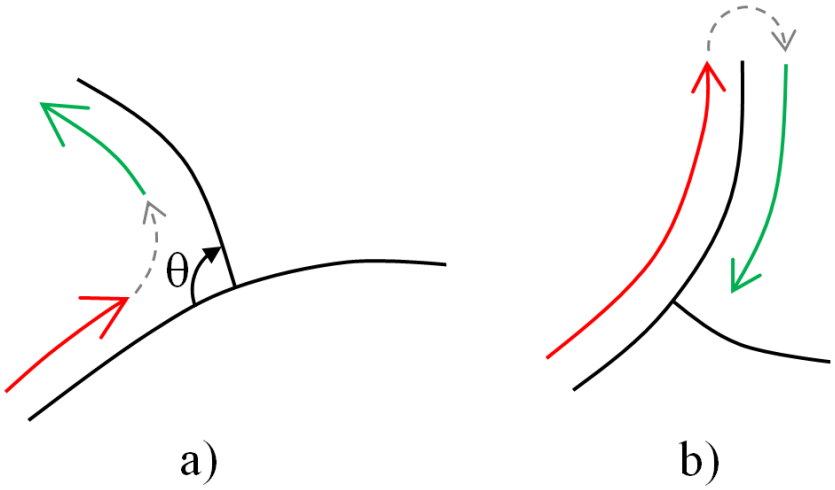


Figure 4.3 – Selecting the next half-edge. In (a), the selection of the next half-edge (shown in green) for T-junctions is chosen such that the angle between the next half-edge and the preceding one (shown in red) is the smallest possible. In (b), the selection of the next half-edge for cusps.

The first cycle that we identify is the one which is composed of the half-edges located along the outer boundary of the sketch. This cycle is oriented clockwise and is unbounded; it corresponds to the outer region of the sketch. We name this the *outer* sketch cycle. Next, we identify the *border* cycles which are cycles that have one or several half-edges belonging to the outer boundary of the sketch. All of these cycles are oriented in a counter-clockwise fashion. The process to construct a border cycle is as follows: we first select a half edge which is located along the outer boundary of the sketch and which has not yet been associated to a cycle. We next find its endpoint by following its direction and select the next half-edge. This next half-edge is chosen such that its direction is the same as the previous half-edge and the clock-wise oriented angle between the two half-edges is the smallest possible, as illustrated in Figure 4.3. The process of finding the next half-edge is iterated until we reach the first half-edge.

Finally, we identify the "island cycles"; island cycles are single half-edges that correspond to a closed curve with no self-intersection and which are not connected to any other curves of the sketch. These island cycles are also oriented counter-clockwise, meaning that they do not represent a hole but rather a surface feature of the animal (such as the eye of cat model shown in Figure 4.1). After processing all the cycles, there may remain some cycles that could not be classified as any of the three types (outer sketch cycles, border cycles and island cycles); we mark these cycles as *others*.

4.3.2 Contour completion

The goal is now to convert the set of cycles into closed contours associated to each structural part of the shape.

Once the graph cycles are classified as described above, we use the hypothesis that structural parts have at least one edge in contact with the outer part of the sketch, and consider only the border cycles. In these cycles, we extract the sets of neighbor embedded edges, i.e. edges whose half-edges both belong to the same cycle, and then annotate these as being suggestive curves. Each set is an open path in the graph with one extremity connected to the rest of the graph, i.e., a closed extremity, and one extremity without any connection, i.e., an open extremity (see Figure 4.1(e)). As we focus on organic models, the suggestive curves come from the drawing of the silhouette of structural parts that stop where the limb smoothly merges with the part of the shape over which they are drawn (for instance the cat legs over the body in Figure 4.1). We require the user to provide two suggestive curves clearly delimiting each merging limb.

Parts need to be individually closed to be identified and reconstructed. To do so, we compute a set of cubic Bézier curves linking open extremities and closed extremities pairwise with a C^1 continuous connection with the edge curves while minimizing the normalized sum of the curvature variations ("Gestalt principles"):

$$E = \frac{1}{l^2} \sum_i |k_{i+1} - k_i| \quad \text{where} \quad k_i = \frac{\|\dot{c}_i \times \ddot{c}_i\|}{\|\dot{c}_i{}^3\|}, \tag{4.1}$$

Here, l is the curve length, k_i is the curvature of the curve at sample point i, $\dot{c}_i$ is the curve first derivative, and $\ddot{c}_i$ is the second derivative at parameter value u_i.

The sum of curvature variations, $l^2 E$, is used as the energy associated with the curve. For curves connecting closed extremities, these energies are denoted E_c and for those connecting open extremities they are denoted E_o. We then compute two energies E_i $(i = 0, 1)$ for each pair of suggestive curves, one for each way the closed extremities can be connected (dashed curves in Figure 4.4(a,b)). These energies are computed as follows:

$$E_i = \left(p_0^i . p_1^i\right) \left(E_o + E_c^i\right), \tag{4.2}$$

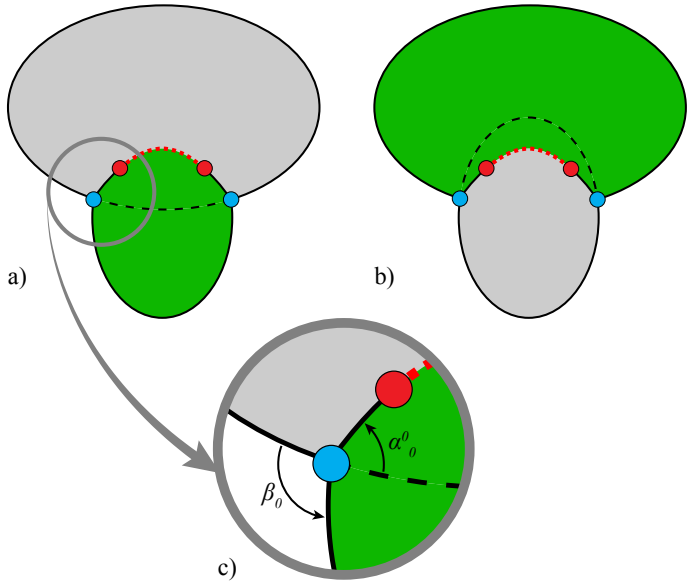


Figure 4.4 – (a) and (b): Illustration of the two possible completions of a pair of closed extremities. (c) Close-up on a closed extremity illustrating the two angles α_0^0 and β_0 used in the computation of the probability p_0^0.

where E_c^i is the energy associated to the curve connecting closed extremities ($i = 0$ for the closure illustrated in Figure 4.4(a) and $i = 1$ for the one illustrated in Figure 4.4(b)), p_0^i and p_1^i are the probabilities that the curve is correctly connected at each extremity respectively. These probabilities are computed as the ratio $p_j^i = min\left(\alpha_j^i/\beta_j, 1\right)$ $(j = 0, 1)$ in which angles α_0^0 and β_0 are illustrated in Figure 4.4(c). All other angles are defined in a symmetric fashion. The best-matching pair of suggestive contours is then chosen as the pair having the minimal sum of energies.

Once the suggestive contours are paired and closed, we define their set of half-edges and derive the new cycles. In these cycles, no edge is embedded in the defined part (i.e. no edge has both half-edges in the same cycle). Together with the remaining border cycles and the island cycles, this gives us the set of the animal structural parts drawn in the sketch.

The minimization of curvature variations is known to provide plausible results when completing organic curves. The selected completion curves linking open and closed extremities does not, in general, intersect other contours. However, if this should occur, it will not be expected as it does not faithfully correspond to the intent of the input sketch, although it does not compromise the rest of the reconstruction. If necessary, the user can correct the completion by editing the extremities.

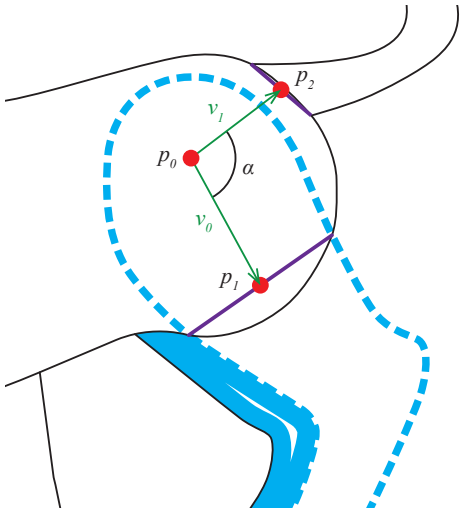


Figure 4.5 – Identification of the cycles belonging to the shape part at the background that is structurally symmetrical to a foreground limb.

4.4 Depth hierarchy of structural parts

4.4.1 Body and symmetrical parts

In order to assign appropriate depth values, we create a graph that represents the desired depth relations. This begins with the choice of a reference part as the root node of the graph, for which it is natural to use the body or torso. For some animals, this main body part may be smaller than the limbs. As such, we did not find a robust method to automatically label this in the input sketch. Instead, we request the user to select it or to draw the sketch so as to locate it in the center of the drawing canvas. Once labeled, this part defines the node of reference for the graph. The method described below automatically locates the remaining structural parts with respect to the body, in terms of their relative depth. We begin by detecting the sketch strokes that correspond to pairs of symmetric parts, as explained next.

We already know the parts, F_i, that should be located in front of the body and thus that may have a symmetrical part in the background. These are defined by the open contours with suggestive curves we just completed at the previous step. The goal is now to identify the strokes from the sketch corresponding to B_i, the structurally symmetric part with respect to F_i, but located in the background. Indeed, B_i should not be processed any further, since we are going to use symmetric geometry to create the background limbs. In practice, B_i may be composed of several distinct cycles, since it is partly occluded. We use a propagation method for fully selecting it, as we now describe.

We initialize B_i with all cycles not belonging to the shape part under F_i, but that share an edge with F_i. We then compute the medial axis of F_i as described in Section 4.5. Let us denote as p_0 the medial axis extremity corresponding to the suggestive silhouette

we closed (see Figure 4.5). We then denote as p_1 the middle of the closed extremities of the suggestive curves and then define $v_0 = p_1 - p_0$. Starting from both closed extremities, we march along both sides (one clock-wise and the other counter-clock wise) of the cycle of the structural part which is located under F_i in the sketch, as long as the opposite half-edge belongs to the outside sketch. If we meet an opposite half-edge belonging to a border cycle, we get both extremities of the shared edge and we compute their middle point p_2. We define $v_1 = p_2 - p_0$ and compute the angle $\alpha = |\widehat{v_0, v_1}|$. If $\alpha < \pi/4$ we consider that this cycle potentially belongs to B_i, since the angle formed by two symmetrical parts is unlikely to be larger than this value in usual standing poses. Note that this criterion could be relaxed to handle a larger set of poses.

We stop when the half-edge we reach is an occluded edge, i.e., when we meet the next foreground part. If selected on multiple occasions, a potentially symmetric cycle is assigned to the B_i for which the angle α with F_i is minimal. The results of this selection process are depicted in Figure 1(g), where pairs of symmetric parts are depicted using the same colors.

4.4.2 Graph construction

In order to support the 3D reconstruction of the model, we build a graph whose nodes are the structural parts (and their associated cycle) and whose edges represent the relations *over*, *under*, *adjacent* or *over-in* with respect to the main body. Foreground parts are defined as lying *over* the part they have been separated from. Symmetric counterparts are located *under* the part that their foreground is *over*. All remaining border parts that have not been designated as part of a symmetric pair are defined as being *adjacent* to parts with which they share an edge. Islands are assigned to the graph by first using a simple 2D ray-cast in the sketch plane in order to identify the part within which the island lies. The island cycles are over and inside a structural part, and this part is also then defined as being *under* the island. Each edge is further tagged with additional information, such as the presence of suggestive curves, which help determine if the region being over a particular part is an island or a limb.

4.5 3D Reconstruction

4.5.1 Medial-axis computation & simplification

Animals have the convenient property that their shape is naturally approximated as a set of generalized cylinders. A generalized cylinder is a surface obtained by moving a circle of varying radius along a 3D arbitrary curve. An advantage of generalized cylinders is that their skeleton and associated radii are easily determined from their projected image. Given a 2D simple closed curve C, the skeleton and the associated radii of the generalized

cylinder whose orthogonal projection matches C are obtained by computing the medial axis and the associated radius function of the curve C.

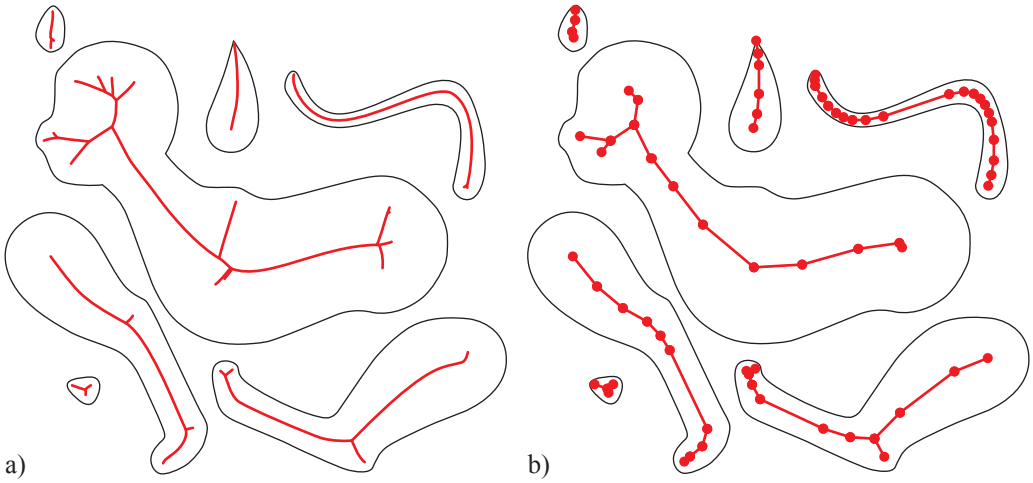


Figure 4.6 – Medial axis computation and simplication. In (a), the medial axes of the sketch components computed using Fortune's sweep-line algorithm [Fortune, 1986]. In (b), the skeletons after removing the branches using the Scale-Axis Transform [Giesen et al., 2009] and after simplifying the medial axes using the Douglas-Peucker algorithm [Douglas and Peucker, 2011].

We start the 3D reconstruction step by reconstructing each structural part independently. The parts which were detected as being symmetric to foreground parts are not considered. The contour curve of each part is first sampled so as to obtain a polygonal curve. We then compute the medial axis transform of each of the polygonal curves using the Fortune's sweep-line algorithm [Fortune, 1986]. The medial axes usually contain many branches because of the noise in the polygonal curves. We use the Scale-Axis Transform method by [Giesen et al., 2009] to remove these branches. In our experiments, the scale factor of the Scale Axis Transform has been set to 1.2.

The last step is to simplify the medial axis; the purpose of this simplification is to lower the computation time for generating the surface of the reconstructed animal. We have implemented a modified version of the Douglas-Peucker algorithm [Douglas and Peucker, 2011] which is a method to reduce the number of points of polygonal curves. A cost function is defined for each point, which is the maximum distance between the original curve and the curve after removing the point. Points are removed iteratively in increasing order of their cost. The algorithm stops when the cost of the point to remove is above a user threshold. Figure 4.6 (b) illustrate the result of the cat skeletons simplification.

The Douglas-Peucker algorithm has been modified to take into account the radii of the medial axis. This modification consists of a new cost function. Let C be a polygonal curve and p_A, p_B, p_C be three neighboring points of the medial axis of C. If the point p_B

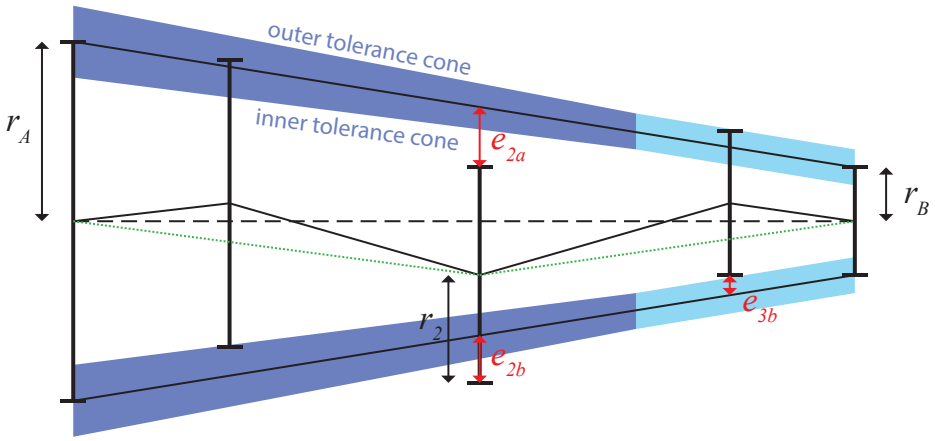


Figure 4.7 – Medial axis simplification. In (a), the polygonal curve (depicted in black) and its medial axis (depicted in red) before the simplification. In (b), the medial axis has been simplified by removing the point p_B. The result is new polygonal curve with fewer segments. The cost value of removing the point p_B is d; it is the maximum distance between the original curve (depicted in light grey) and the curve after the simplification.

is removed, the two segments $\overline{p_A p_B}$ and $\overline{p_B p_C}$ are replaced with the segment $\overline{p_A p_C}$ (see Figure 4.7). Similarly, the curve corresponding to the medial axis is simplified; two of its segments are removed. The new cost function is defined as the maximum distance d between the original curve and the curve after the simplification of the medial axis.

4.5.2 Surface reconstruction of the shape components

Once the skeletons of all the shape components have been computed, the next step is to reconstruct the 3D surface of these shape components. We have implemented the SCALe-invariant Integral Surfaces (SCALIS) method proposed by [Zanni et al., 2013]. This method generates an implicit surface using a skeleton and its radius function. In Zanni's method, the scalar field is computed with the Homothetic Polynomial kernel of degree 8. In our method, we have implemented the kernel of degree 6 instead of degree 8. Although the quality of the reconstructed surface is slighly lower, the computation time is much smaller, and allows for surface reconstruction at near-interactive rates. The result is a set of implicit surfaces, one for each shape component (see Figure 4.1(k)).

4.5.3 Placement & embedding of the shape components

The last step of the surface reconstruction is to assemble the implicit surfaces to generate the 3D shape of the animal. At this stage, the skeletons of the implicit surfaces are all located in the same plane which is the sketching plane. The goal is to place these implicit surfaces so that their depth order complies with that of the sketch; the shape

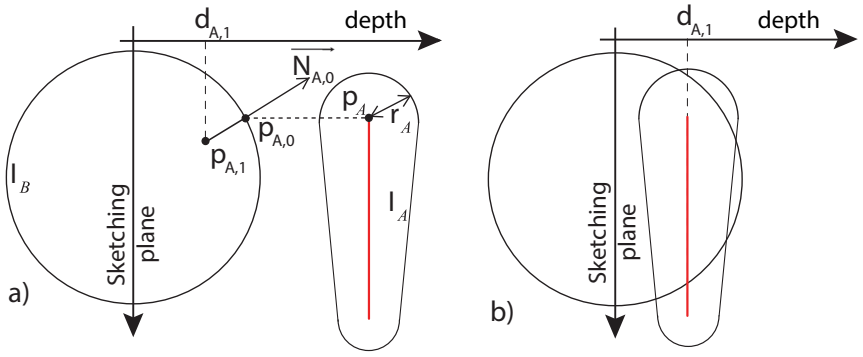


Figure 4.8 – Computation of the depth coordinates of the implicit surface I_A. In (a), the depth coordinate $d_{A,1}$ of the skeleton extremity p_A is computed. In (b), the skeleton (depicted in red) of I_A is translated to the depth coordinate $d_{A,1}$.

components that are drawn in the sketch foreground should be placed in front of the sketching plane and those drawn in the sketch background should be placed behind the sketching plane. We make use of the depth graph that has been previously computed (see Section 5.2).

We start by placing the implicit surface corresponding to the main body component in the sketching plane. Other implicit surfaces are sequentially placed as follows. Using the depth graph, we find a shape component that has not yet been placed and which is adjacent to a shape component which has already been placed. Using the depth order between these two shape components, we compute the depth position of the one to be placed.

Let I_A and I_B be two implicit surfaces, with I_A being in front of I_B. The depth coordinates of I_B have been already computed; those of I_A have to be computed. Let p_A and r_A be the skeleton extremity of the implicit surface I_A and its radius respectively. We first compute $p_{A,0}$ the orthogonal projection of p_A onto I_B. Next, we compute $\overrightarrow{N_{A,0}}$, the surface normal of the implicit surface I_B at the point $p_{A,0}$. Finally, we compute $p_{A,1}$ such that $\|\overrightarrow{p_{A,1}p_{A,0}}\|$ is equal to r_A and such that the two vectors $\overrightarrow{N_{A,0}}$ and $\overrightarrow{p_{A,1}p_{A,0}}$ are parallel and have same direction (see Figure 4.8(a)). The depth coordinate $d_{A,1}$ of $p_{A,1}$ is then used as the depth coordinate of all the points of the skeleton of I_A (see Figure 4.8(b)).

Once the depth coordinates of all the implicit surfaces have been computed, the 3D shape of the animal is then generated by blending these implicit surfaces using the well-known Ricci blending operator.

4.6 Results and discussion

We implement our method as a Maya plugin, enabling us to use the existing sketching interface of Maya to design the vectorized sketches we need as input. Processing the 2D

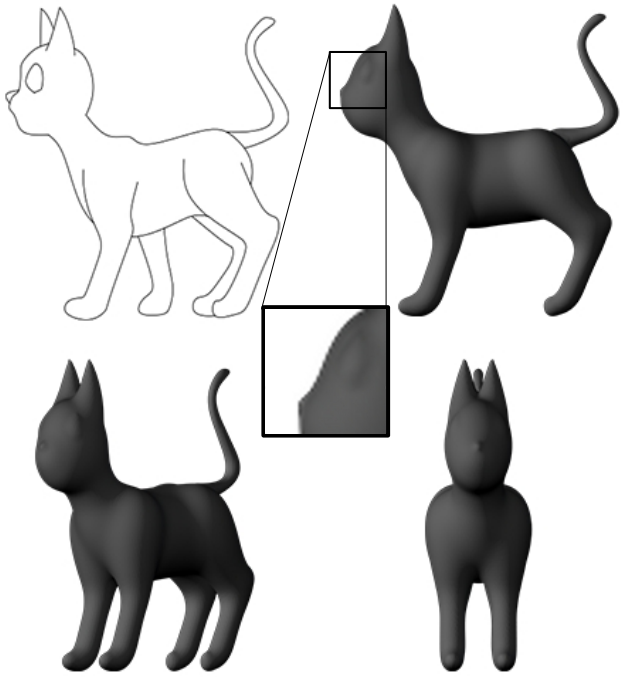


Figure 4.9 – Different views of the 3D reconstruction of the cat sketched in the top-left.

sketches into 3D models is then a fully automatic process.

The teaser image and Figures 4.9–4.12 show a set of results from our method. They validate the fact that our method is able to generate plausible rough shapes for various animals from a single sketch. Note that since we only used a structural symmetry hypothesis around the sagittal plane, our animal models may have an arbitrary number of protruding subparts such as limbs, ears (interpreted as volumes) or horns. Moreover, our choice of scale invariant implicit surfaces (SCALIS) as the 3D representation enables us to capture both the smoothness of organic shapes and the singularities that may come from the sketch, such as at the tip of the cat ears.

Pairs of limbs are reconstructed symmetrically, but the user can manually adjust the pose in 1 to 3 minutes by deforming the geometric skeleton (see Figure 4.1(i) the geometric skeleton of the cat model). Background limbs can be positioned in sketch pose as in Figure 4.12(c), or the model can be globally deformed as in Figure 4.12(mantis-d). In Figure 4.12(elephant), the ears are not reconstructed as they do not represent any of the different cycles considered in the sketch for our reconstruction (as explained in Section 4.3.1).

The results have been produced on an Intel Core i7 3770K computer, and the code is compiled with ICC 15.0 in release mode. For all our examples, the full sketch-processing and 3D modeling process run in less than one second. This makes the method applicable within an interactive sketching software tool. More precisely, the computational time for the giraffe is 0.35 seconds, for the gazelle is 0.30 seconds and for the cat 0.37 seconds.

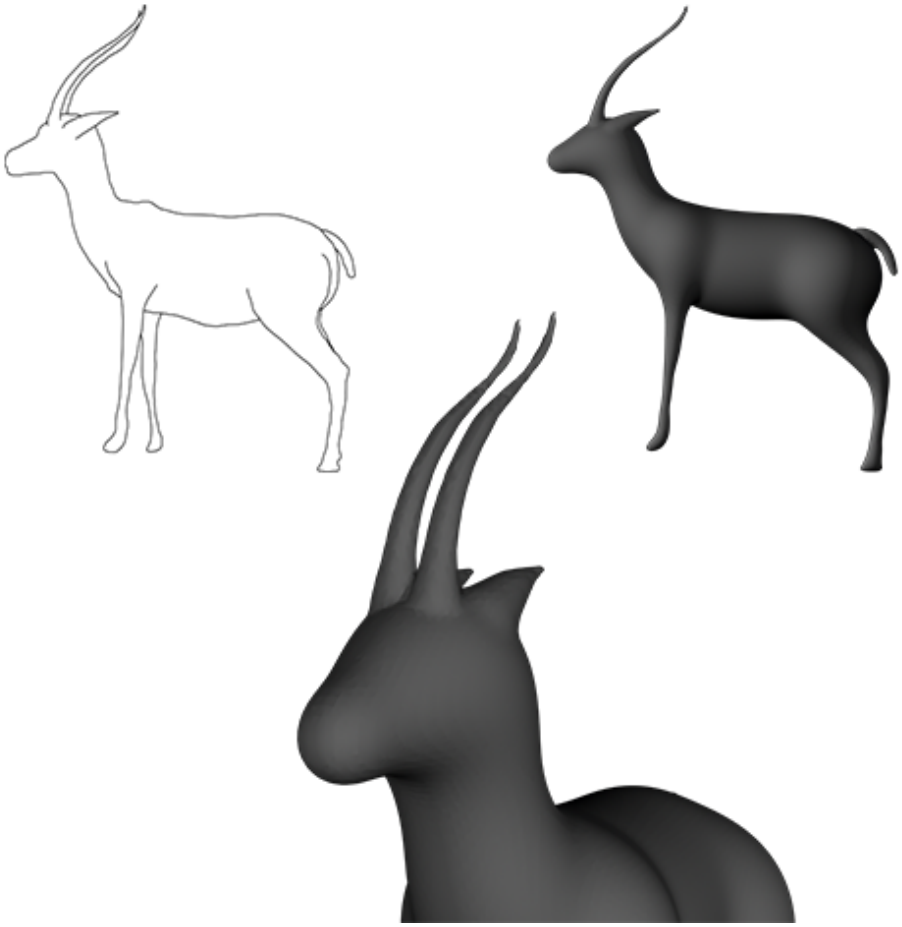


Figure 4.10 – Different views of the 3D reconstruction of the gazelle sketched in the top-left.

To further validate our system we performed a preliminary user study with 4 different users ranging from beginners to a professional computer artist. Their sketches and resulting models are shown in Figure 4.13. They all found the concept of suggestive curves to delimit the limbs quite easy to understand. When asked how well the 3D model met their expectations, on a scale from 1–9, they gave a score of 7. The time spent to create the input drawings using the Pencil Curve Tool in Autodesk Maya was less than 2 minutes for all the drawings, with the exception of the rabbit, which took 10 minutes to draw.

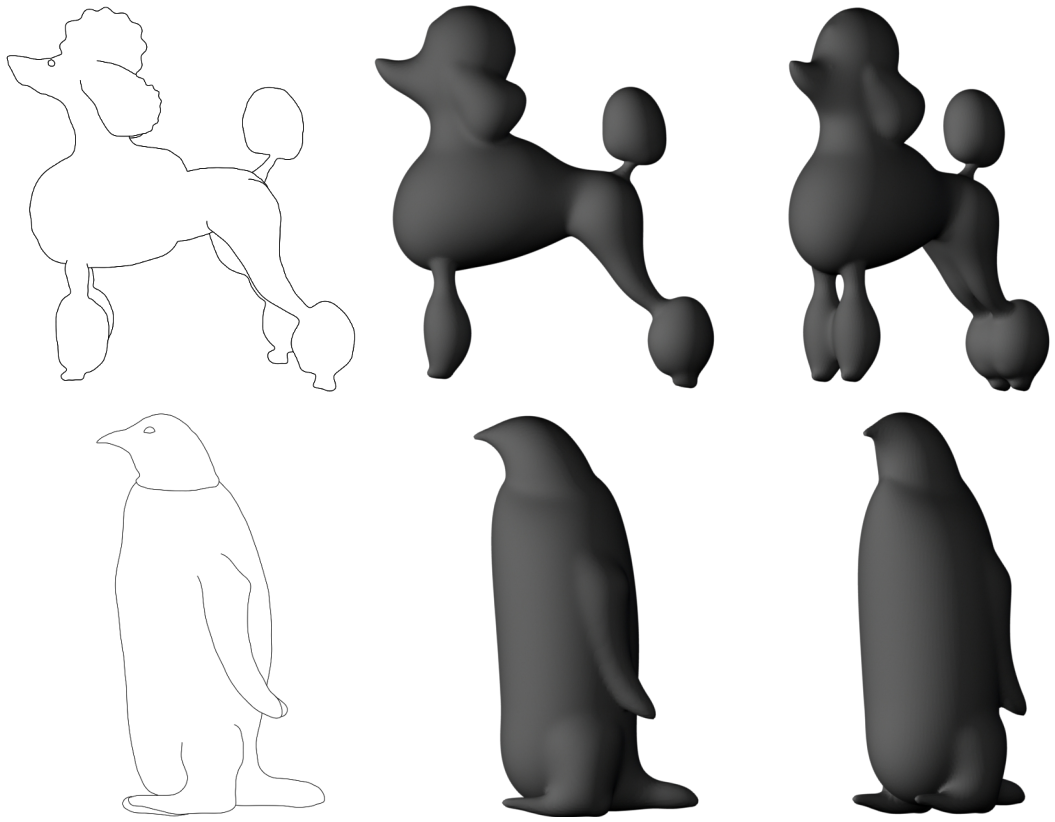
Limitations:

Our method has a number of remaining limitations.

First, we are limited in the range of sketches we are able to handle:

— The sketch cannot include adjacent internal parts (*islands* in our terminology) such as if the cat had two adjacent eyes. Otherwise, the classification algorithm will fail;

— A shape part cannot self-overlap in the sketch (such as an animal tail looping and self-occluding itself), nor overlap any other shape part (such as a tail occluding the animal's body). Our completion algorithm does not handle these cases. Note that



(a) (b) (c)

Figure 4.11 – Reconstruction of a dog and a penguin. (a) The input vectorial sketch, (b) the automatically generated result and (c) different view of the reconstructed model.

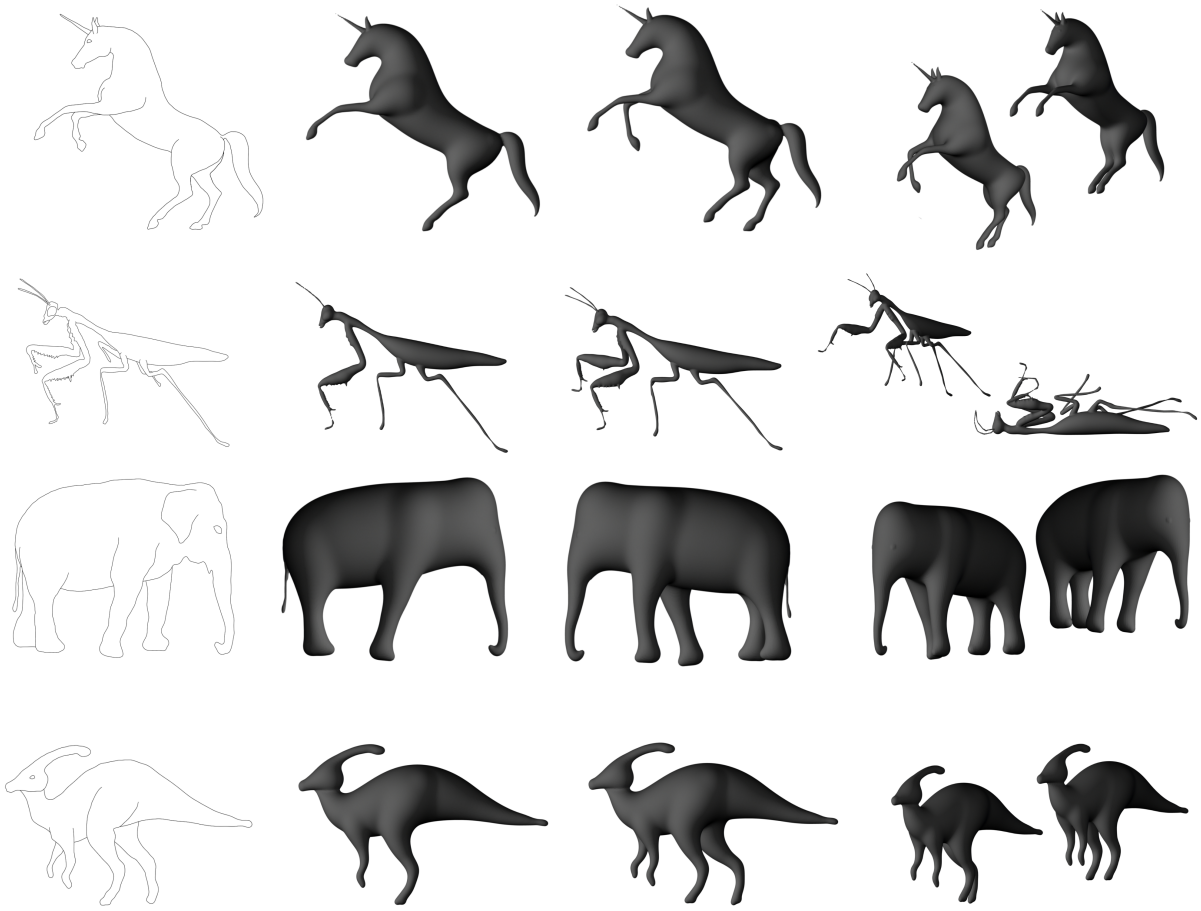
we partly avoid this occlusion problem by using symmetry when creating the limbs, but developing a more general solution would be desirable.

— Drawing pairs of suggestive contours is mandatory when two shape parts blend (such as at the top of a front leg, where it blends with the body), and these contours should be such that our completion algorithm keeps the reconstructed part of the contour inside the main body-shape. Although this holds true in most cases, we currently offer no guarantees regarding this point.

A second family of limitations concerns the 3D shape we output:

— All the shape parts we create have a planar medial axis, which we locate in a vertical plane. Although this works quite well for the legs, this method needs to be improved for parts such as the ears or the horns, which should preferably be oriented to lie in a direction that is normal to the surface.

— We only use segment skeletons for generating the implicit shapes, as opposed to surface skeletons, which restricts our construction to generalized cylinders. This prevents us from capturing the flat surfaces that one can readily observe on the flanks of many animals, or flat features such as elephant ears that cannot be reconstructed



(a) (b) (c) (d)

Figure 4.12 – Reconstruction of a mantis, an elephant and a dinosaur. (a) The input vectorial sketch, (b) the automatically generated result, (c) the model with the foreground limbs manually adjusted to the sketch and (d) different views of the reconstructed models. The mantis is shown with different poses in (d).

with our approach (see Figure 4.12).

— We reconstruct pairs of limbs using symmetric poses. Automatically fitting the reconstructed background parts is non-trivial since we do not reconstruct a rigging skeleton and most of the drawing of the background limbs is missing. Performing this fitting remains a challenging open problem on its own.

4.7 Future Works

In future work, we would like to address some of the limitations we listed. Being able to convert any sketch to the vector graphic complexes proposed in [Dalstein et al., 2014] would be an excellent intermediate step for our application. In addition, exploiting the 2D detection of partial symmetries in the cycles, using techniques such as the one by [Mitra et al., 2006] may be a promising direction for improving the pairing stage of our approach.

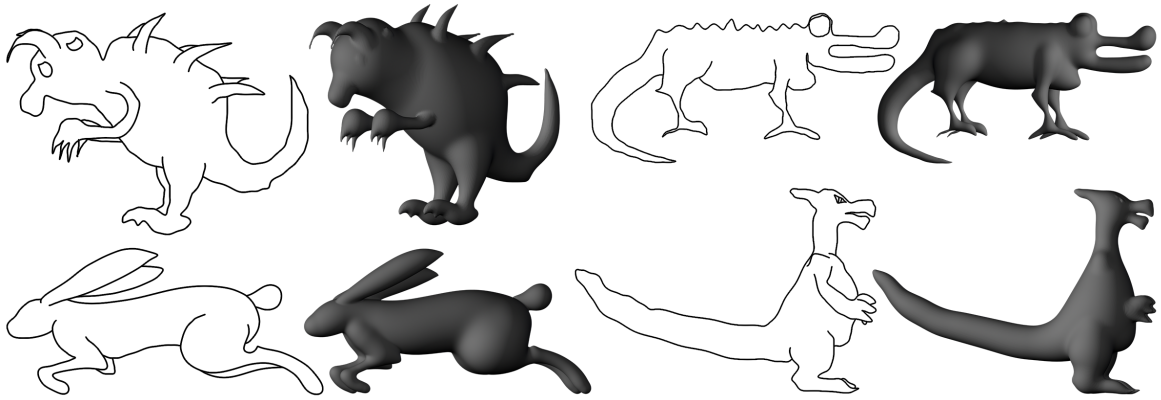


Figure 4.13 – Reconstruction of imaginary creatures and a rabbit made by four different users ranging from beginners to a professional computer graphics artist (the rabbit model).

This could also help for handling more complex inputs such as sketches in arbitrary views. A last caveat is that the technique we currently use is not able to reconstruct large flat parts such as wings. A direction is to use more complex skeletons including surface parts as was previously done in the context of reconstruction using convolution surfaces [Alexe et al., 2007].

4.8 Conclusion

We have presented the first method, to the best of our knowledge, for creating 3D animal models from a single sagittal-view sketch. Our method handles complex sketches with open curves, closed curves, and T-junctions. Once the main body part is selected, the method is fully automatic for reconstructing a symmetric version of the 3D shape. The reconstructed shape is inferred using only two hypotheses: shape smoothness, except at singular points appearing on the sketch contour; and structural symmetry. The resulting 3D model can then be posed using standard software or by directly manipulating its skeleton.

CHAPTER 5

Structuring and Layering Contour Drawings of Organic Shapes

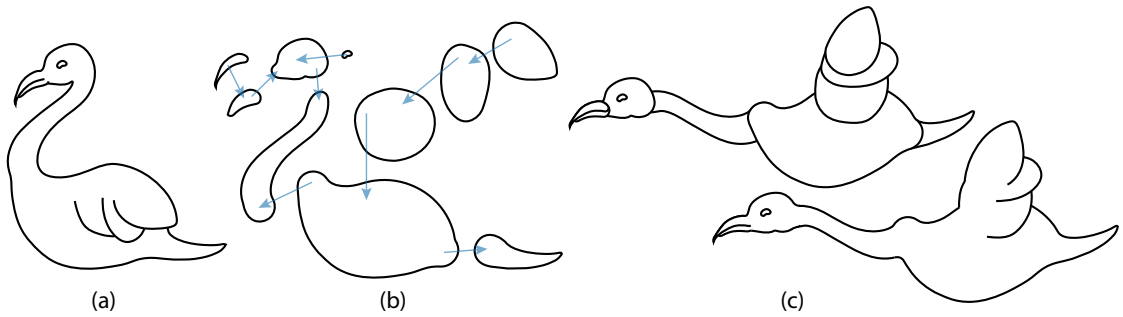


Figure 5.1 – An illustration of the steps of our automatic structuring and layering system. (a) The input drawing. (b) Structure analysis and part-completion estimate the constituent parts and their layering. The arrows represent the partial depth ordering ("over" relation). (c) Manipulation of the depth ordering as desired (top), followed by the union of these elements composed with the original internal contours to produce a new drawing (bottom).

5.1 Introduction

Contour drawings are commonly used for shape depiction. They are both easy to create and easy to interpret for a human and thus it makes them a convenient and expressive solution for visual communication. They are found in children's books, advertisements, technical books, and more. In contrast, these drawings are difficult for a computer to interpret. They usually depict silhouette curves and internal contours as well as expressive strokes, and they may represent fully visible, self-occluding, or locally hidden regions.

Many contour drawings are directly authored in vector graphics applications or are easily converted to a compatible representation using vectorization tools. Automatically decomposing them into distinct and simple structural parts, layered in depth (as in Figure 5.1 (b)), permits users to edit and manipulate the drawings intuitively by rescaling, moving, rotating, copying, and pasting parts without the need for intricate manual modifications and corrections which would otherwise be required for such operations.

We inspired this work from an intermediate step that artists sometimes use during the drawing process: drawing volume primitives () These primitives coarsly define the shape in an intuitive way. The method developed here can be seen as doing the inverse process which is to identify the salient volumes in a finished line drawing.

Figure 5.2 – Artists often use temporary guidelines (left) to pose characters and get a first visual of the volumes. It suggests that it could be an intermediate step in the reverse process: the understanding of shapes. Drawing by © Tracy Butler (lackadaisy).

In this chapter, we present an automated geometry-based method for extracting apparent structure and depth layers from clean contour line-drawings. We assume that the input drawing is intended to represent an organic shape, i.e., any free-form 3D solid with smooth connections between its 3D structural parts. The extracted depth-ordered structure is similar to the collection of blobs that artists sometimes use to temporarily define the construction lines and volumes of the shape they want to depict (see results of a web image search with the terms "tutorial drawing construction animals").We record additional information, namely, where these volumes blend together and where contours should be erased. This information can then be used for both current and nearby views editing to achieve new poses such as in Figure 5.1 (c). Although view-dependency may prevent the structure from being complete relative to the actual structure of the 3D depicted shape, we sill claim that it is a useful reference for editing the current drawing to create other postures of the shapes or near-by viewpoints.

The input drawing may be composed of silhouette curves as well as different categories of internal and external curves. They include internal open contours connected to silhouette curves, e.g., the contours of the feather groups in Figure 5.1 (a). Regions in the drawing that are demarcated by silhouette curves may also include a number of internal

regions depicting sub-shapes, possibly lying on top of one another, such as the eye of the swan in Figure 5.1. Highly ambiguous curves, such as disconnected internal curves and connected external open curves, are considered in our work as decorative curves. We also detect and discard internal elements that fail to define their own silhouettes (see Section 5.2.2).

Our three contributions towards solving structuring and layering problems for drawings are as follows:

— We describe a simple and efficient method for the aesthetic closing of sub-parts contours. This method provides a consistent solution when open-end points are not explicitly defined in the input drawing (Section 5.3).

— We introduce the *radial variation metric* (RVM), a novel part-aware metric for complex 2D drawings, inspired by the *volumetric shape image* used for shape segmentation of 3D models [Liu et al., 2009]. Its variation along the medial axis of parts in a drawing enables the identification of salient connections between sub-parts (Section 5.4).

— We describe a recursive algorithm enabling the successive identification of sub-parts in a complex sketch and their assignment to depth-layers (Section 5.5). The key insight lies in processing the possible junction zones between the identified sub-parts in a specific order based on the types of contours involved. This enables us to handle cases of multiple connected internal contours forming a tree-like structure, as can be observed for the swan wing in Figure 5.1.

The structure and layering information we obtain can be used to represent and edit the input sketch in current or nearby views (Section 5.6).

5.2 Overview

Our method decomposes an input drawing into a set of 'structural parts', layered in depth, and possibly overlapping in 2D, together with rules for combining these. The decomposition is based on connected internal silhouettes and inner closed contours in the drawings, which provide explicit indicators of part layering; the resulting parts are intended to be meaningful for artists as shape-defining-volume silhouettes, and can be used for editing and posing in nearby views.

This section introduces the terminology we use throughout this chapter and defines the assumptions we make on the input drawing, and presents the main features of our algorithm.

5.2.1 Terminology and assumptions

The input of our method is a vector line-drawing $\mathcal{D}$ defined in the (x, y) plane as a set $\mathcal{C}$ of parametric curves that may only intersect at their endpoints. The drawing may be either directly created in this form or obtained from a rasterized drawing using a vectorization algorithm [Noris et al., 2013] and then cutting curves at all intersections. Paramatric curves are also uniformely sampled as polylines for some of the further processing. In the following, we therefore refer to points along the curves $\mathcal{C}$ as *samples*.

As in SmoothSketch [Karpenko and Hughes, 2006], our algorithm is designed to handle contour-drawings of smooth, closed shapes, which we refer to as *organic shapes*. However, to be able to handle a larger category of drawings, we also allow them to include specific categories of decorative curves such as those often used in cartoon drawing.

Therefore, our algorithm includes a mechanism for the automatic detection of contour curves within $\mathcal{C}$. Since we are not asking for any additional information (such as contour orientations) from the user, we assume that the depicted shapes have no surface-to-surface contact, have genus 0, and are not self-overlapping (e.g., no animal's tail passing under and behind the body and forming a new background region).

The following terminology is specific to this chapter (see Figure 5.3):

Contour Graph: It is the planar graph structure formed by the contours and their intersections. Each counterclockwise face cycle of this graph corresponds to a region $\mathcal{R}_\rangle$.

Part: Structural parts (or parts) are the 2D counterparts of the structural elements of the 3D shape represented by the drawing, as depicted in Figure 5.1(b). Our goal is to extract them.

Decorative elements: These include all curves in $\mathcal{D}$ that are not visible contours of the depicted shape, but can instead represent ornamental details, 1D elements such as hair, or strokes used to represent bas-relief carvings. In our case, a curve is considered as a decorative curve if it falls in any of the following categories:

- Inner isolated subgraphs that do not contain external contours when processed as independent input drawings (in purple in Figure 5.3). Although they might contain or actually be inner contours, such subgraphs are highly ambiguous. We leave interpreting and processing them for future work.
- Inner trees of curves that are connected to the external contour of a region, but without tangent continuity (blue curves in Figure 5.3)
- Trees of curves located outside of the region they are connected to (green curves in Figure 5.3, which we call *hair*).

In our method, the decorative curves are identified and ignored from further contour processing, but are kept in the description of the corresponding to-be-segmented part.

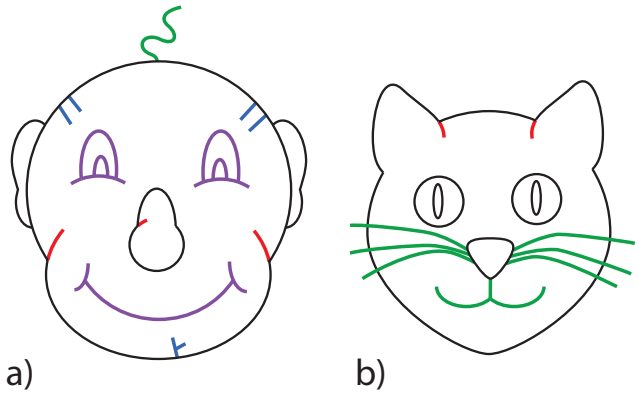


Figure 5.3 – Stroke classification. Red curves are suggestive contours; green curves are hair; blue and purple curves are part of non-hair decorative elements. The remaining contours, in black, are part of the external silhouettes of shape parts.

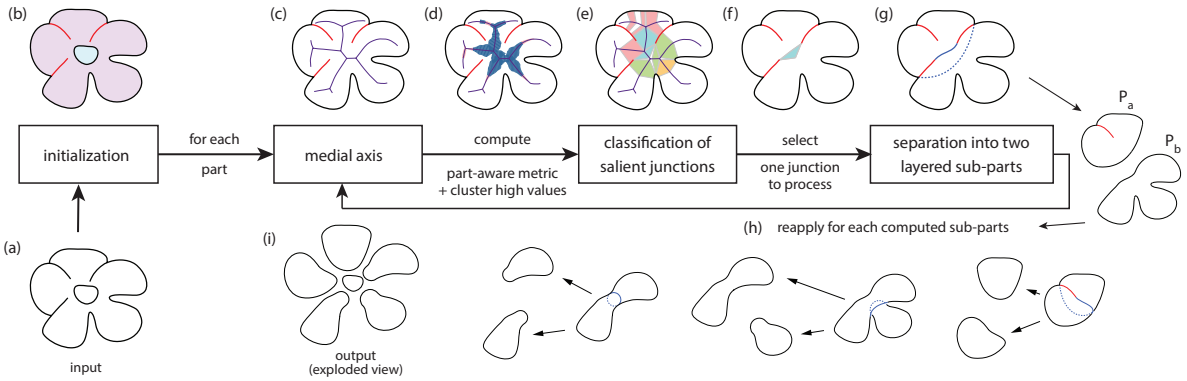


Figure 5.4 – Processing pipeline: The input is (a) and the output is (h) with partial depth ordering (here depicted in exploded view).

5.2.2 Processing pipeline

Let $\mathcal{G}$ be the contour graph, the planar half-edge graph defined by the curves $\mathcal{C}$ that constitute the input drawing $\mathcal{D}$. As in standard planar graph processing, each half-edge corresponds to a given orientation of a curve. If half-edges are part of a closed contour, they are considered to lie respectively in the interior and in the exterior of the corresponding face cycle of the graph. In the remainder of this section we use both "edge" or "curve" to denote the edges of $\mathcal{G}$, depending on context.

The goal of our method is to process $\mathcal{G}$ in order to extract and progressively refine the set $\mathcal{P}$ of 2D structural parts of $\mathcal{D}$, as well as the associated partial depth ordering $PO_{\mathcal{P}}$ expressing their relative depths. More precisely, $\mathcal{P}$ is defined as: $\mathcal{P} = \{P_i = (e_i, \mathcal{S}_i, \mathcal{O}_i)\}$, where e_i is the external silhouette contour (as a list of edges) of P_i and where $\mathcal{S}_i$ (respectively $\mathcal{O}_i$) is the subgraph of $\mathcal{G}$ corresponding to the suggestive contours

(respectively, the decorative elements) located within P_i, or attached to it.

Our processing pipeline, detailed below, is depicted in Figure 5.4. Starting with an initial set of silhouette-complete parts extracted at the initialization stage, our algorithm recursively decomposes each part into simpler sub-parts and updates PO accordingly. This is done until none of the parts in $\mathcal{P}$ has any suggestive contour left (for all i, $\mathcal{S}_i$ is empty).

Initialization:

We aim at initializing $\mathcal{P}$ to a first set of structural parts, such as the heart of the flower versus the part with all its petals in Figure 5.4 or the head versus the ears and the nose of the head in Figure 5.3 (a), and to set $PO_{\mathcal{P}}$ to the corresponding partial depth ordering. This involves completing the contours of partially hidden parts, such as the ears. Although this decomposition and figural completion problem was already solved in the past [Williams and Hanson, 1996], it was for drawings only depicting visible silhouette outlines of occluded surfaces. To handle more complex cases of cartoon drawings with decorative lines such as the whiskers in Figure 5.3 (b), we extend the original method into a four-stage process:

Firstly, if $\mathcal{G}$ contains any vertex v of valence 4 or more (such as vertices where the whiskers of the cat cross the head's contour in Figure 5.3 (b)), we convert $\mathcal{G}$ into a non-planar graph by dissociating the curves at v, enabling us to further process the whiskers as decorative elements attached to the nose. This is done based on Gestalt's perceptual rule of continuity, as follows: For each vertex v of valence 4 or more with adjacent edges in $\mathcal{G}$ that correspond to pairs of tangent-continuous curves, we join each pair of curves at v into a single curve. This disconnects the pairs of curves from each other, enabling them to be attached to different structural parts. A constraint is also set to prevent more than one pair of merged curves from being interpreted as a contour curve in further processing, since this would violate our hypotheses of organic shape depiction (e.g., two overlapping circles do not correspond to any valid contour of organic shape, and are thus interpreted as a single 2D region crossed by a closed decorative curve). After this stage, any connected component of $\mathcal{G}$ that still contains a vertex of valence larger than 3 is considered in its whole as a decorative element, since a set of curves that connect without any tangent continuity cannot include silhouettes of smooth organic shapes.

Then, we process each remaining connected component $\mathcal{CC}_j$ of $\mathcal{G}$ to separate contour curves from curves corresponding to the associated suggestive silhouettes or decorative elements.

Since the input drawing is supposed to contain no self-overlapping part, this can be done by simply moving every edge whose half-edges both belong to the same face cycle (ie. lie in the same 2D region) from $\mathcal{CC}_j$ to another subgraph $\mathcal{A}_j$, which gathers candidate edges for either $\mathcal{S}_i$ or $\mathcal{O}_i$. Note that this operation may split $\mathcal{CC}_j$ into smaller connected components, since the edges moved to $\mathcal{A}_j$ may include bridges between different

subgraphs. In this case, $\mathcal{CC}_j$ and the corresponding $\mathcal{A}_j$ are split into smaller subgraphs. We decide to which sub-connected component the bridge sub-graph should be associated to by looking at its tangent continuity with the neighboring curves: We insert the bridge into the $\mathcal{A}_k$ set associated with the subgraph $\mathcal{CC}_k$ of $\mathcal{CC}_j$ to which it has tangent continuity at one of its endpoints (e.g., a suggestive curve partially hidden by the contour of an inner part). If there is tangent continuity at both ends, the decision is taken at random. If there is none, the bridge is considered as a decoration and is attached to the subgraph corresponding to the contour of the region where it lies.

At this stage, each $\mathcal{CC}_j$ should only contain contour edges (for instance, the drawing in Figure 5.3 (a) is split into two connected components, namely (1) the nose and (2) the head plus ears, where only black curves remain). This enables us to use the existing algorithm in [Williams and Hanson, 1996] to split then into structural parts, by using T-junctions to identify partially hidden parts (such as the ears) and smoothly complete their contours. Failure cases of this algorithm and other limitations are be discussed in Section 5.6.

In contrast with previous work, we use our own efficient solution, presented in Section 5.3, to compute closure curves. All resulting parts $\mathcal{P}_i$ are stored in $\mathcal{P}$ together with their contour edges e_i. The corresponding partial depth ordering information is added to $PO_{\mathcal{P}}$. We also add temporary depth relations with the remaining connected components, depending on the number and order of intersections with each other components to reach the background of the drawing with a line (parts with the same number of intersections being considered at the same depth level).

In a final stage, the set $\mathcal{S}_i$ of suggestive contours of each part $\mathcal{P}_i$ is extracted from the set $\mathcal{A}_j$ associated to its former connected component $\mathcal{CC}_j$, by selecting curves with smooth T-junctions with contours e_i and that lie within $\mathcal{P}_i$. The other curves in $\mathcal{A}_j$ connected to e_i are classified as decorative elements and added to $\mathcal{O}_i$. Lastly, every curve in other $\mathcal{A}_k$ sets that were initially crossing any of the current contour curves are stored in the set $\mathcal{O}_k$ of the part in which it is located since they no longer can be contours.

The set of parts $\mathcal{P}$ is now ready for further decomposition.

Recursive part decomposition:

The core of the algorithm is a recursive loop that processes each part in $\mathcal{P}$ and recursively decomposes it into sub-parts. This enables us to process complex suggestive contours indicating embedded sub-parts, such as the wing of the swan in Figure 5.1: The full wing is extracted first and is then recursively split into partially overlapping sub-parts. The recursive loop proceeds as follows:

For each part P in $\mathcal{P}$, we identify the salient potential junction zones between sub-parts, and iterate from best-to-worst until a valid pair (P_a, P_b) of sub-parts is identified. Missing contours are then inferred for P_a and P_b (see Figure 5.4 (g)) and the depth

ordering relation between them is added to $\mathcal{PO}$. P_a and P_b are added to the list of parts $\mathcal{P}$, enabling us to recursively apply this process until no further decomposition is possible (Figure 5.4 (h)).

This recursive decomposition method raises three challenges, leading to our three key technical contributions. (1) The description of a robust method for completing the contours of the extracted sub-parts in a perceptually valid way. (2) The design of an effective metric for identifying the salient junction zones between the sub-parts of a 2D shape. In contrast with previous work, the 2D shapes we process may include suggestive contours, which are not constrained to come in even numbers, or to be small cusps. (3) The definition of an order in which to process the identified alternative solutions for segmentation into sub-parts. This order is important for extracting consistent sub-parts as it enables us to reuse the same algorithm in a recursive fashion.

Our aesthetic and efficient contour completion is presented in Section 5.3 and we discuss our new metric for identifying junctions in Section 5.4. Section 5.5 details our recursive structuring algorithm. It makes use of our new metric and completion method, with an emphasis on the priority order we set for processing possible junctions.

5.3 Aesthetic and efficient contour completion

In this section, we present the completion method we use for closing contours of both partially occluded structural parts (initialization step) and of sub-parts extracted during recursive part decomposition. Although not proved to be the best possible perceptual completion method, our solution is simple, efficient, and produces adequate results in practice.

5.3.1 Scale-Invariant MVC

We first note that perceptually pleasing contour completion is a different problem from that of completing illusory contours, e.g., [Williams and Hanson, 1996]. We seek to find aesthetic curves that are appropriate for the editing and subsequent animation of the drawing, during which hidden or omitted contour segments may become visible. In the literature, both curves minimizing the total curvature (i.e. "smooth" curves, MEC's) and curves minimizing the total variation of curvature (i.e., "fair" curves, MVC's) have been proposed as possible solutions to the problem. We choose to use MVC's because they tend to form more circular arcs; these are particularly well suited to later editing operations.

Let A and B be the two end-points of the open contour to be connected, along with corresponding unit tangent directions T_A and T_B as shown in Figure 5.5 (a). Our goal is to generate a perceptually plausible curve between A and B that best matches an inferred

silhouette for the resulting part.

We define the curve connecting these two input points as follows. Let B_{AB} be a Bézier cubic curve connecting A and B, defined by the four control points (A, P_1, P_2, B), and whose tangents are aligned along the unit vectors T_A and T_B, i.e., $P_1 = A + c_1 T_A$, and $P_2 = B + c_2 T_B$. We optimize the free parameters c_1 and c_2 in order to minimize a "fair" curve functional that is a variation of the SIMVC energy originally introduced in [Moreton, 1992] as:

$$E_{SIMVC-Moreton} = \left(\int \mathrm{d}s\right)^3 \int \left(\frac{\mathrm{d}K(s)}{\mathrm{d}s}\right)^2 \mathrm{d}s \tag{5.1}$$

where $\left(\int \mathrm{d}s\right)^3$ is the product of a regularization term $\left(\left(\int \mathrm{d}s\right)/\|B-A\|\right)^3$ and scale-invariance term $\|B-A\|^3$. This regularization term relies on the cube of the scale-relative arc length of the curve. We increase the regularization term's exponent from 3 to 5 in order to reward slightly shorter curves that avoid cases where closure curves would slightly jut out from the desired boundaries and thus propose to minimize the following energy:

$$E_{SIMVC} = \frac{\left(\int \mathrm{d}s\right)^5}{\|B-A\|^2} \int \left(\frac{\mathrm{d}K(s)}{\mathrm{d}s}\right)^2 \mathrm{d}s \tag{5.2}$$

The use of Bézier curves guarantees that the curve lies in the convex hull of its control points and is therefore well suited to interactive sampling and intersection queries. The choice to optimize its parameters with the SIMVC functional produces a scale-invariant result.

In practice, we use the standard Gauss-Kronrod quadrature to numerically integrate the different integral terms. Powell's method is used for minimizing E_{SIMVC}.

5.3.2 Efficient implementation

Given that the completion curves we compute are invariant to scaling, translation and rotation, there remain only two configuration parameters, namely θ and φ. They are the oriented angles formed by the two tangents with respect to a line between the two points to be connected, as shown in Figure 5.5 (a).

The SIMVC energy of our curve defined as a function of these two parameters is continuous and smooth in the sub-space of non self-intersecting curves, as shown in Figure 5.5 (b). This enables us to precompute a table of the different SIMVC Bézier cubic curves, with their SIMVC energy and parameters c_1 and c_2, as a function of θ and φ. Parameters can then be interpolated, with either bilinear or bicubic interpolation, to provide both an approximated curve and good initial parameters for the final curve computation, leading to an important speedup of the gradient descent with two variables.

In practice, Bezier curves enable us to quickly detect invalid contours: indeed, closures for partially occluded or partially occluding parts should not protrude outside of the union

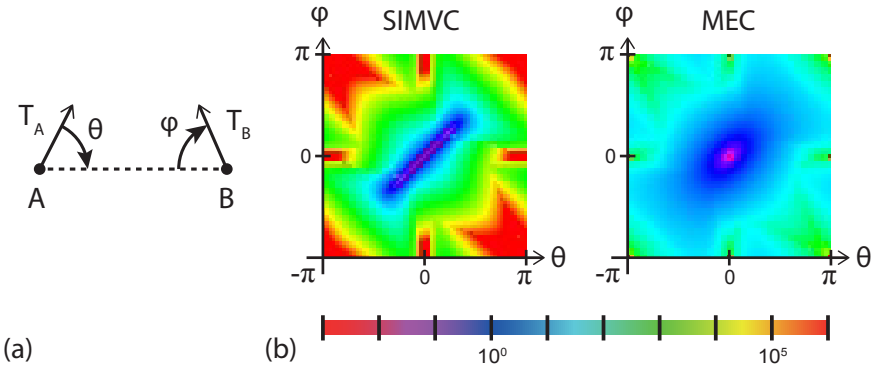


Figure 5.5 – To optimize the computation of our SIMVC curves, we precompute a table of the energies and associated parameters that the curve can take as a function of the two angles θ and φ defined in (a). We illustrate the sampled function SIMCV energy values and compare them with the MEC energy that minimized total curvature.

of the related parts thus we look for intersections. To avoid unintended intersections we rotate tangents, at the points to be connected, inwards and by a small angle, keeping the alignment of tangents almost imperceptible. During our experiments, we ended with a value of 2 degrees.

5.4 Extraction of Salient Junctions within a Part

While previous work already addressed the segmentation of 2D shapes into perceptually salient sub-parts, these methods do not tackle the segmentation of shapes carrying extra structural information in the form of suggestive contours. This is the problem we are addressing here.

5.4.1 Salient Junctions

In the remainder of this paper, we define a **salient junction** as a region where the drawing of a part exhibits a perceptual change, enabling it to be divided into two sub-parts in a perceptually consistent way.

We define **junction boundaries** as being the portions of the part contours that delimit such a junction. These portions can either be a point (e.g., the open end of a suggestive contour) or a contour segment, as will be the case when the exact point where the segmentation should occur is unclear.

A common approach to 2D shape segmentation that we have seen in Section 3.3.3, is computing junction boundaries as the segments of the contour of maximal negative curvature. However it cannot capture approximate regions such as those indicated by the green and red junction boundary in Figure 5.6 (a), since the red one is not a curvature

maximum and thus would not be detected. Similarly, segmentation methods that directly make use of the branches of a skeletal representation such as the Medial-Axis Transform to identify sub-parts, also fail in a number of cases, as shown in Figure 5.6 (b,c). Fortunately, such correspondence between a Medial-Axis' branch and a structural part remains generally true for foreground structural parts delimited by suggestive contours, such as the middle sub-part in Figure 5.6 (b). The solution we describe below therefore builds on skeletal representations, while being based on a new metric.

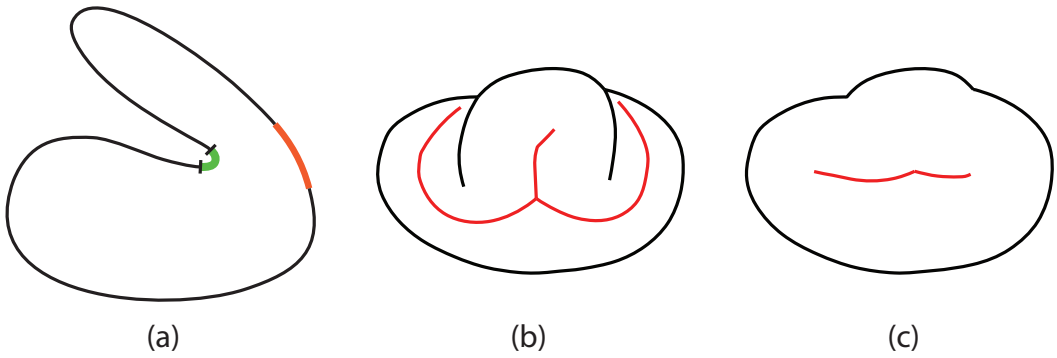


Figure 5.6 – (a) A maximum of negative curvature (in green) defines a junction boundary on the contour but its visual counter-part boundary (in red) shows no such maximum. (b) S-skeleton of a part (in red), i.e. Medial-Axis considering internal silhouettes: the lower structural sub-part is not captured, while the middle sub-part at the top is. (c) Processed Medial-Axis of the external contour of (b), where no branch is associated with the top structural sub-part.

5.4.2 Radial Variation Metric

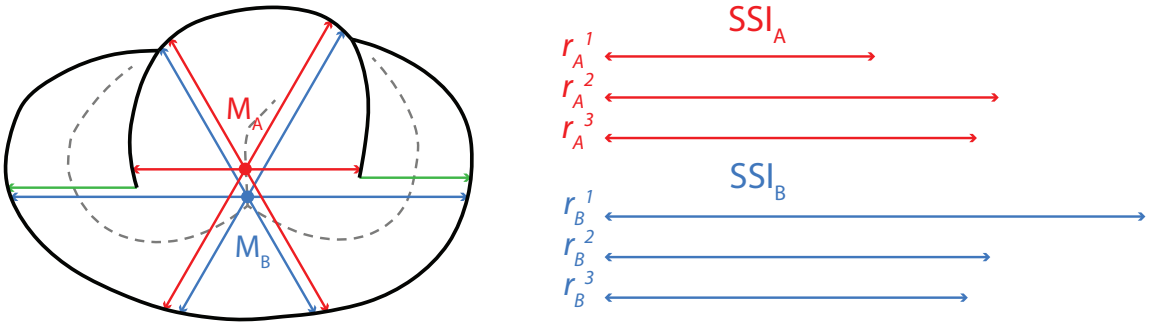


Figure 5.7 – Example of SSI for points M_A (resp. M_B) constructed by measuring the local reaches r_A^k (resp. r_B^k) in the k'th direction of a uniformly sampled set of m directions (here $m = 3$). Some discontinuities of reach are present and shown here as green arrows.

Our new parts-aware metric is inspired by a similar metric [Liu et al., 2009] developed

for 3D shape segmentation. In particular, it makes use of a *Surfacic Shape Image*, a modified 2D version of the Volumetric Shape Image (VSI) introduced in [Liu et al., 2009]. However, we tailor this more specifically to our problem, as detailed below.

We define the *Surfacic Shape Image* (SSI) as a signature of silhouette visibility from a given point of view inside a part of the input drawing (see Figure 5.7). A signature SSI_i is defined as a set of m distances l_i^k between closest points $p0_i^k$ and $p1_i^k$ to the point i in the k^{th} direction of a pre-defined uniformly sampled set of m directions. From 60 to 100 directions per set are used for the different examples shown in this chapter. Both external contours and internal contours are considered at this step, while decorative curves are discarded.

Similar to the 3D case, the local change of SSI (differential of SSI) between two neighboring points can be used to detect junctions between two structural shape parts. We propose to compute this differential between points A and B as follows:

$$\Delta(SSI)_{A,B} = \frac{1}{\sum_k w_{k,A,B}} \sum_{k=1}^{m} w_{k,A,B} \frac{\left| l_A^k - l_B^k \right|}{\|B - A\|} \tag{5.3}$$

where

$$l_i^k = \left\| p1_i^k - p0_i^k \right\| \tag{5.4}$$

Discontinuities near open ends of suggestive contours as depicted in Figure 5.7 generate outliers. To tackle this problem, we fit a Gaussian to the distribution of such values (as in [Liu et al., 2009]) using the weights w_k defined as follows:

$$w_{k,A,B} = \begin{cases} e^{-(d_{k,A,B}-u)^2/(2\sigma^2)}, & \text{if } d_{k,A,B} < u + 2\sigma \\ 0, & \text{if } d_{k,A,B} >= u + 2\sigma \end{cases} \tag{5.5}$$

where $d_{k,A,B} = \left| l_A^k - l_B^k \right| / \|B - A\|$, u is the mean of the $d_{k,A,B}$'s, and σ the standard deviation.

Note that in contrast to [Liu et al., 2009], $\left| l_A^k - l_B^k \right|$ is not squared in Equation (5.3), in order to make the equation more linear. In practice, we regularize this term by the distance between neighboring points of view, in order to allow for similar results whatever the sampling resolution at which this measure is used. Thus $\Delta(SSI)$ tends to be scale-invariant for high resolutions when there are no discontinuities of visibility. Salient discontinuities, even pruned, should still contribute to the $\Delta(SSI)$ in salient junctions. However, we note that their contribution is related to the number of rays comprised within the parallax angle of a discontinuity location seen from neighbor points. Thus we propose two sampling methods that correctly balance this contribution. The first uses a uniform spacing that is relative to the drawing size (in practice $0.005 * D_{height}$) and the second uses a dynamic spacing relative to local Medial-Axis disk radii (in practice $0.02 * local\_radius$). While the first is closer to perceptual principles, the second allows for small details to be processed. Our presented results have been generated with the first method for better

readability of figures. We emphasize that even if it extends to the continuous case when there are no discontinuities, the resolution of SSI sample points should not be higher than that of the input polylines so as not to reflect the lack of curvature of the polylines (otherwise a wave like pattern can emerge).

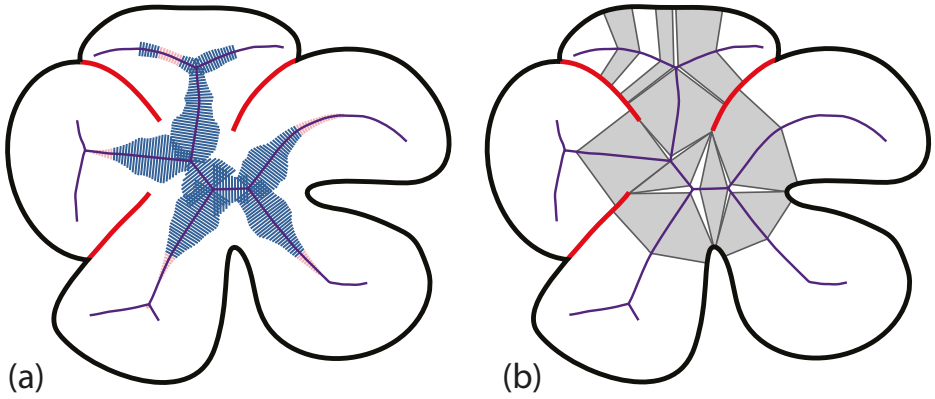


Figure 5.8 – (a) Computed d_{SSI} between vertices of each edge of the S-skeleton of a part, and represented for each skeleton edge as the scale of an orthogonal segment passing through the center of this edge (values are squared for visibility). (b) Salient junctions (in grey).

Let us now define a part-aware metric from the SSI. Note that we cannot reuse the method introduced by Liu et al. [Liu et al., 2009], where the distance used to segment a 3D mesh was defined as the integral of the VSI distance along a geodesic path between vertices. Indeed, in our case, there is no 3D surface to support and define the actual shortest path between two facing vertices on opposite sides of a shape. Therefore, our approach identifies salient sub-parts by defining a 2D Part-Aware metric *along the curves of a specific skeleton*, called the **salient skeleton (S-skeleton)**, as described below.

We initialize the S-skeleton of a shape part as the medial axis of the region bounded by the external contour and the suggestive contours of this part, as shown in Figure 5.6 (b). Decorative curves are discarded. As usual, it is defined as the locus of the disks of a Medial-Axis Transform (abbreviated MAT) which are the maximal disks that do not intersect the set of contours (see Figure 5.4). For the S-skeleton to only reflect the main shape feature, we proceed in a fashion similar to the Scale-Axis Transform [Giesen et al., 2009]. The goal is to locally remove small disks by considering those that can be covered by others in a version of the MAT with larger radii. The final result can be realized by computing the MAT of the grown shape defined by the union of scaled up disks from the initial MAT, and then scaling down the radii. However for reasons of computational efficiency and simplicity, we choose to iteratively remove disks from branch extremities in the grown MAT that are covered by others and then scale down the radii of the remaining disks. This saves the computation of a union and a MAT for similar results at the scaling

factor used in our algorithm, which is 1.3. Additionally, the associated contour points of these removed disks are given to the nearest remaining neighbor in order to keep a mapping between contours and the S-skeleton.

Given that the drawing represents the silhouettes of a volumetric, organic shape, the S-skeleton is a good candidate for extracting structural information about salient sub-parts. It enables us to recover junction boundaries that either lie along the external contour of the part, or at the open end of a suggestive contour. To correctly identify these boundaries, all the curves in the drawing, as well as the S-skeleton itself, are represented using half-edges. This enables us to assign the duplicate vertices on two sides of a suggestive contour to different branches of the S-skeleton, as illustrated in Figure 5.8.

Our new 2D part-aware metric, called **radial variation metric** and noted d_{SSI} is then defined over the S-skeleton as the integral of the SSI differential (Equation (5.3)) along the shortest path joining two skeleton vertices:

$$d_{SSI}(M_A, M_B) = \sum_{e \in E} \Delta(SSI)_e \tag{5.6}$$

where M_A to M_B are vertices of the S-skeleton and E the set of edges of the shortest path between them.

5.4.3 Salience junction detection

With the S-skeleton S being computed from a medial axis transform, each edge $e \in S$ is associated with two facing portions of the contours, i.e. the segments or point of the contours that could be generated by drawing discs from this specific edge of the skeleton. This enables us to use the S-skeleton to define both salient junctions (the regions we are looking for) and the junction boundaries that delimit them on the contours:

We initialize salient junctions as the 2D regions corresponding to segments of the S-skeleton with d_{SSI} values over a threshold k (see Figure 5.8 (b)). Thanks to the scale-independent nature of the metric, a single threshold value k is used regardless of the scale of the input drawing (we use $k = 0.45$ for all our results). These segments are stored using lists of edges of the S-skeleton. Since sharp extremities of structural sub-parts may correspond to large-but-irrelevant d_{SSI} values, we remove them in a second pass: Starting from S-skeleton extremities, we iteratively remove edges while their d_{SSI} values decrease. Increases in d_{SSI} values due to noise can lead to unwanted decomposition of pointy ends, but are mostly avoided by smoothing the d_{SSI} values along the S-skeleton first. Due to the nature of the SSI, sampling points near the middle of a punctually symmetrical transition will yield low d_{SSI} values compared to other transitions as illustrated in Figure 5.9. However, we note that the negative curvature cues on both sides is implicitly given by the diamond shape formed between the junction zones (Figure 5.9 (a)). We merge these salient junctions if the distances d_{C0} and d_{C1} between the junctions zones along contours

are both: inferior to the distance d_M along the S-skeleton, and inferior to the half of the average radius of the Medial-Axis disks of the junctions zones.

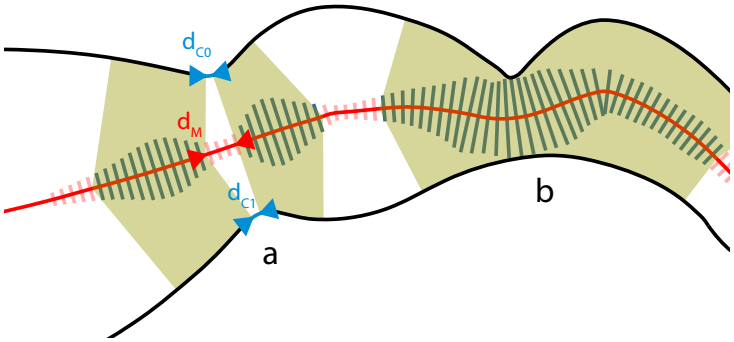


Figure 5.9 – (a) A salient junction misidentified as two junctions due to punctual symmetricity of the shape at the center of the transition (in near radial directions to the Medial-Axis). Both are merged before further processing. (b) A more common case of salient junction with no merging necessary.

5.5 Recursive part decomposition

Given the general methods that we have described for closing contours and for extracting salient junctions within a structural part, we now detail how a given part is decomposed, i.e., how its salient junctions are prioritized, and how the corresponding sub-parts are extracted and completed.

5.5.1 Prioritizing salient junctions

Decomposing a part not only requires extracting possible junction regions between sub-parts, but also assigning them a prioritized order. We achieve this via a classification of salient junctions, depending on the type of contour segments that contributed to this specific part of the S-skeleton.

Since they are defined by segments of the S-skeleton, each salient junction comes together with a pair of junction boundaries (the associated parts of the contours, possibly reduced to a point) found on each side of the skeleton. Salient junctions are classified as follows, based on the nature of this pair of junction boundaries (see Figure 5.10):

1. **Two segments of suggestive contours, that do not belong to the same tree of internal silhouettes:** The junction is either classified (SF, SF) and (SB, SB), depending on whether the suggestive contour's curves (a set of half edges) correspond to a front (occluding) or to a back (occluded) sub-part of the shape. The occluding side is given by the T-junction properties.

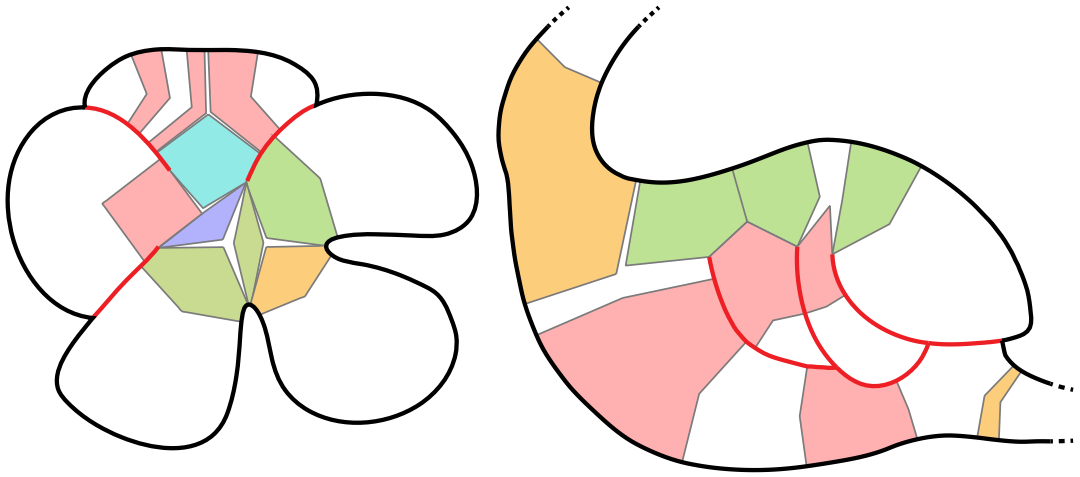


Figure 5.10 – Salient junctions: (SF, SF) in dark blue, (SB, SB) in cyan, (C, SF) in green, (C, C) in orange, discarded regions in pink.

2. **A pair formed by an external contour segment and a suggestive contour segment:** We only consider the case when the suggestive contour side corresponds to the front of the shape, denoted as (C, SF).
3. **Two portions of the external contour**: the junction is classified (C, C).

If a curve segment in a pair spans different types of contours, the associated salient junction is subdivided. Regions that do not fit into the categories described above (in pink in Figure 5.10) are discarded, since they have a bounding contour on one side and an occluding one on the other, and this case is not handled by our decomposition.

This classification is used to select the salient sub-parts to be extracted at each stage of the recursive part decomposition algorithm described in Section 5.2.2: (SF, SF), (SB, SB), (C, SF) and (C, C) salient junctions are respectively given highest to lowest priority. This enables us to give priority to sub-parts that are unambiguously in front of their neighbors, such as for the bottom-right part of the flower in Figure 5.4, before processing partially occluded sub-parts and those with weaker depth clues.

5.5.2 Processing complex suggestive contours

In addition to the sorting order we just defined, sub-parts defined by (C, SF) junctions need to be given a priority order. This enables us to handle cases where the suggestive contour forms a tree of branching curves, such as the swan's wing in Figure 5.1.

Let us look at the similar shapes on Figure 5.11: sub-shapes with high label values on the edges need to be extracted first, since they embed the other ones. This enables us to extract the full wing, which can then be progressively decomposed into three consistent sub-parts.

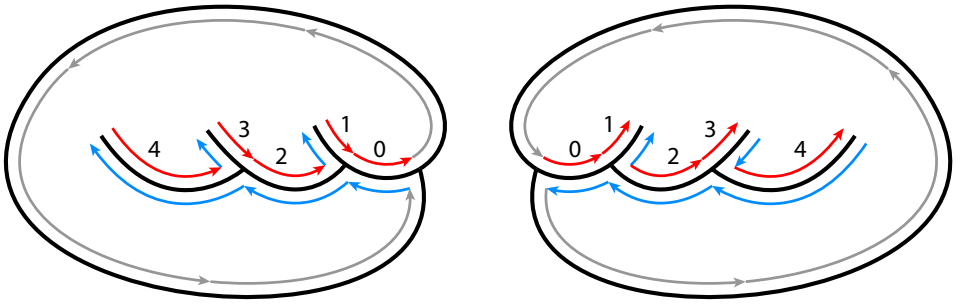


Figure 5.11 – Adding relative depth information along a suggestive contour, illustrated here for the case of a complex curve forming a tree. As shown here, the numbering must be made in forward or backward fashion depending on the root T-junction.

Given that suggestive contours represent internal silhouettes of volumetric sub-parts of a shape that smoothly blend with the parent part, they should have a G^1 continuous junction to the silhouette they are attached to (see Figure 5.11). The side of this smooth junction indicates which part comes above the rest. Therefore:

1. If the connection point with the external contour is only C^0, the curve tree is re-classified as decoration.
2. If G^1 continuity is detected, we use a traversal of the suggestive contour from the connection point to the open end, on the side of the G^1 continuous curve in order to enumerate and prioritize these half edges for decomposition.
3. During the traversal of the tree, only suggestive contours that demarcate a sub-part that is on the same side as the occluder at the root T-junction are allowed. Other contours are left-out as decorative strokes and are not processed by our algorithm.

Note that even in the case of complex suggestive contours that form a tree as in Figure 5.11, the suggested sub-shape is always on the same side of the curve, given that the organic shape hypothesis would otherwise be violated.

5.5.3 Part decomposition method

Decomposing a shape part at a given salient junction always involves generating two contours for closing the two resulting sub-parts. We use the terms *front closure* and *back closure* to refer to the closure curve that closes the sub-parts lying at the front and back, respectively, given the depth clues provided by the suggestive contours. To decompose a part, we first generate the two closure curves for all the junctions having the highest priority, using specific algorithms for each type of salient junction, as detailed below. If either of the resulting closure curves intersect the shape contour or if the front closure curve intersects a decoration curve, the current junction is discarded. Finally, the

junctions that result in most plausible closure, as defined by the minimal sum of their closure curves' energy, are selected for the decomposition. The resulting sub-parts are created and included in the partial depth set $\mathcal{P}$ according to their classification as a front or back closure curve.

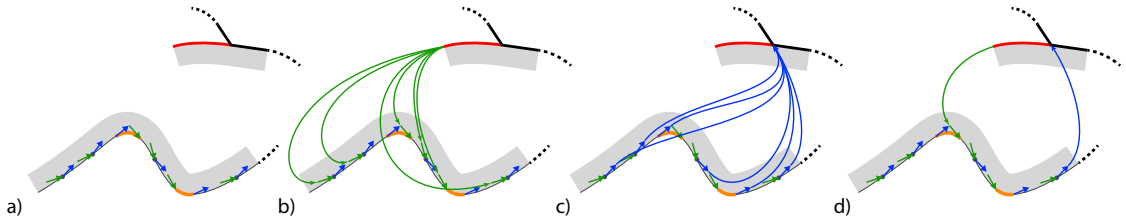


Figure 5.12 – Part decomposition at a salient junction. a) junction boundaries in this case are described by an external contour at the bottom and a red suggestive contour at the top. The external contour is uniformly sampled into a set of points and their associated pair of tangents. Corner segments (orange) samples are given two points instead of one to handle proper connection where curvature is the highest. b) all the closure curves corresponding to pairs of one sample point from the contour and the suggestive contour extremity are computed and their plausibility is evaluated for closing the rightmost sub-part. Curves that intersect contours are eliminated. c) the same closing procedure is applied to the other region, therefore the suggestive contour's T-junction is used instead of its extremity; d) resulting closure curves.

Note that the two newly created parts may have overlaps between their respective closure curves, but this is valid since they are each assigned a different depth layer. To assign such depth in the case of (C, C) salient junction without a relative depth cue, e.g., the bottom-right petal of the flower in Figure 5.4, we use the convention that the largest shape part should be in front, which is often the best choice when the resulting sub-shapes are to be animated.

While inferring the closure of a sub-part given two end-points and the associated tangent vectors is easy (Section 5.3), and can be done for connecting two suggestive contours (SF, SF) and (SB, SB) salient junctions, the connections in the (C, SF) and (C, C) cases are much more challenging. Indeed, the best pairs of contour points in the salient junction zone should be computed for the front and back closure curves. Our methods for solving these two cases are presented next.

5.5.4 Contour / Suggestive contour (C,SF) closure

Given that we are in the case where the suggested sub-part is on top, we compute all the possible closure curves that join the tip of the suggestive contour to the sample points on the facing contour segment in order to generate the front closure (Figure 5.12 (b)).

We also generate all the possible closure curves joining the contour segment with the T-junction at the base of the suggestive contour tree (Figure 5.12 (c)). Keeping only pairs of closure curves whose tips on the contour are not further from each other than the blending radius, we select the most plausible pair of closure curves of minimal energy using the sum of their SIMVC energies (Equation 5.2). This enables us to efficiently select the best pair of closure curves among the n^2 possible choices.

For this task we must define an adapted sampling to explore the space of possible closure curves. We first compute a blending radius that reflects the size of the transition between the salient sub-part of interest and the remaining part of the current shape. This radius is set as the average of radii at the two ends of the salient junction along the S-skeleton. We then identify *corner segments* within the junction boundary, defined as parts of the curve where curvatures are larger than the blending radius (orange curve segments in Figure 5.12 (a)). Corners are associated with specific virtual sampling points at two different locations, namely at the beginning and end of the curve-segment that forms the corner, and with two possible tangents depending on which of the two closure curves is being computed (Figure 5.12 (b) and (c)). Other contour parts are regularly sampled with a distance between consecutive samples equals to the half of the blending radius. For each such sample point we store the incoming (or outgoing) tangent vectors, with a small tilt outwards (or inwards) in order to avoid undesired intersections.

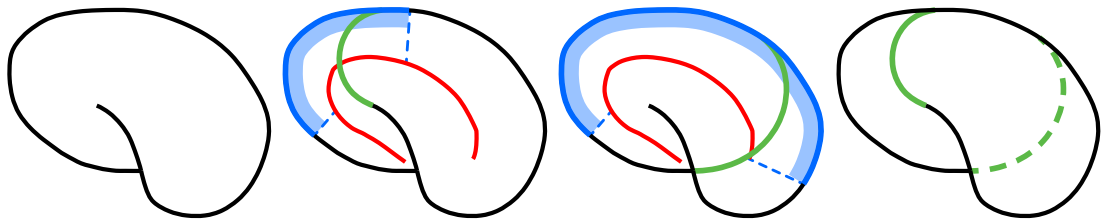


Figure 5.13 – (a) Example of a suggestive contour comparable to a cusp requiring a (C,SF) closure. (b) The front closure unknown end-point is searched along the blue contour part defined by the identified salient junction. (c) The transition is extended for the back closure. (d) Result of the decomposition with sub-parts sharing a wide part of contour.

When dealing with suggestive contours that could be considered as big cusps (Figure 5.13 (a)), the front closure is processed normally (Figure 5.13 (b)) while the the back closure requires to extend the relevant transition contour (Figure 5.13 (c)). The transition contour is extended by following its associated S-skeleton branch until either an extra branch is encountered, or the facing contour is a neighbor of the T-junction. This produces the result exposed in Figure 5.13 (d).

5.5.5 Contour / Contour (C,C) closure

We sample the boundaries of the salient junction in the same way as for the (C, SF) case. The two closure curves' extremities may be located anywhere on these segments. With n sample points on both contours, a naive method would lead to n^2 possible curves to generate for each of the sub-parts, and thus to n^4 pairs of closure curves to evaluate. To reduce the complexity back to n^2, we only consider the pairs of closure curves between a given pair of points, i.e., the same sample for the two curves.

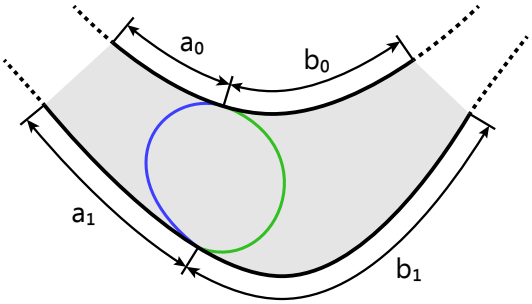


Figure 5.14 – For (C, C) closures we define a new coefficient for $\bar{E}$ based on the samples' positions relative to their respective sampling contours as defined in Section 5.5.5.

We use a variation on the energy in this case for selecting the most plausible pair of curves. Rather than selecting short curves, we wish to favor a decomposition close to the middle of the junction zone. Therefore, we use the energy $\hat{E}$ of the two curves to select the best closing pair, defined as:

$$\hat{E} = \left(1 + \frac{(1 - \frac{a_0}{0.5 * l_0})^2 l_0 + (1 - \frac{a_1}{0.5 * l_1})^2 l_1}{l_0 + l_1}\right)(E^0_{SIMVC} + E^1_{SIMVC})$$

with $l_i = a_i + b_i$ and a_i, b_i the junction boundary segments arc lengths shown in Figure 5.14, E^0_{SIMVC} and E^1_{SIMVC} the energies of the closure curves.

We also wish to reward the use of corners over a solution with two circular arcs forming a circle since it has an energy close to zero. Thus valid closures that use a sample at a corner as an implicit end point are given priority.

5.6 Results and discussion

Figures 5.1, 5.4, 5.19 to 5.20, and 5.21 show a variety of shape decomposition and layering results that are automatically computed by our method. Note that for Figure 5.19, we reused drawings from a recent paper [Bessmeltsev et al., 2015], showing that our method achieves the structuring and layering of such drawings without the need of any extra information, whereas a user-defined 3D skeleton was used in the original paper. Figures 5.1

and 5.15 (top-right) show even more challenging cases where some of the suggestive contours form a chain of T-junctions, requiring the labeling method of Section 5.5.2 in order to be properly processed. Figure 5.21 shows results computed on cat drawings found on the web using a simple query and vectorized using Adobe Illustrator.

Validation: Segmentation in depth ordered structural parts is a fundamental first step for further applications such as the edition of vector drawings with robust completion of partially hidden parts, 2D animation, or sketch-based 3D modeling. While developing such applications remains out of the scope of this research, we tested our method with two applications in mind: the conversion of the input drawing into a Vector Graphics Complex [Dalstein et al., 2014] that then enables easy and meaningful editing, and 2D vector animation. Posing and animation results are shown respectively in Figure 5.1 and in the supplemental video. The decomposition of the wing of the swan may not fit the perceived structure for every viewer, but wings are not known to be easily animatable in 2D.

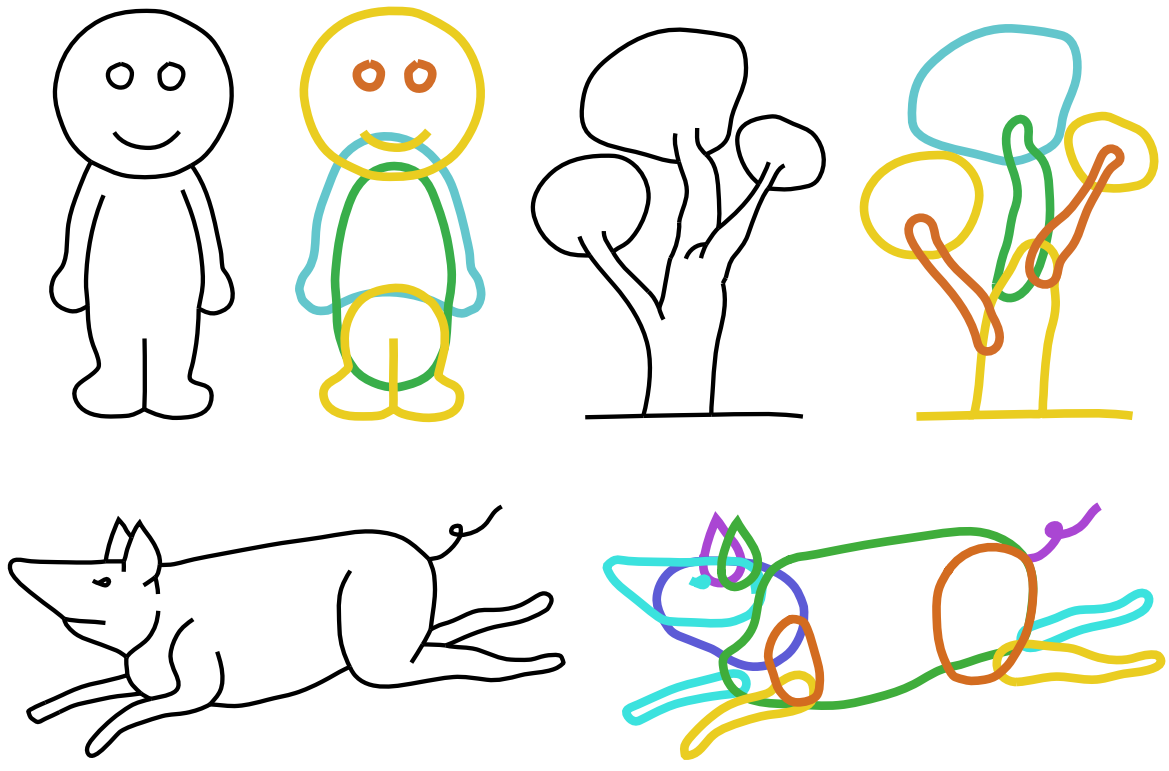


Figure 5.15 – Results on drawings of a cartoonish man, a tree and a pig. The two legs of the man are seen from a special view, thus the surface contact is classified as a decorative element. Warmer colors are in the foreground.

Discussion: Our method for structuring complex drawings performs as expected in most cases. We point out that once a drawing is processed, the union of the extracted parts does not necessarily exactly correspond to the initial outline since the blending between parts is not conserved. However, it is very similar and it would be possible to retrieve a similar outline by representing the parts' contours as iso-contours of 2D scalar fields and blending them together. Scalar fields with skeleton could also allow for easy directional rescaling of structural parts to allow for illusory 3D rotations. For instance it could help for improving the animation of our swan's wings.

In some cases, a valid drawing may be ambiguous and gives rise to several different interpretations. This is the case for the example in Figure 5.16, where the shape could either be interpreted as a boxing glove (b) or as a snail head protruding out of the shell (c). Our method will output a single result in such cases, the one in (c), because of the way we process complex suggestive contours. Some curves that would be processed naturally by a human as contours are not processed when there is no T-junction such as for the beak of the bird in Figure 5.1.

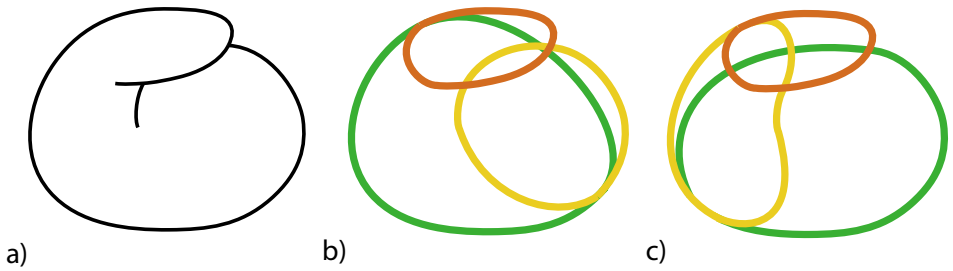


Figure 5.16 – An ambiguity of interpretation. Our algorithm can only produce the result (c) while (b) would also be a perceptually plausible result.

Lastly, similarly to the metric in [Liu et al., 2009], our d_{SSI} metric could also be used to define a distance between two points of the contour, which we would define as the d_{SSI} distance between their two corresponding S-skeleton's vertices. In future work we wish to explore the possible applications of this new metric to contour drawings.

Comparison with previous work: The closest work we can compare with is SmoothSketch [Karpenko and Hughes, 2006]. While they look for a plausible completion of the hidden contour hinted by cusps, we instead use the facing contour to smoothly wrap a curve around the hypothetical 3D shape as shown in Figure 5.17. We emphasize that the segment of our hidden closure that is near the T-junction is a plausible hidden contour. Thus, our method could benefit figural completion of cusps in SmoothSketch's failure cases. However, we think that both this algorithm and ours should be used in a complete application since hidden contour completion is more meaningful than our decomposition

for the case of cusps that are small relative to the local thickness of the shape, and specific cases such as the swan's wing.

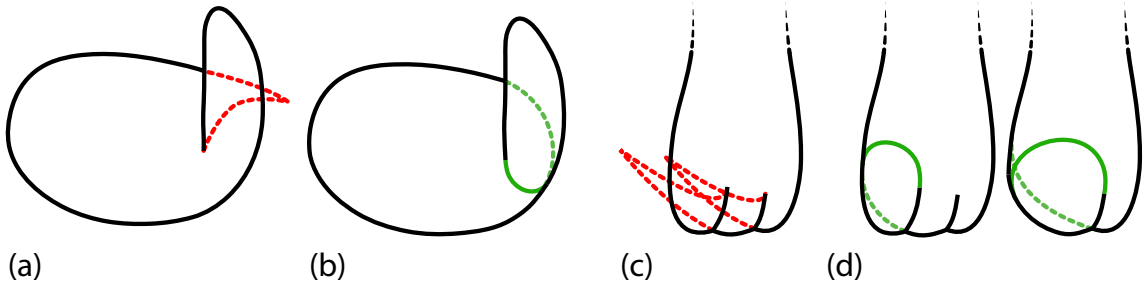


Figure 5.17 – Comparison of our results (b,d) with SmoothSketch's failure cases (a,c).

The recursivity of our decomposition hides some perceptual information such as similarity, grouping, symmetry. In the paw example in Figure 5.17(c,d), we perceive the similarity and symmetry of the fingers. However our algorithm first decompose the foreground finger, and considers the two others as a whole, thus the background finger is eventually perceived as big as the middle one once the first is extracted. Designing a global method from our recursive one is a non trivial problem since the closures are interdependent in many cases.

We also show results on inputs used to test our previous method in Figure 5.20 (top, middle). Though the segmentation is similar, our initialization step cannot complete complex hidden contours. This limitation would locally require a completion similar to the one used in [Sýkora et al., 2014] but it is non trivial to combine this method with figural completion to be able to completed parts with distinct visible regions in the absence of distinct similarities between these regions such as color or grouping annotations. However our algorithm tackles the case of sub-parts hinted by single suggestive contours as shown in Figure 5.20 (bottom).

Limitations: Even though our method is giving good results in most of the cases, we have few limitations as well. One main problem is the cyclic arrangement of parts over one another. Figure 5.18(a) shows an example in which the petals are overlapping to one another in a cyclic fashion. In this case, layering cannot be done using the proposed method and makes our part decomposition phase to fail. Since we are assuming that the occlusion results in either T-junctions or cusps, the more complex junctions are not processed in the current system.

Many limitations are found in the initialization step, when drawings carry ambiguities at curve intersections. Notably when T-junctions are not well defined either due to special view or surface contact as in Figure 5.18 (b,c) and Figure 5.21 (2,3), or misleading because the hypothetical occluded contour is in fact a texture change as in Figure 5.18 (e, left).

The latter limitation can be manually worked around by erasing a part of the stroke as in Figure 5.18 (e, right).

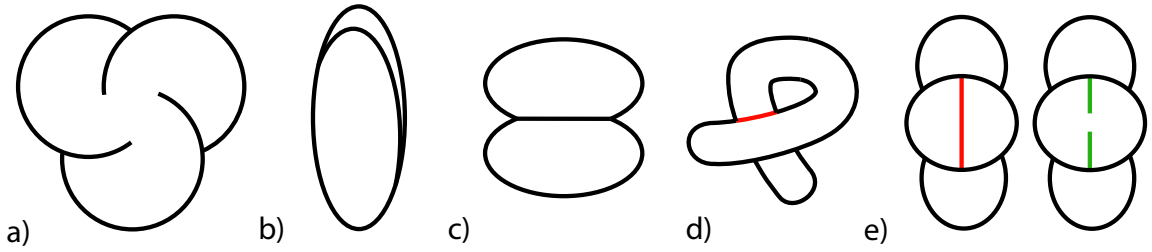


Figure 5.18 – Different limitation cases of the initialization step (b, c, d, e) and recursive decomposition algorithm (a).

5.7 Future Works

Our current method can be improved in various directions. One of them is to include user interventions to solve ambiguous cases (recognize whether two concentric circles represent two spheres or a torus), and to do assisted-segmentation as in [Sýkora et al., 2014]. Another interesting roadway is to use decorative curves as suggestive contours. Figure 5.22 shows a case in which the decorative curve represents a suggestive curve and our method would ignore it.

5.8 Conclusion

We presented the first automatic method able to use complex inner contours in the analysis and recursive decomposition of drawings that represent smooth shapes. Our decomposition method outputs a structure of closed 2D shapes layered in depth. It relies on the inference and progressive refinement of a partial depth tree to store depth information. A new metric computed along a skeleton was proposed to detect salient parts of complex drawings including internal silhouette curves. We introduced a new, perceptual-based criterion for selecting the most salient possible junctions, prioritizing them, and using them to recursively segment a shape into parts. An efficient implementation of curve closures using a variation of Scale-Invariant MVC functional was defined for closing the extracted sub-parts, hidden or salient. Lastly, we managed to keep most parameters scale-invariant, enabling us to achieve structural decomposition of drawings with different resolutions of features.

Many applications can benefit from our method. As we have illustrated, it enables organic sketches to be easily edited in a meaningful way. Subdivision and depth layering

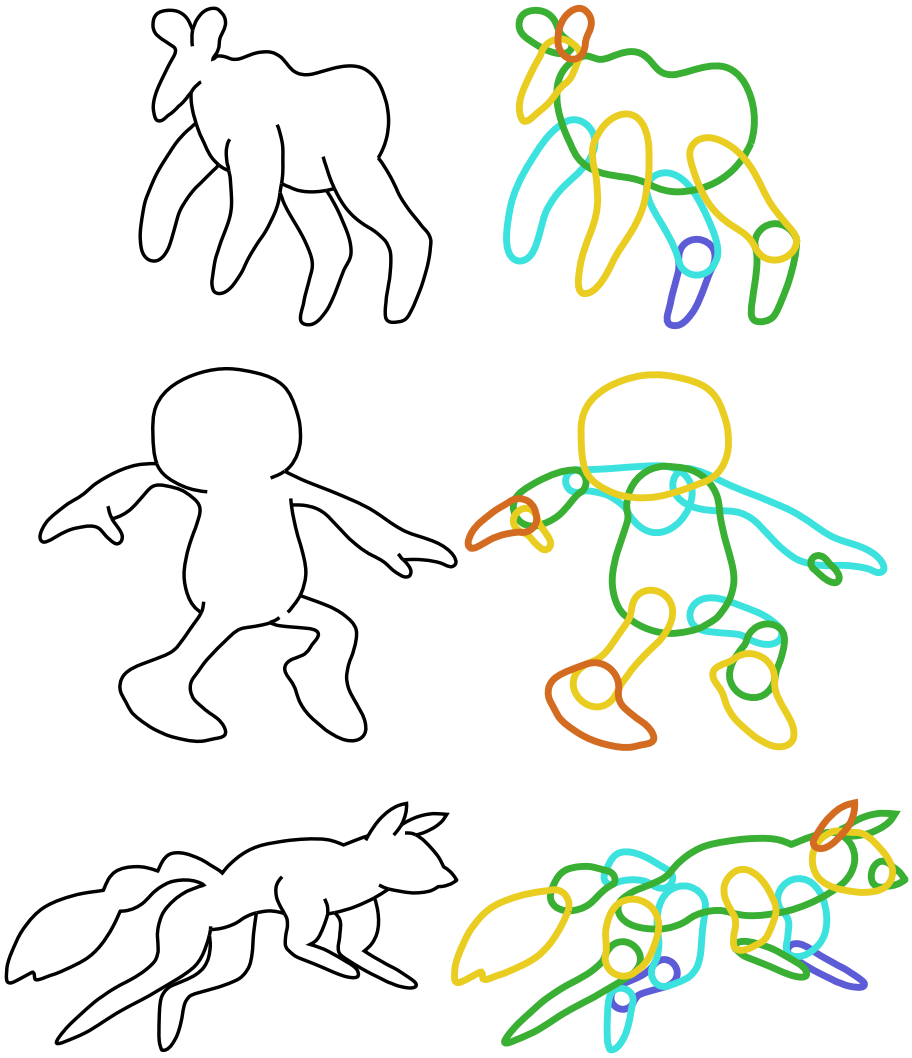


Figure 5.19 – Results on three drawings also used in [Bessmeltsev et al., 2015], except that we removed the hat of the character. Warmer colors are in the foreground.

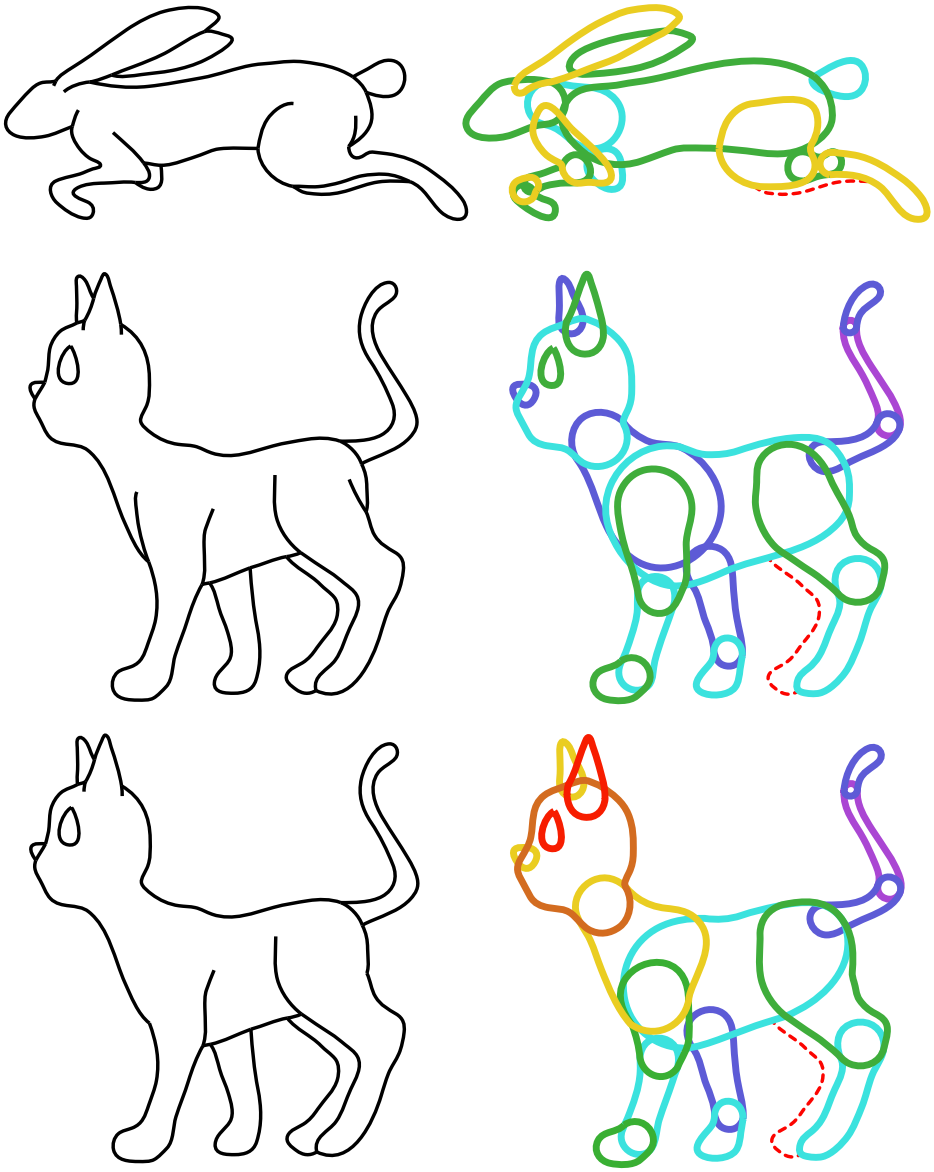


Figure 5.20 – Results for a rabbit and a cat. The drawing in the bottom is the middle drawing minus two suggestive contours, a case that our previous algorithm could not handle. Red dotted curves are contours that make our initialization stage fail.

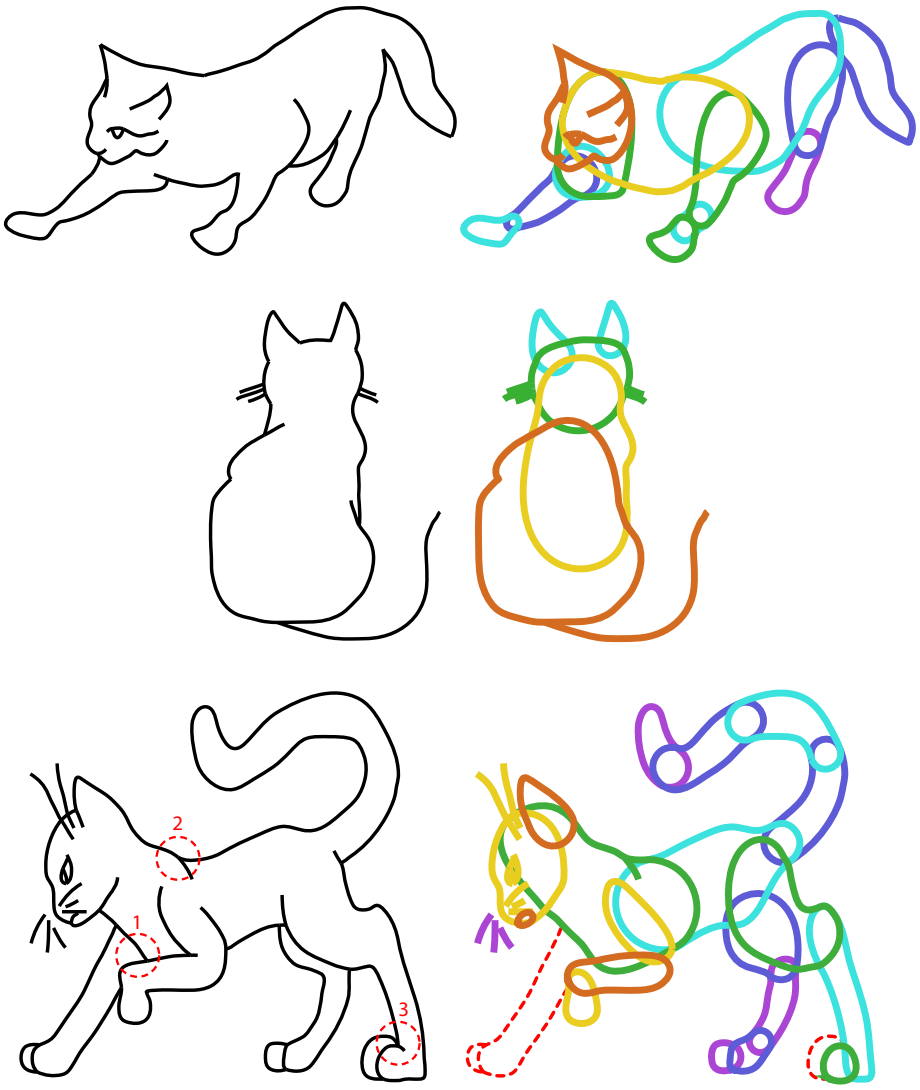


Figure 5.21 – Results for three drawings found on the first results page of Google Images with the following search terms: “cat line drawing”. The second result has been produced with salience threshold parameter that is more sensitive than the one used for all the other examples. The third example is subject to three limitations: 1) non trivial hidden part completion; 2) badly defined T-junction; 3) 4-valence vertex.

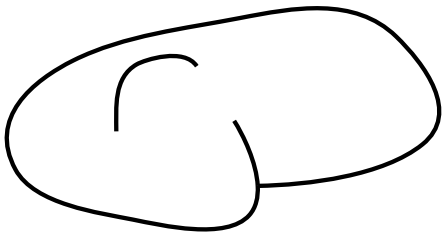


Figure 5.22 – An input where a suggestive contour is not connected to the outer contour and thus misclassified as as a decorative element in our initializaton step. We would like to use it as a possible segment of a foreground closure in future works.

makes the model ready for simple 2D animations. As future work, the automatic determination of the implied articulations between overlapping shape parts would make the application to vector drawing animation even more straightforward. We could also locally blend the contours at each animation step in order to enable smooth transitions between silhouette curves where and when needed, as done by the user in Figure 5.1 (c). This could be done with the help of 2D implicit contours and specially designed operators. We also plan to investigate 3D shape modeling from our part decomposition method, similarly to what was done in [Entem et al., 2015, Bessmeltsev et al., 2015] for much more constrained input. When applied to vector drawing animations, such a 3D intermediate representation would enable us to apply out-of-plane rotations to limbs and to change the viewpoint, two cases in which the silhouettes and occlusion between shape parts need to be recomputed.

CHAPTER 6

Conclusion

We do perceive a shape when we look at its depiction and it feels so natural that the processes involved are difficult to identify. Formalizing and identifying emerging principles of perception has been widely studied by psychologists in the past century and provided numerous insights and rules that are exploited in Computer Vision. However we are still far away from understanding and mimicking human perception. Therefore the Computer Graphics community also exploited other information or techniques such as prior knowledge and machine learning to create 3D models from 2D drawings. Nowadays a number of different sketch-based modeling techniques provide intuitive ways to produce 3D content. This thesis follows this direction.

Summary

Drawings often lacks some key elements to be explicit. The implicit part does not exist in reality but in our minds. Artists help on tackling this problem as they play with our perception and deformed shapes to determine the limits of recognizability. Artists also developed drawing techniques that replaced naive approaches with efficient ones. Since this thesis is about modeling shapes we focused on an intermediate drawing step that can be found in different drawing tutorials: placing the main volumes in space. This is done either with sharp-edged primitives to explicitly show the sagital planes or with simple silhouette contours of egg-shaped volumes. As the construction lines of sharp-edged primitives mostly disappear in the final drawings, we focused on the idea of simple primitives and how we could transpose this concept into the inverse problem: finding the structure of volumes in an already existing drawing. This leads to various scientific contributions in two publications which both leverage on this idea of constituent volumes or

structural sub-parts. We introduced the notion of *structural part completion* and proposed to study its application to two families of drawings. First we considered line drawings of animals and creatures for which we developed both a *structural analysis method for side-view sketches of animals* and a *3D model generation from structured drawings of animals* method. Then we considered free-form organic shapes and developed *a metric to identify salient structural joints* and an *aesthetic contour completion of structural parts* technique for the different configurations of visible contours in the vicinity of salient joins. Leveraging these two contributions, we finally proposed *an iterative pipeline for structural decomposition* of line drawings depicting free-form organic shapes.

6.1 Future Directions

As suggested in Sections 4.7 and 5.7, the two frameworks we developed can be extended in a number of ways for either practical or theorical purposes.

Improving the animal models with adapted blending operators

Looking at the results of our sketch-based modeling of animals, one would expect the interior part of joints between limbs and body to include contact regions. This limitation comes from our current naive composition of the individual implicit surfaces. We could use the gradient-based implicit blending proposed by [Gourmel et al., 2013], the implicit skinning by [Vaillant et al., 2013] or even the sketch-based implicit blending by [Angles et al., 2017] to improve the appearence of structural joints.

Sketch-based modeling of repetitive details on implicit surfaces

Although the use of implicit surfaces for the modeling of creatures is interesting in regard to animation purposes, such representation cannot be easily mapped to a texture. Thus the use of color texture, as well as displacement and bump maps for geometric details, remains too complex. However [Zanni et al., 2012] proposed a solution to model *Geometric Details on Skeleton-based Implicit Surfaces* and [de Groot et al., 2013] proposed a method for *Interactive Editing of Repetitive Patterns on Surfaces*. This could be leveraged in an efficient sketch-based tool for the modeling of repetitive details. Moreover it shows the advantage of behaving nicely in respect to the manipulation of the supporting skeleton: more detail features appear when the main shape is streched and details can blend with others at joints or simply fade away.

The general aspect is pleasing but the shape of individual details can vary wildly as shown in our early experiments in Figure 6.2(a,b,c). This was expected from a noise-based displacement function and is non trivially manageable even when manually tweaking the parameters. To tackle this problem we tried to propose different pre-defined ranges of

parameters for the different types of details that we wanted to create such as: spikes, bulges, holes, oriented creases. Some of them are shown in Figure 6.1.

As for the sketching interface, I did not find a simple solution considering the number of characteristics that are to be deduced: type, size, orientation, density. I first tried to project each stroke on the sagittal plane orthogonal to the tangent plane at the location of the stroke on the underlying surface when the angle to the viewer is big enough. This could give the correct profile of the detail to determine its height. However in practice this proved to not be intuitive since geometrically accurate perspective of free-form details is not something a standard user could achieve. It remains an interesting project whose promizing first results would justify further research 6.2(d).

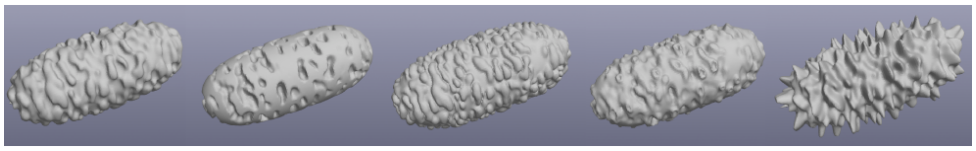


Figure 6.1 – Different kinds of procedurally generated details produced using [Zanni et al., 2012]'s technique.

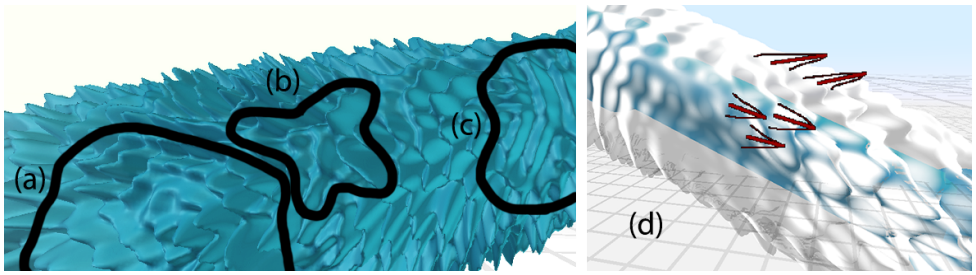


Figure 6.2 – (a,b,c) Areas on the surface where the shape of individual details is not as expected. (d) An example of sketched details and the resulting surface. The details are interpolated with another kind of details specified further along the shape on the left side.

Combining structural and figural completion

Sometimes suggestive contours do not imply a structural joint but a small local self-overlap. Thus combining both the figural completion proposed by [Karpenko and Hughes, 2006] and our structural decomposition technique could lead to a better final representation. Moreover, if a figural completion is desired, our method could be extended to infer an actual shape contour as illustrated in Figure 6.3.

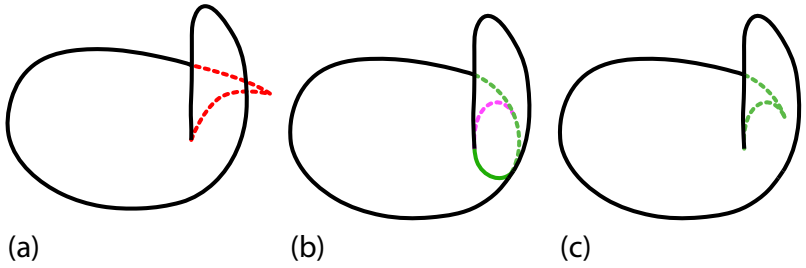


Figure 6.3 – (a) A failed figural completion using [Karpenko and Hughes, 2006]. (b) Our structural completion in green and the curve in pink that we would have to compute to obtain the figural completion in (c).

Repetitive-feature-aware structural completion

Considering the drawing in Figure 6.4(a) we see that current contour completion techniques would not be able to provide any solution in this case. In an effort to tackle this problem I came up with a partial solution that uses the Medial-Axes Transform to abstract a certain level of details (LOD) to obtain a result similar to Figure 6.4(b). It fixes the usual contour completion as the contours incident to T-junctions are now free of local features. However the repetitive nature of the features is not used in the completion and thus it is non trivial to integrate the computed hidden contour in the original drawing. As future works, I would first define a distance function between the original contour and the approximated contour to identify a recurring pattern. It might then be possible to deform the naively completed hidden contour to provide a smooth transition between the recurring patterns found at its endpoints.

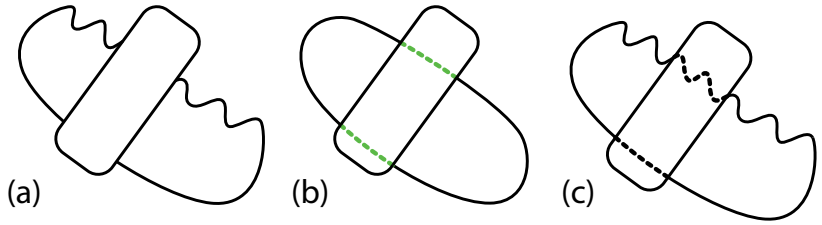


Figure 6.4 – **(a)** A drawing that remains untackled by figural completion methods. **(b)** A simpler drawing derived from (a) by approximating visible contours using a variant of the Medial-Axis Transform of [Giesen et al., 2009] to work on partial Medial-Axes. Figural completion can be computed with the current techniques, ours produce the hidden contours in green. **(c)** Ground truth.

The Medial-Axes used here are open ended and are computed for each shape unit using only its visible contours and a rule to cut the skeleton where Medial-Axis disks start to describe the outside of the shape. In practice I used an hysteresis on the angle formed at

the center of the disk between the two points of the contour that are tangent to the disk. This solution also suggest that we could try to directly interpolate a skeleton segment between partial Medial-Axes in the shape completion step. This would be equivalent to testing a pair of plausible hidden contours.

CHAPITRE 7

Conclusion (FR)

Nous percevons une forme lorsque nous regardons sa représentation 2D et cela est tellement naturel que les processus impliqués sont difficiles à identifier. Formaliser et identifier les principes de perception émergents a été le travail de différents psychologues au cours du siècle dernier et a fourni à la communauté de Vision par Ordinateur un certain nombre de pistes à suivre. Cependant, nous sommes encore loin de comprendre et de savoir imiter la perception humaine. Par conséquent, la communauté de Computer Graphics n'a pas attendu pour tirer parti de ces découvertes afin de créer des modèles 3D à partir de dessins 2D. De nos jours, une multitude de techniques de modélisation par esquisse permettent de produire des contenus 3D de manière intuitive. Cette thèse fait partie de ce mouvement.

Résumé

Les dessins manquent souvent de certains éléments clés pour être explicites. La partie implicite n'existe pas dans la réalité mais dans nos esprits. Les artistes ont aidé à résoudre ce problème en jouant avec notre perception et en essayant de déformer leurs représentations pour déterminer les limites du reconnaissable. Ils ont également développé des techniques de dessin qui ont remplacé les approches naïves par des méthodes efficaces. Puisque cette thèse concerne la modélisation des formes nous nous sommes concentrés sur une étape de dessin intermédiaire que l'on peut trouver dans différents tutoriels de dessin : placer les principaux volumes de construction dans l'espace. Ceci se fait soit avec des primitives à arêtes vives pour montrer explicitement les plans sagittaux, soit avec des contours de silhouette de volumes lisses. Comme les lignes de construction des primitives à arêtes vives disparaissent pour la plupart dans les dessins finaux, nous nous

sommes concentrés sur l'idée de primitives simples et sur la manière de transposer ce concept lors de l'étude du problème inverse : trouver la structure des volumes dans un dessin existant. Cela a conduit à diverses contributions scientifiques regroupées en deux publications, toutes deux mettant à profit cette idée de volumes constituants ou de sous-parties structurelles. Nous avons introduit la notion de *complétion de partie structurelle* et proposé d'étudier son application dans deux familles de dessins. Nous avons d'abord examiné des dessins aux traits d'animaux et de créatures pour lesquels nous avons développé à la fois une *méthode d'analyse structurelle de dessins d'animaux vus de côté* et une méthode de *génération de modèles 3D à partir de dessins structurés*. Nous avons ensuite considéré les formes organiques de forme libre et développé *une métrique pour identifier les jointures structurelles saillantes* ainsi qu'une technique de *complétion esthétique de contour des parties structurelles* pour les différentes configurations de contours visibles au voisinage des jointures saillantes. En tirant parti de ces deux contributions, nous avons finalement proposé *un pipeline itératif pour la décomposition structurelle* de dessins au trait représentant des formes organiques libres.

Pistes de recherche

Comme suggéré dans les Sections 4.7 et 5.7, les deux frameworks que nous avons développés peuvent être étendus de différentes manières pour des applications pratiques ou théoriques. Ainsi je propose plusieurs pistes de recherche (c.f. la version anglaise de la conclusion pour plus de détails) :

Améliorer les résultats de notre outil de modélisation d'animaux en utilisant le mélange implicite basé gradients proposé par [Gourmel et al., 2013].

Modélisation par esquisse de détails répétitifs . Nos modèles d'animaux pourraient être améliorés grâce la génération procédurale de détails sur surfaces implicites proposée par [Zanni et al., 2012]. Esquisser quelques détails pourrait suffir à définir les paramètres de cette génération procédurale.

Combiner complétion structurelle et complétion figurale afin d'éviter les limitations de chacune de ces approches.

Une complétion structurelle robuste à la présence de motifs répétitifs au sein des contours. Ceci pourrait se faire en utilisant le Medial-Axis Transform de [Giesen et al., 2009] sur les silhouettes partielles afin d'en abstraire les motifs.

Bibliography

[Alexe et al., 2007] Alexe, A., Barthe, L., Cani, M. P., and Gaildrat, V. (2007). Shape modeling by sketching using convolution surfaces. In *ACM SIGGRAPH 2007 courses on - SIGGRAPH '07*. ACM Press.

[Amer et al., 2014] Amer, M. R., Yousefi, S., Raich, R., and Todorovic, S. (2014). Monocular extraction of 2.1d sketch using constrained convex optimization. *International Journal of Computer Vision*, 112(1):23–42.

[Angles et al., 2017] Angles, B., Tarini, M., Wyvill, B., Barthe, L., and Tagliasacchi, A. (2017). Sketch-based implicit blending. *ACM Transactions on Graphics*, 36(6):1–13.

[Bernhardt et al., 2008] Bernhardt, A., Pihuit, A., Cani, M.-P., and Barthe, L. (2008). Matisse: Painting 2d regions for modeling free-form shapes.

[Bessmeltsev et al., 2015] Bessmeltsev, M., Chang, W., Vining, N., Sheffer, A., and Singh, K. (2015). Modeling character canvases from cartoon drawings. *ACM Transactions on Graphics*, 34(5):1–16.

[Blum, 1967] Blum, H. (1967). A Transformation for Extracting New Descriptors of Shape. In Wathen-Dunn, W., editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge.

[Blum and Nagel, 1978] Blum, H. and Nagel, R. N. (1978). Shape description using weighted symmetric axis features. *Pattern Recognition*, 10(3):167–180.

[Brady and Asada, 1984] Brady, M. and Asada, H. (1984). Smoothed local symmetries and their implementation. *The International Journal of Robotics Research*, 3(3):36–61.

[Bui et al., 2015] Bui, M. T., Kim, J., and Lee, Y. (2015). 3d-look shading from contours and hatching strokes. *Computers & Graphics*, 51:167–176.

[Carlier et al., 2016] Carlier, A., Leonard, K., Hahmann, S., Morin, G., and Collins, M. (2016). The 2d shape structure dataset: A user annotated open access database. *Computers & Graphics*, 58:23–30.

[Clowes, 1971] Clowes, M. B. (1971). On seeing things. *Artificial Intelligence*, 2(1):79–116.

[Cook and Agah, 2009] Cook, M. T. and Agah, A. (2009). A survey of sketch-based 3-d modeling techniques. *Interacting with Computers*, 21(3):201–211.

[Cordier et al., 2016] Cordier, F., Gingold, Y., Entem, E., Cani, M.-P., and Singh, K. (2016). Sketch-based Modeling. In *Eurographics 2016*, number T7, Lisbonne, Portugal. The Eurographics Association, The Eurographics Association.

[Cordier and Seo, 2007] Cordier, F. and Seo, H. (2007). Free-form sketching of self-occluding objects. *IEEE Computer Graphics and Applications*, 27(1):50–59.

[Cordier et al., 2011] Cordier, F., Seo, H., Park, J., and Noh, J. Y. (2011). Sketching of mirror-symmetric shapes. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1650–1662.

[Dalstein et al., 2014] Dalstein, B., Ronfard, R., and van de Panne, M. (2014). Vector graphics complexes. *ACM Transactions on Graphics*, 33(4):1–12.

[de Groot et al., 2013] de Groot, E., Wyvill, B., Barthe, L., Nasri, A., and Lalonde, P. (2013). Implicit decals: Interactive editing of repetitive patterns on surfaces. *Computer Graphics Forum*, 33(1):141–151.

[DeCarlo et al., 2003] DeCarlo, D., Finkelstein, A., Rusinkiewicz, S., and Santella, A. (2003). Suggestive contours for conveying shape. In *ACM SIGGRAPH 2003 Papers on - SIGGRAPH '03*. ACM Press.

[Douglas and Peucker, 2011] Douglas, D. H. and Peucker, T. K. (2011). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. In *Classics in Cartography*, pages 15–28. John Wiley & Sons, Ltd.

[Edwards, 2001] Edwards, B. (2001). *The New Drawing on the Right Side of the Brain*. HarperCollins Publishers Ltd.

[Eisemann et al., 2009] Eisemann, E., Paris, S., and Durand, F. (2009). A visibility algorithm for converting 3d meshes into editable 2d vector graphics. *ACM Transactions on Graphics*, 28(3):1.

[Eitz et al., 2012] Eitz, M., Hays, J., and Alexa, M. (2012). How do humans sketch objects? *ACM Transactions on Graphics*, 31(4):1–10.

[Entem et al., 2015] Entem, E., Barthe, L., Cani, M.-P., Cordier, F., and Van De Panne, M. (2015). Modeling 3D animals from a side-view sketch. *Computers and Graphics*, 46:221–230. Shape Modeling International 2014.

[Entem et al., 2016] Entem, E., Barthe, L., Cani, M.-P., and van de Panne, M. (2016). From drawing to animation-ready vector graphics. In *ACM SIGGRAPH 2016 Posters*, SIGGRAPH '16, pages 52:1–52:2, New York, NY, USA. ACM.

[Entem et al., 2018] Entem, E., Parakkat, A. D., Cani, M.-P., and Barthe, L. (2018). Structuring and layering contour drawings of organic shapes. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*, Expressive '18, pages 4:1–4:14, New York, NY, USA. ACM.

[Fonseca et al., 2009] Fonseca, M. J., Ferreira, A., and Jorge, J. A. (2009). Sketch-based retrieval of complex drawings using hierarchical topology and geometry. *Computer-Aided Design*, 41(12):1067–1081.

[Fortune, 1986] Fortune, S. (1986). A sweepline algorithm for voronoi diagrams. In *Proceedings of the second annual symposium on Computational geometry - SCG '86*. ACM Press.

[Geiger et al., 1998] Geiger, D., Pao, H., and Rubin, N. (1998). Salient and multiple illusory surfaces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '98, pages 118–, Washington, DC, USA. IEEE Computer Society.

[Giesen et al., 2009] Giesen, J., Miklos, B., Pauly, M., and Wormser, C. (2009). The scale axis transform. In *Proceedings of the 25th annual symposium on Computational geometry - SCG '09*. ACM Press.

[Gingold et al., 2009] Gingold, Y., Igarashi, T., and Zorin, D. (2009). Structured annotations for 2D-to-3D modeling. In *ACM SIGGRAPH Asia 2009 papers on - SIGGRAPH Asia '09*. ACM Press.

[Gourmel et al., 2013] Gourmel, O., Barthe, L., Cani, M.-P., Wyvill, B., Bernhardt, A., Paulin, M., and Grasberger, H. (2013). A gradient-based implicit blend. *ACM Transactions on Graphics*, 32(2):1–12.

[Guzmán, 1968] Guzmán, A. (1968). Decomposition of a visual scene into three-dimensional bodies. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I on - AFIPS '68 (Fall, part I)*. ACM Press.

[Hoffman, 1998] Hoffman, D. D. (1998). *Visual Intelligence, How We Create What We See*. Norton and Company Ltd.

[Horn, 1983] Horn, B. K. P. (1983). The curve of least energy. *ACM Transactions on Mathematical Software*, 9(4):441–460.

[Huffman, 1971] Huffman, D. A. (1971). Impossible Objects as Nonsense Sentences. *Machine Intelligence*, 6:295–323.

[Igarashi et al., 1999] Igarashi, T., Matsuoka, S., and Tanaka, H. (1999). Teddy: A sketching interface for 3D freeform design. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99*. ACM Press.

[Igarashi and Mitani, 2010] Igarashi, T. and Mitani, J. (2010). Apparent layer operations for the manipulation of deformable objects. *ACM Transactions on Graphics*, 29(4):1.

[Ijiri et al., 2006] Ijiri, T., Owada, S., and Igarashi, T. (2006). Seamless integration of initial sketching and subsequent detail editing in flower modeling. *Computer Graphics Forum*, 25(3):617–624.

[Jacobs, 1996] Jacobs, D. W. (1996). Robust and efficient detection of salient convex groups. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):23–37.

[Kanade, 1980] Kanade, T. (1980). A theory of Origami world. *Artificial Intelligence*, 13(3):279–311.

[Kanizsa, 1979] Kanizsa, G. (1979). *Organization in Vision*. Praeger, New York.

[Karpenko and Hughes, 2006] Karpenko, O. A. and Hughes, J. F. (2006). SmoothSketch: 3D Free-form Shapes from Complex Sketches. *ACM Transactions on Graphics*, 25(3):589.

[Koffka, 1935] Koffka, K. (1935). *Principles of Gestalt psychology*. Harcourt, Brace & World.

[Kraevoy et al., 2009] Kraevoy, V., Sheffer, A., and van de Panne, M. (2009). Modeling from contour drawings. In *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling - SBIM '09*. ACM Press.

[Kunsberg and Zucker, 2018] Kunsberg, B. and Zucker, S. W. (2018). Critical contours: An invariant linking image flow with salient surface organization. *SIAM Journal on Imaging Sciences*, 11(3):1849–1877.

[Larsson et al., 2015] Larsson, L. J., Morin, G., Begault, A., Chaine, R., Abiva, J., Hubert, E., Hurdal, M., Li, M., Paniagua, B., Tran, G., and Cani, M.-P. (2015). Identifying perceptually salient features on 2d shapes. In *Association for Women in Mathematics Series*, pages 129–153. Springer International Publishing.

[Latecki and Lakämper, 1999] Latecki, L. J. and Lakämper, R. (1999). Convexity rule for shape decomposition based on discrete contour evolution. *Computer Vision and Image Understanding*, 73(3):441–454.

[Leonard et al., 2016] Leonard, K., Morin, G., Hahmann, S., and Carlier, A. (2016). A 2d shape structure for decomposition and part similarity. In *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE.

[Lesser et al., 1995] Lesser, V. R., Nawab, S. H., and Klassner, F. I. (1995). IPUS: an architecture for the integrated processing and understanding of signals. *Artificial Intelligence*, 77(1):129–171.

[Liu et al., 2014] Liu, G., Xi, Z., and Lien, J.-M. (2014). Dual-space decomposition of 2d complex shapes. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

[Liu et al., 2009] Liu, R., Zhang, H., Shamir, A., and Cohen-Or, D. (2009). A part-aware surface metric for shape analysis. *Computer Graphics Forum*, 28(2):397–406.

[Liu et al., 2015] Liu, X., Wong, T.-T., and Heng, P.-A. (2015). Closure-aware sketch simplification. *ACM Transactions on Graphics*, 34(6):1–10.

[Lun et al., 2017] Lun, Z., Zou, C., Huang, H., Kalogerakis, E., Tan, P., Cani, M.-P., and Zhang, H. (2017). Learning to group discrete graphical patterns. *ACM Trans. Graph.*, 36(6):225:1–225:11.

[Macworth, 1973] Macworth, A. K. (1973). Interpreting pictures of polyhedral scenes. *Artificial Intelligence*, 4(2):121–137.

[McCann and Pollard, 2009] McCann, J. and Pollard, N. (2009). Local layering. *ACM Transactions on Graphics*, 28(3):1.

[Mi et al., 2009] Mi, X., DeCarlo, D., and Stone, M. (2009). Abstraction of 2d shapes in terms of parts. In *Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering - NPAR '09*. ACM Press.

[Mitra et al., 2006] Mitra, N. J., Guibas, L. J., and Pauly, M. (2006). Partial and approximate symmetry detection for 3d geometry. In *ACM SIGGRAPH 2006 Papers on - SIGGRAPH '06*. ACM Press.

[Moreton, 1992] Moreton, H. (1992). *Minimum Curvature Variation Curves, Networks, and Surfaces for Fair Free-form Shape Design*. PhD thesis, University of California at Berkeley, Berkeley, CA, USA. UMI Order No. GAX93-30652.

[Nitzberg and Mumford, 1990] Nitzberg, M. and Mumford, D. (1990). The 2.1-D sketch. In *Proceedings Third International Conference on Computer Vision*. IEEE Comput. Soc. Press.

[Noris et al., 2013] Noris, G., Hornung, A., Sumner, R. W., Simmons, M., and Gross, M. (2013). Topology-driven vectorization of clean line drawings. *ACM Transactions on Graphics*, 32(1):1–11.

[Noris et al., 2012] Noris, G., Sýkora, D., Shamir, A., Coros, S., Whited, B., Simmons, M., Hornung, A., Gross, M., and Sumner, R. (2012). Smart scribbles for sketch segmentation. *Computer Graphics Forum*, 31(8):2516–2527.

[Olsen et al., 2009] Olsen, L., Samavati, F. F., Sousa, M. C., and Jorge, J. A. (2009). Sketch-based modeling: A survey. *Computers & Graphics*, 33(1):85–103.

[Pessoa and Weerd, 2003] Pessoa, L. and Weerd, P. D. (2003). *Filling-In: From Perceptual Completion to Cortical Reorganization*. Oxford University Press.

[Pihuit et al., 2010] Pihuit, A., Cani, M.-P., and Palombi, O. (2010). Sketch-based modeling of vascular systems: a first step towards interactive teaching of anatomy. *Eurographics Workshop on Sketch-Based Interfaces and Modeling*.

[Richards et al., 1987] Richards, W. A., Koenderink, J. J., and Hoffman, D. D. (1987). Inferring three-dimensional shapes from two-dimensional silhouettes. *Journal of the Optical Society of America A*, 4(7):1168.

[Rivers et al., 2010a] Rivers, A., Durand, F., and Igarashi, T. (2010a). 3d modeling with silhouettes. *ACM Transactions on Graphics*, 29(4):1.

[Rivers et al., 2010b] Rivers, A., Igarashi, T., and Durand, F. (2010b). 2.5d cartoon models. *ACM Transactions on Graphics*, 29(4):1.

[Schmidt et al., 2009] Schmidt, R., Khan, A., Singh, K., and Kurtenbach, G. (2009). Analytic drawing of 3d scaffolds. *ACM Transactions on Graphics*, 28(5):1.

[Schmidt et al., 2007] Schmidt, R., Wyvill, B., Sousa, M. C., and Jorge, J. A. (2007). Shapeshop: Sketch-based solid modeling with blobtrees. In *ACM SIGGRAPH 2007 courses on - SIGGRAPH '07*. ACM Press.

[Shamir, 2008] Shamir, A. (2008). A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539–1556.

[Sharon et al., 2000] Sharon, E., Brandt, A., and Basri, R. (2000). Completion energies and scale. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1117–1131.

[Singh, 2014] Singh, M. (2014). *Visual Representation of Contour and Shape*. Oxford University Press.

[Sun et al., 2012] Sun, Z., Wang, C., Zhang, L., and Zhang, L. (2012). Free hand-drawn sketch segmentation. In *Computer Vision – ECCV 2012*, pages 626–639. Springer Berlin Heidelberg.

[Sýkora et al., 2014] Sýkora, D., Kavan, L., Čadík, M., Jamriška, O., Jacobson, A., Whited, B., Simmons, M., and Sorkine-Hornung, O. (2014). Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters. *ACM Transactions on Graphics*, 33(2):1–15.

[Sýkora et al., 2010] Sýkora, D., Sedlacek, D., Jinchao, S., Dingliana, J., and Collins, S. (2010). Adding depth to cartoons using sparse depth (in)equalities. *Computer Graphics Forum*, 29(2):615–623.

[Tu and Zhu, 2002] Tu, Z. and Zhu, S.-C. (2002). Parsing images into region and curve processes. In *Computer Vision — ECCV 2002*, pages 393–407. Springer Berlin Heidelberg.

[Turmukhambetov et al., 2015] Turmukhambetov, D., Campbell, N. D., Goldman, D. B., and Kautz, J. (2015). Interactive sketch-driven image synthesis. *Computer Graphics Forum*, 34(8):130–142.

[Turquin et al., 2007] Turquin, E., Wither, J., Boissieux, L., paule Cani, M., and Hughes, J. (2007). A sketch-based interface for clothing virtual characters. *IEEE Computer Graphics and Applications*, 27(1):72–81.

[Ullman, 1976] Ullman, S. (1976). Filling-in the gaps: The shape of subjective contours and a model for their generation. *Biol. Cybern.*, 25(1):1–6.

[Vaillant et al., 2013] Vaillant, R., Barthe, L., Guennebaud, G., Cani, M.-P., Rohmer, D., Wyvill, B., Gourmel, O., and Paulin, M. (2013). Implicit skinning. *ACM Transactions on Graphics*, 32(4):1.

[Voronoi, 1908] Voronoi, G. (1908). Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les paralléllοèdres primitifs. *Journal für die reine und angewandte Mathematik*, 134:198–287.

[Wiley and Williams, 2007] Wiley, K. and Williams, L. R. (2007). Representing interwoven surfaces in 2-1/2d drawings. *IEEE Computer Graphics and Applications*, 27(4):70–83.

[Willats, 1997] Willats, J. (1997). *Art and Representation: New Principles in the Analysis of Pictures*. Princeton University Press.

[Williams, 1997] Williams, L. R. (1997). Topological Reconstruction of a Smooth Manifold-Solid from Its Occluding Contour. *International Journal of Computer Vision*, 23(1):93–108.

[Williams and Hanson, 1996] Williams, L. R. and Hanson, A. R. (1996). Perceptual Completion of Occluded Surfaces. *Computer Vision and Image Understanding*, 64(1):1–20.

[Wither et al., 2007] Wither, J., Bertails, F., and Cani, M.-P. (2007). Realistic hair from a sketch. In *IEEE International Conference on Shape Modeling and Applications 2007 (SMI '07)*. IEEE.

[Wither et al., 2009] Wither, J., Boudon, F., Cani, M.-P., and Godin, C. (2009). Structure from silhouettes: a new paradigm for fast sketch-based design of trees. *Computer Graphics Forum*, 28(2):541–550.

[Witkin and Tenenbaum, 1983] Witkin, A. P. and Tenenbaum, J. M. (1983). On the role of structure in vision. In *Human and Machine Vision*, pages 481–543. Elsevier.

[Xu et al., 2014] Xu, B., Chang, W., Sheffer, A., Bousseau, A., McCrae, J., and Singh, K. (2014). True2form: 3D curve networks from 2D sketches via selective regularization. *ACM Transactions on Graphics*, 33(4):1–13.

[Yang et al., 2008] Yang, M., Kpalma, K., and Ronsin, J. (2008). A Survey of Shape Feature Extraction Techniques. In Yin, P.-Y., editor, *Pattern Recognition*, pages 43–90. IN-TECH. 38 pages.

[Yeh et al., 2017] Yeh, C.-K., Huang, S.-Y., Jayaraman, P. K., Fu, C.-W., and Lee, T.-Y. (2017). Interactive high-relief reconstruction for organic and double-sided objects from a photo. *IEEE Transactions on Visualization and Computer Graphics*, 23(7):1796–1808.

[Yeh et al., 2015] Yeh, C.-K., Jayaraman, P. K., Liu, X., Fu, C.-W., and Lee, T.-Y. (2015). 2.5d cartoon hair modeling and manipulation. *IEEE Transactions on Visualization and Computer Graphics*, 21(3):304–314.

[Yu et al., 2014] Yu, C.-C., Liu, Y.-J., Wu, M. T., Li, K.-Y., and Fu, X. (2014). A global energy optimization framework for 2.1d sketch extraction from monocular images. *Graphical Models*, 76(5):507–521.

[Zanni et al., 2012] Zanni, C., Bares, P., Lagae, A., Quiblier, M., and Cani, M.-P. (2012). Geometric Details on Skeleton-based Implicit Surfaces. In Andujar, C. and Puppo, E., editors, *Eurographics 2012 - Short Papers*. The Eurographics Association.

[Zanni et al., 2013] Zanni, C., Bernhardt, A., Quiblier, M., and Cani, M.-P. (2013). SCALe-invariant integral surfaces. *Computer Graphics Forum*, 32(8):219–232.

[Zeng et al., 2008] Zeng, J., Lakaemper, R., Yang, X., and Li, X. (2008). 2d shape decomposition based on combined skeleton-boundary features. In *Advances in Visual Computing*, pages 682–691. Springer Berlin Heidelberg.

[Zhu et al., 2006] Zhu, W., Chan, T., and Esedolu, S. (2006). Segmentation with depth: A level set approach. *SIAM Journal on Scientific Computing*, 28(5):1957–1973.