Example-based Video Editing

Oriel Frigo

UNIVERSITÉ PARIS DESCARTES

École doctorale 386 : Sciences Mathématiques de Paris Centre

Laboratoire MAP5 UMR CNRS 8145

Example-based Video Editing

Par Oriel Frigo

Thèse de doctorat de Mathématiques Appliquées

Soutenue publiquement le 19 octobre 2016

Devant le jury composé de :

Julie Delon	Université Paris Descartes	Directrice de thèse
George Drettakis	INRIA Sophia-Antipolis	Examinateur
Yann Gousseau	Télécom ParisTech	Examinateur
Pierre Hellier	Technicolor R&I	Directeur de thèse
Nikolas Papadakis	Institut de Mathématiques de Bordeaux	Rapporteur
Edoardo Provenzi	Université Paris Descartes	Examinateur
Neus Sabater	Technicolor R&I	Directrice de thèse
Josef Sivic	INRIA Paris	Rapporteur

Acknowledgments

First of all, I must say that the accomplishment of this thesis would not be possible without the precious guidance of my advisors: Julie Delon, Neus Sabater and Pierre Hellier. They taught me, each of them in different ways, the rigour in research, the ethics of science, and the joy of good mood. I feel very lucky that I was surrounded by them for these three years.

I am very grateful to the reviewers of this manuscript, Nicolas Papadakis and Josef Sivic, for their careful analysis of the document and their rich observations and suggestions. Also I would like to thank the jury members George Drettakis, Edoardo Provenzi and Yann Gousseau for their thoughtful questions and for providing a pleasant discussion during the defense of this thesis.

I should say I enjoyed a great time during my PhD with all my colleagues at University Paris Descartes and Technicolor. I was delighted with the nice atmosphere in the laboratory MAP5 at University Paris Descartes, and it was really a pleasure to work with the image processing team: Alasdair, Anne-Sophie, Artur, Bruno, Edoardo, Joan, Julie, Nora, Pierre and Remi. Thanks for the good memories of having lunch or coffee with you at "le terrasse de MAP5", with panoramic view to the beautiful Paris. I am particularly thankful to Nora for introducing me to some secrets of Inkscape, and for helping me creating nice diagrams for my presentation. Julie and Remi, I am glad for sharing with you my enthusiasm about Emacs, and thanks for introducing me the revolutionary Orgmode.

People at Technicolor were essential for me during these three years, as I stayed most of my time there. In particular, Dmitry, I was very glad to have you as the official company for the summer festivals, and thanks for not letting me starve by giving me some food when I was finishing to write this manuscript. Juan, I had the immense luck to have you as my office neighbor, and also to travel with you to attend two international conferences together (I hope you have finally recovered from your foot injury). Thanks to both of you for the happy moments together and for our lively discussions. Marc and Robin, thanks for teaching me a lot about code optimization and Star Wars. Other members of babyfoot and pingpong team: Jean, Matthieu, Charlotte, Martin, Wei, Fatma.. thanks for sharing with me the adrenaline of practicing these radical sports. Praveen, Saurabh and Himalaya, thanks for the nice vegetarian lunches together and for letting me taste some of the best indian food.

Finally, I want to thank my close friends in Brazil and my family: my mother Marli for her constant care and love, my father Paulo for the inspiration and support in my quest, and Bruna for being a lovely and supportive company for many years.

Contents

Chapter 1

Résumé en Français

L'objectif de cette thèse est d'étudier et de proposer de nouvelles approches pour l'édition d'images, et plus généralement de vidéos, en tenant compte de caractéristiques telles que la couleur ou la texture. L'approche basée exemple pour l'édition de vidéos est particulièrement adaptée aux activités de Technicolor dans la post-production de films et dans la mise en valeur des médias. Ce manuscrit de thèse présente trois problèmes connexes dans ce cadre : le *transfert de couleur*, la *stabilisation tonale* et le *transfert de style*. Dans ce qui suit, nous discutons brièvement chacun de ces problèmes et les résultats obtenus.

Le **transfert de couleur** est le processus consistant à modifier la distribution de couleur d'une image d'entrée pour qu'elle corresponde à la palette de couleurs d'une image d'exemple. Le transfert de couleur pour les images et vidéos est utile pour une grande variété d'applications, telles que l'homogénéisation et l'amélioration de couleurs ou l'étalonnage en post-production cinématographique.

Nous présentons dans le manuscrit une nouvelle méthode pour résoudre ce problème. Cette méthode se base sur des outils de transport optimal, et la transformation obtenue est régularisée par une interpolation dans l'espace des couleurs. Ce transport régularisé ne crée pas de défauts dans l'image transformée, ce qui est un net avantage par rapport aux méthodes de transfert de couleur de la littérature.

La **stabilisation tonale de vidéos** consiste à corriger l'instabilité tonale, défaut temporel caractérisé par des fluctuations artificielles des couleurs d'une vidéo. Ces instabilités sont principalement causées par un mauvais fonctionnement des réglages automatiques des caméras vidéos, comme la balance des blancs ou l'exposition automatique.

L'algorithme état de l'art pour la stabilisation tonale de vidéos présente une complexité de calcul élevée et d'un manque de robustesse par rapport au mouvement entre les trames. Nous proposons donc une méthode pour la stabilisation tonale informatiquement simple et utilisant le mouvement dominant entre les images de la séquence. Dans ce travail, une attention particulière est apportée à l'établissement d'un modèle approprié pour ces instabilités tonales. Étant donné que le modèle exact de réponse de couleur de la caméra est inconnu et non trivial, un modèle empirique est proposé. Après plusieurs expériences avec des modèles de transformation de couleur paramétriques et non paramétriques, une transformation de loi de puissance a été trouvée pour résoudre efficacement le problème de la stabilisation tonale. Un avantage du modèle paramétrique proposé est la possibilité d'obtenir ses coefficients

de manière exacte par régression, avec le potentiel pour le traitement de vidéos en temps réel. Enfin, la transformation tonale proposée est temporellement pondérée en fonction de l'amplitude de mouvement dans la vidéo, pour éviter l'écrêtage des couleurs et la génération de défauts.

Le **transfert de style** consiste à transformer une image de telle manière qu'elle imite le style d'une autre image d'exemple. Ce problème est typique du domaine du rendu d'image non-photoréaliste, où l'on peut souhaiter par exemple donner à une photographie le style d'un tableau d'un peintre donné. On montre dans le manuscrit que le problème du transfert de style peut être approché par un transfert de textures par "patches" combiné avec un transfert global de couleur. Nous proposons une décomposition "Split et match", où le critère d'arrêt pour le fractionnement d'un "quadtree" dépend à la fois de la variance des patches et de la similitude entre les patches de l'image d'entrée et d'exemple.

Le problème du transfert de style est ensuite modélisé comme la recherche de la labellisation optimale de tous les patches sur un champ de Markov. Une des innovations de notre approche est de tirer parti d'une partition adaptative pour un transfert de style qui prend en compte des textures à plusieurs échelles. En outre, on a montré que cette technique peut être adapté pour le transfert de style de vidéos à partir d'une propagation de style guidé par le champ de mouvement de la séquence.

Ces travaux de thèse ont donné lieu à plusieurs brevets et publications. Les travaux sur le transfert de couleur ont donné lieu à deux brevets et ont été publiés dans les actes de la conférence internationale ACCV 2014. Le travail en stabilisation tonale de vidéo a conduit à une demande de brevet, un article publié à la conférence nationale GRETSI 2015, un article publié à la conférence internationale ICIP 2015, et un manuscrit accepté pour publication dans la revue "IEEE Transactions in Image Processing". En outre, l'algorithme est actuellement en phase d'optimisation pour être utilisé dans les puces Qualcomm. Finalement, le travail sur le transfert de style a jusqu'à présent mené à une demande de brevet et a été publié dans les actes de la conférence internationale CVPR 2016.

Chapter 2

Introduction

> *An example picture is worth a thousand words.*
>
> Adapted from English proverb, *Unknown Author*

Video editing is a crucial step of the film making process. It takes place after the stage of shooting and it consists in altering and enhancing the sequence of images from which the video is made of, in order to achieve a finished aspect, possibly with an artistic intent. This procedure has been traditionally associated with professional film post-production, where color grading, sound editing or the inclusion of special effects are performed in the video. With the diffusion of accessible consumer cameras and the success of photo and video sharing services in the Internet, video editing currently reached a larger audience of non-professional users. Therefore, the conception of new approaches to edit video content has a significant impact for both professional and consumer applications in the multimedia industry.

In this thesis, we approach different techniques to edit digital videos with the help of examples. Examples can be seen as a powerful resource for learning in general, and they can be naturally employed to estimate image transformations in the context of *Image Processing*, *Computer Vision* and *Computer Graphics*. The example-based approach for video editing is particularly well suited for applications in the context of film post-production and media enhancement. For example, a user wishing to modify a video or a picture could simply provide an example of what he wants, instead of manually adjusting filter parameters or choosing from a list of predefined transformations.

Throughout this manuscript, we are generally interested in learning an image transformation T that makes an input image u look like an example image v in terms of some desired characteristics. In particular, we are interested in transferring two main characteristics from example images: color and texture features.

When dealing with example-based color and texture video editing, there are three main questions that need to be addressed. The first, is how to estimate a transformation T that is both accurate enough to reproduce the characteristics from the example, and computationally simple enough to be adopted in practice for video editing applications. The second question, is how to guarantee that the transformation T preserves most of the structures of the original image u and has minimal visible artifacts in $T(u)$.

Finally, the third question is how to generalize the transformation T from a single image u to a video in order to guarantee temporal coherence. Certainly, temporal stability of color and texture is essential for the perceived quality of edited videos, as flickering or jitter can be very unpleasant for a video observer.

This manuscript presents efficient techniques to address the questions raised above and to properly solve three related problems that have promising applications in the multimedia industry: *Color Transfer*, *Tonal Stabilization*, and *Style Transfer*.

Color transfer and stabilization are two important issues in color video processing, which we deal with in the first chapters of this thesis. We suggest that these problems can be efficiently solved with smooth color transformations derived from example images.

In the last part of this manuscript, we approach the problem of style editing, where we are interested in transferring the style from an example image. We show that style can be seen as a combination of color and texture and we present an efficient technique for example-based non-photo-realistic rendering (NPR) that produces convincing stylization of images and videos.

In the following, we discuss briefly each one of these research problems and its achieved results and applications.

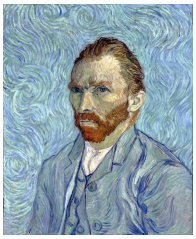
2.1 Color Transfer

In the movie industry, to ensure that a movie has the appropriate color characteristics, color grading, which consists of changing the color, the contrast, or the white balance and hues, is usually performed manually by a colorist. Certainly, color grading is a creative endeavor relying on the experience of the colorist. However, this professional could clearly benefit from automatic and consistent color transfer methods, as a completely manual color grading can be often a tedious and time consuming task.

Color transfer is the process to modify the color of an input image according to the color palette of an example image. Image and video color transfer is useful for a variety of applications, such as color homogenization, color enhancement and color grading in film post-production. In Chapter 4, we present the procedure of color transfer and a novel method to solve this problem. The color transfer algorithm was formulated in two steps: first, an example-based Chromatic Adaptation Transform (CAT) has been designed to obtain a linear illuminant matching between the input and example images. Secondly, the dominant colors from the input and example images are optimally matched by a non-linear chromaticity mapping. The efficiency of the method comes from computing a color distribution mapping through optimal transportation, and regularizing the mapping by fitting a thin-plate splines interpolation. This regularized chromaticity mapping does not create artifacts in the transformed image, which is a clear advantage in comparison to previous color transfer methods.

This project has resulted in two patents and a publication in the international

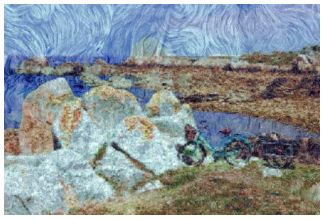
Example color



Example style

Original image

Color transfer result



Style transfer result

Figure 2.1: Illustration of color transfer and style transfer. From an original image, a user may transform it to have the color of a vintage autochrome picture, or the style of a Van Gogh painting.

conference ACCV 2014 [Frigo 2014]. In addition, this color transfer algorithm has been praised by the post-production team at Technicolor Creative Services who tested it in the generation of color grading Look-Up-Tables (LUT) with the aim of mimicking a vintage color look for films inspired by autochrome photographs. An illustration of such an application is shown in Figure 2.1.

Furthermore, our color transfer algorithm is being integrated as a plugin for the film editing software AutoDesk Lustre, and it is currently being used by colorists to assist the post-production of films. Some example of video results of our color transfer method can be found at the project website: `http://oriel.github.io/color_transfer.html`.

2.2 Video Tonal Stabilization

Closely related to color transfer, video tonal stabilization is the problem of properly correcting tonal instability, which is a particular temporal artifact characterized by fluctuations in the colors of adjacent frames of a video. As expected, color transfer for videos may not suffer with the problem of tonal instabilities if we simply use a single color transformation for a video shot. However, tonal instabilities in modern videos are mainly caused by a malfunction of automatic settings of consumer cameras, notably automatic white balance and automatic exposure.

We show in Chapter 5 that the baseline algorithm for video tonal stabilization has the disadvantage of high computational complexity and the lack of robustness with respect to motion between frames. Hence, our established goal is to propose a

method which is computationally simpler and takes advantage of dominant motion between frames. An illustration of the problem and results by our method are shown in Figure 2.2.

Figure 2.2: Illustration of tonal stabilization in sequence "entrance". **First row**: Frames extracted from the original sequence. **Second row**: Tonal stabilization with our proposed method.

A special attention is devoted to derive an appropriate model to compensate tonal instabilities observed in videos. Since the exact camera color response model is unknown and not trivial to estimate, an empirical rather than a theoretical model is proposed. After several experiments with parametric and non-parametric color transformation models, a simple and effective power law color transformation was found to solve effectively the problem of video tonal stabilization. An advantage of the proposed parametric model is the possibility to obtain its coefficients with closed form solution through linear least squares regression in logarithmic domain, which can be computed fastly and exactly, with potential for real-time video processing.

Note that our approach for tonal stabilization has an example-based inspiration: the keyframes are used as examples from which the color correction of a frame is performed. As a measure to avoid color clipping and artifact generation, the proposed tonal transformation is temporally weighted in function of the motion magnitude in the video.

This work in Video Tonal Stabilization has led to a patent application, a paper published at the national conference GRETSI 2015 [Frigo 2015b], a paper published at the international conference ICIP 2015 [Frigo 2015a], and a paper accepted to IEEE Transactions in Image Processing. In addition, the algorithm is being currently adapted to an optimized implementation to be used in Qualcomm smartphones. Video results of our tonal stabilization method can be found at the project website: `http://oriel.github.io/tonal_stabilization.html`.

2.3 Example-based Style Transfer

In the second part of the thesis, we change our feature of interest from color to style. Example-based Style Transfer can be seen as transforming an image in such a way that it mimics the style of a given example. A typical example is in the context of non-photorealistic rendering: to render a picture to make it look like it has the style of a painting; as we illustrate in Figure 2.1. The difficulty of this task is bound to the complexity of defining the style as a composition of different visual attributes such as color, shading, texture, lines, strokes and regions.

It is shown in Chapter 6 that the problem of style transfer can be approached by computing a patch-based transfer of texture and a global transfer of color. It is evidenced that patch dimensionality is crucial for the effectiveness of example-based texture transfer, as patch dimensions should be large enough to represent the patterns that characterize the example style, but small enough to forbid the synthesis of structures present in the example image.

We consider that a robust method for local texture transfer should capture the style of the example while preserving the structures of the source, and that this can be achieved with a spatially adaptive partition of patches, where patch sizes are adapted to image structures. We propose a Split and Match example-guided decomposition, where the stopping criteria for a quadtree splitting depends both on the patch variance and on the patch similarity between the input and example images.

The problem of texture transfer is modeled as searching for the optimal labeling configuration for all quadtree patches over a Markov Random Field (MRF). Then, the Loopy Belief Propagation algorithm is applied to compute an approximate optimal labeling. Usually, patch-based MRF models are computed over a graph in a regular grid, and one of our innovations is to propose a MRF model over an adaptive partition for a style transfer that takes into account multiple texture scales.

This work on style transfer has led to a patent application, and it was published in the international conference CVPR 2016 [Frigo 2016]. Some videos processed by our style transfer technique can be found at the project website: `http://oriel.github.io/video_style_transfer.html`.

2.4 Context of the Thesis

This thesis was prepared in the context of an industrial PhD agreement (*ANRT CIFRE*) between Technicolor R&I Rennes[1] and Laboratory MAP5 at Paris Descartes University. Technicolor is an industry leader in the science of film coloring and post-production and has been around for over 100 years.

Within the Image Science Laboratory at Technicolor, the present thesis was conducted in a research group focused on Example-guided video modification. Providing an example image instead of a list of predefined filters or Look Up Tables

[1] https://research.technicolor.com/rennes/

(LUT) is very interesting for Technicolor's business. For instance, colorists trying to reproduce a specific "look and feel" may find in an example image what they search for. In this sense, example-based editing can be used to guide post-production services. Technicolor is also involved in the development of color correction software, such as the CineStyle[2].

We believe that the close connection between research and post-production film services at Technicolor has a clear advantage for the preparation of this thesis. Fruitful collaborations between researchers and film professionals allow to understand technical necessities in post-production, which can eventually give birth to research problems.

2.5 Thesis Organization

This thesis is organized into the following structure:

- In Chapter 3 we provide a background on fundamental concepts of computational color, color constancy, color spaces and radiometric calibration.
- In Chapter 4 we present our approach for color transfer as an optimal transportation between color distributions.
- In Chapter 5 we discuss the problem of tonal instabilities found in modern videos, and we propose a motion driven technique for color stabilization.
- In Chapter 6 we introduce the problem of style transfer and we propose an example-based technique for image and video stylization that relies on adaptive patch sampling.
- In Chapter 7, we make final considerations about our contributions in this thesis and we discuss a number of limitations and perspectives for future research.
- Finally, Appendix A, B and C provides a number of supplementary results generated by the techniques proposed in this thesis.

2.6 Author's Publications

A number of paper publications took place during the preparation of this thesis. The preprints of the paper publications mentioned below can be found at `http://oriel.github.io/`.

[2] http://www.technicolor.com/en/solutions-services/cinestyle

2.6.1 Journal Papers

- Oriel Frigo, Neus Sabater, Julie Delon, Pierre Hellier: *Motion driven tonal stabilization.* Accepted to IEEE Transactions on Image Processing.
- Oriel Frigo, Neus Sabater, Julie Delon, Pierre Hellier: *Video Style Transfer by Adaptive Patch Sampling.* In preparation.

2.6.2 International Conferences

- Oriel Frigo, Neus Sabater, Julie Delon, Pierre Hellier: *Split and Match: Example-based Adaptive Patch Sampling for Unsupervised Style Transfer.* CVPR 2016.
- Oriel Frigo, Neus Sabater, Julie Delon, Pierre Hellier: *Motion driven tonal stabilization.* ICIP 2015.
- Oriel Frigo, Neus Sabater, Vincent Demoulin, Pierre Hellier: *Optimal Transportation for Example-Guided Color Transfer.* ACCV 2014.

2.6.3 National Colloquium

- Oriel Frigo, Neus Sabater, Julie Delon, Pierre Hellier: *Stabilisation tonale de vidéos.* GRETSI 2015.

2.6.4 Patents

- WO2014184244 A1 - Method for transferring the chromaticity of an example-image to the chromaticity of an image.
- WO2014184157 A1 - Method for adapting the colors of an image to the colors of an example image.
- 2014110015 - Motion-based video tonal stabilization
- 2015100086 - Unsupervised Style Transfer by Adaptive Patch Sampling

Chapter 3

Computational Color

Contents

Color is one of the basic attributes of human visual perception, together with shape, texture and line. It plays an important role in daily life, either as a way of identifying objects or as a source of expression in visual arts.

In the field of Digital Image Processing and Computer Vision, color images were not widely used until the 90's, when the storage and processing capacity of computers increased allowing to manipulate three-dimensional images [Gonzalez 2007].

Since then, color image processing has been mainly performed by predefined filters or transformations that perform specific operations such as correction and enhancing. White balance is possibly the most used form of color transformation and is one of the main steps in the camera imaging pipeline.

In this chapter, we cover basic concepts related to color, which are useful as a background to explore the problems of color transfer (Chapter 4) and tonal stabilization (Chapter 5). We start with a discussion on color image formation, along with color constancy, color spaces and finally, radiometric calibration and the camera pipeline.

3.1 Color image formation

A color image is originated from the combination of a light source, a reflective or transmissive material and a photosensor. The spectral properties of the light source are generally described according to the relative amount of energy emitted at each wavelength, namely the *spectral power distribution* of the light source. The light from the source is either absorbed by the surface or reflected.

The fraction of the light reflected by the surface defines the *surface reflectance* function. As a first approximation, we can calculate the light reflected towards the eye by multiplying the spectral power distribution and the surface reflectance function together, forming scattered light (as shown in Figure 3.1). The scattered

Figure 3.1: Illustration of main components in color image formation. a) Example of light interacting with reflective surface and reaching human retina. b) The plots show scattered light as a product of illumination and reflectance, and the amount of cone absorptions. Image courtesy of [Wandell 1995].

light can be called the *color signal* because it serves as the signal that ultimately leads to the experience of color. The color signal leads to different amounts of absorptions in the three cone photoreceptor classes, and the interpretation of these cone absorptions by the nervous system is the basis of our color perception.

Regarding the computational representation of color, the procedure of color image formation is similar. A sensor based on technologies such as CCD or CMOS absorbs the scattered light according to the sensor's sensitivity function, and the resultant colors are represented in a specific color space.

3.1.1 Color Constancy

Starting with Land and McCain work on the Retinex theory of color vision [Land 1971] [Land 1977] [Brainard 1986] [Provenzi 2007], color constancy has been extensively studied and remains an active research topic [Hordley 2006] [Gijsenij 2011] [Mazin 2015]. In computational color constancy, we search for a representation of a scene that is not biased to the illuminant's color, or in other words, a representation that approaches the color constancy ability of human perception. This ability is responsible, for example, to make a white paper be perceived as white regardless of the illuminant in the scene. For an interesting illustration of the color constancy phenomenon, see Figure 3.2. In this illustration, we can see an illusion based on the mechanism of human illuminant adaptation, where we perceive differently the color of the helmet in the left and right images because of the illuminant in the scene. However, physically the helmet in these two images have exactly the same RGB intensity.

Land and McCain have observed the color constancy phenomenon in their ex-

periments with *Mondrian scenes*. These experiments have shown that colors perceived from surface reflectances do not depend on the absolute radiant energy of their stimuli, but that their relative lightness remained invariant independent of the illumination uniformity and color [Sustrunk 2005].

Figure 3.2: Illustration of color constancy. Note that the left and the middle images are the same, except that the left one is overlayed by a blue filter (simulating a blueish illuminant). While the left image seems to have a blue color cast, the original hues can be still identified (the helmet still appears yellowish). In the right image, the filter was just overlayed on the helmet, which now appears greenish because our visual system assumes the hypothesis of neutral illuminant in the scene. Image courtesy of [Sustrunk 2005].

In digital cameras, the feature intended to approximate color constancy is known as automatic white balance (AWB). The most common approach to perform AWB is to first estimate the color temperature of the illuminant in the scene, and then correct the image by compensating the ratio of the estimated illuminant to the canonical (neutral) illuminant with some variant of a Von Kries diagonal transformation.

Hence, it is usual to assume that if the illuminant of a scene could be accurately estimated, the color cast of the illuminant could be discounted [McCann 2005] and the scene could be effectively rendered under a neutral illuminant.

Under controlled conditions, where camera spectral responses are known, the color of the illuminant in a scene could be easily measured [Wyszecki 2000]. However, for digital image processing applications, illuminant estimation is an ill-posed problem. Under the assumption of Lambertian color formation model, the radiance at a pixel $e_c(\boldsymbol{x})$ is given by tristimulus integration

$$e_c(\boldsymbol{x}) = m(\boldsymbol{x}) \int_{\Lambda} I(\lambda)\, p_c(\lambda) S(\boldsymbol{x}, \lambda) d\lambda, \tag{3.1}$$

where $\lambda \in \Lambda$ is the visual range of wavelength spectrum (usually the interval $[380nm, 780nm]$), $m(\boldsymbol{x})$ is the Lambertian shading, $c = \{R, G, B\}$, $I(\lambda)$ is the spectral power distribution (SPD) of the light source, $p_c(\lambda)$ is the camera spectral sensitivity and $S(\boldsymbol{x}, \lambda)$ is the reflectance of the scene under wavelength λ and

spatial coordinate $\boldsymbol{x}$. In practice, the camera spectral sensitivity function is usually not known, thus the color of the illuminant $\boldsymbol{i}$ depends on two variables, $I(\lambda)$ and $p_c(\lambda)$:

$$\boldsymbol{i} = \begin{bmatrix} i_R \\ i_G \\ i_B \end{bmatrix} = \int_\Lambda I(\lambda)\, p_c(\lambda) d\lambda. \tag{3.2}$$

When working with digital images as available data, the only variable we know from Eq. 3.1 is the observed intensity $e_c(\boldsymbol{x})$[1], thus it is clear that the estimation of $\boldsymbol{i}$ is an under-constrained problem. Hence, additional assumptions about the scene need to be made in order to perform illuminant estimation.

Gijsenij *et al* [Gijsenij 2011] in their extensive survey on computational color constancy, have categorized illuminant estimation into *static*, *gamut-based* and *learning-based* methods. Static methods are based on fixed parameter settings, in opposition to gamut or learning based methods which tune their parameters to a specific set of images. In this regard, static methods are simple to implement since they are based on low-level statistics or physics models. The underlying benefits of static illuminant estimation methods is their simplicity and low computational complexity, since estimation and correction are possible to be performed in real time. For performance reasons, automatic white balance in video cameras is likely to be based on static methods for illuminant estimation, more precisely, those methods which make assumptions from low-level statistics.

Most well known low-level statistics methods are based on variants of grey-world or max-RGB assumptions. The *grey-world* hypothesis [Buchsbaum 1980] assume that the average color in a scene corresponds to the color of the illuminant. Thus, the color of the illuminant i is given by

$$\frac{\int \boldsymbol{e}(\boldsymbol{x}) d\boldsymbol{x}}{\int d\boldsymbol{x}} = k\boldsymbol{i}, \tag{3.3}$$

where k is the intensity of the illuminant reflectance, between 0 (black) and 1 (white). The gray-world hypothesis suffers from several limitations: it is highly sensitive to large areas with homogeneous colors in an image, and it is easily biased in images containing few colors. To solve some of this limitations, several works adapted the gray-world method for local instances, such as selecting pixels from gray point candidates, or from segments in the image.

On the other hand, the *white patch* hypothesis, derived from the Retinex theory of color perception, assumes that pixels with maximum reflectance value in a scene correspond to the color of the illuminant:

$$\max_{\boldsymbol{x}} \boldsymbol{e}(\boldsymbol{x}) = (\max_{\boldsymbol{x}} R(\boldsymbol{x}), \max_{\boldsymbol{x}} G(\boldsymbol{x}), \max_{\boldsymbol{x}} B(\boldsymbol{x})) = k\boldsymbol{i}. \tag{3.4}$$

[1] more precisely, in digital images we only know sRGB intensities $F(e_c(\boldsymbol{x}))$

The main limitation of the white patch (*Max RGB*) hypothesis is that a scene should contain a white object, which is not always a realistic assumption for image processing.

Finlayson *et al* [Finlayson 2005] has shown that the white patch and gray-world hypothesis can be generalized as instantiations of the Minkowski L^p norm

$$\left(\frac{\int(\boldsymbol{e}(\boldsymbol{x}))^p d\boldsymbol{x}}{\int d\boldsymbol{x}}\right)^{\frac{1}{p}} = k\boldsymbol{i}. \tag{3.5}$$

For the gray-word hypothesis, we have $p = 1$, while for white patch we have $p = \infty$, both could be considered strong hypotheses in practice, with extreme norm values. In an attempt to find an intermediary optimal norm value, the *shades of gray* illuminant estimation method [Finlayson 2005] is based on a norm value of $p = 6$, which was shown to perform better than the extreme cases of gray-world or white patch.

A number of other low-level statistics methods have been described as extensions to Eq. 3.5. For example, Weijer *et al* [Weijer 2007] presented the *gray-edge* method, which included a derivative operator motivated by the hypothesis that edges in an image provide useful information about the color of the illuminant.

Accordingly, we can generalize the Minkovski norm based illuminant estimation to include the Gaussian smoothing and the image derivative operator, rewriting Eq. 3.5 as

$$\left(\int \left|\frac{\partial^n(\boldsymbol{e}^\sigma(\boldsymbol{x}))^p d\boldsymbol{x}}{\partial \boldsymbol{x}^n}\right|\right)^{\frac{1}{p}} = k\boldsymbol{i}^{n,p,\sigma}, \tag{3.6}$$

where

$$\boldsymbol{e}^\sigma = \boldsymbol{e} * G^\sigma, \tag{3.7}$$

G^σ is the local smoothing kernel with standard deviation σ; n is the derivative order, and p is the norm value.

Finally, some low-level statistics methods have proposed a progressive estimation of the illuminant until a given stopping criteria (such as minimal error or maximum number of iterations) is achieved. For example, in [Gijsenij 2012] an *iterative weighted gray-edge* approach is presented. The method improves the accuracy of gray edge by giving more weight to specular and shadow edges and less weight to material edges. Another iterative illuminant estimation was elaborated by [Huo 2006] as a feedback loop composed of gray color points filtering, error minimization and adaptive gain adjusting.

As we can see, much work has been done in the illuminant estimation literature, notably with respect to the scene assumptions intended to cope with the ill-posed nature of the computational color constancy. We should note that color constancy is still recognized as a problem that has not been solved satisfactorily, neither for computer or human vision.

In practice, the lack of accuracy of automatic white balance algorithms implemented in consumer video cameras comes from the aforementioned limitations of

color constancy. Moreover, for the specific case of video footage, there is the additional requirement to rely in simple assumptions with low computational complexity algorithms, so that illuminant color can be estimated in real time.

Note that some Digital Single Lens Reflex (D-SLR) cameras are equipped with a built-in light temperature detection from its sensor to measure the illuminant of the scene and provide a more accurate automatic white balance [Freeman 2010]. But in low cost cameras such as smartphones, illuminant estimation needs to be approached by strong assumptions about the scene content.

Since we observe that automatic white balance in low cost video cameras tend to perform poorly (producing videos with tonal instabilities that motivate our discussion in Chapter 5), in this thesis we avoid to perform temporal color correction based on explicit illuminant estimation.

3.1.2 Color Spaces

It is well known that colors can be completely described only if considered as a three-dimensional signal. This observation has biological evidence, as humans with normal vision have three types of color photoreceptors (cones) in the retina [Wyszecki 2000].

Color spaces are vector spaces (most of them variants of $\mathbb{R}^3$) which represent color as point coordinates. There are three main categories of color spaces: trichromatic (XYZ, sRGB, CIE RGB, etc.), perceptual (HSI, HSV, HSL, etc.) and luma/chroma (CIELAB, CIELUV, YUV, etc.).

While most digital images are represented in sRGB or some device-dependent variant of RGB color space for storage and displaying, CIELAB has been the most widely space used in Color Science [Wyszecki 2000]. In the sequence, we briefly introduce device-independent color spaces that are fundamental in color science and are related to the scope of this thesis.

3.1.2.1 CIE 1931 XYZ

XYZ color space was one of the first mathematically defined color spaces, created by the International Commission on Illumination (CIE, in French: *Commission internationale de l'éclairage*) in 1931.

The CIE's color matching functions $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ and $\bar{z}(\lambda)$ (shown in Figure 3.3) are called as standard observer, for being numerical descriptions of the chromatic response of the human observer.

The color matching functions $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ and $\bar{z}(\lambda)$ were found by user tests known as color matching experiments and can be considered as the spectral sensitivity curves of three linear light detectors yielding tristimulus values. The CIE tristimulus values X, Y and Z are given in terms of the standard observer by

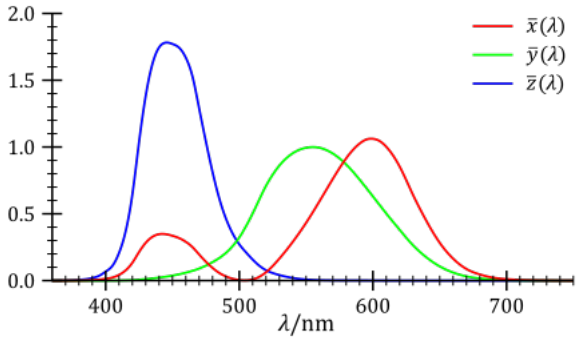


Figure 3.3: Spectral sensitivity curves $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ and $\bar{z}(\lambda)$ [Wandell 1995].

$$X = \int_{380}^{780} I(\lambda)\,\overline{x}(\lambda)\,d\lambda \tag{3.8}$$

$$Y = \int_{380}^{780} I(\lambda)\,\overline{y}(\lambda)\,d\lambda \tag{3.9}$$

$$Z = \int_{380}^{780} I(\lambda)\,\overline{z}(\lambda)\,d\lambda, \tag{3.10}$$

where $I(\lambda)$ is the spectral power distribution of the scattered light at wavelength λ. The equations (3.8), (3.9) and (3.10) are all inner products that can be considered as a projection of an infinite-dimensional light spectrum[2] to a three-dimensional color space. The CIE XYZ color space was deliberately designed so that the Y parameter was a measure of the brightness or luminance of a color.

The chromaticity of a color was then specified by the two derived parameters x and y, two of the three normalized values which are functions of all three tristimulus values X, Y, and Z:

$$x = \frac{X}{X+Y+Z}, \tag{3.11}$$

$$y = \frac{Y}{X+Y+Z}. \tag{3.12}$$

The xy chromaticity diagram represented in Figure 3.4 is an approximation of all chromaticities (maximum saturated colors) that can be perceived by a human observer with normal color vision.

3.1.2.2 CIELAB

CIELAB (also called $L^*a^*b^*$) is a color space developed in 1976 by CIE, as a correction of the Hunter Lab model created in 1948. Similarly to other color systems

[2] λ is defined in the wavelength range of visual sensitivity (between $380nm$ and $780nm$)

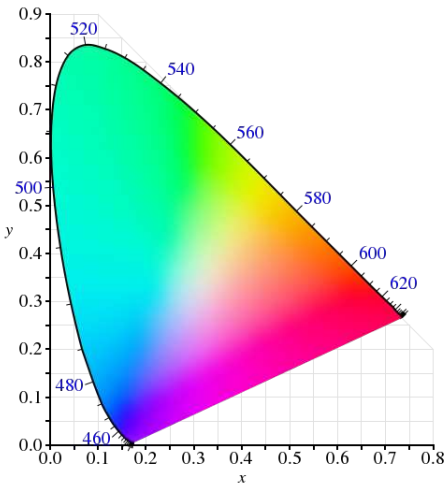


Figure 3.4: Illustration of xy chromaticity diagram [Wandell 1995].

derived from CIE XYZ, CIELAB represents colors with a luminance parameter (L^*) and two chrominance parameters (a^* and b^*). CIELAB is an appropriate color space to compute color differences since it was designed so that Euclidian distances between color points approximately correspond to perceptual color differences for a standard observer.

In order to convert a color value from XYZ to CIELAB, the following non-linear transformation is applied:

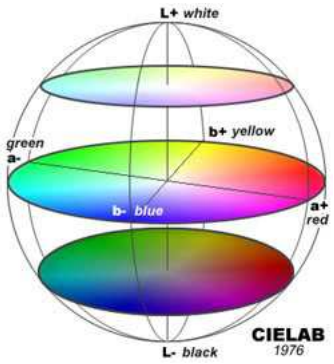


Figure 3.5: Illustration of CIELAB color space [Wandell 1995].

$$L^\star = 116 f(Y/Y_n) - 16 \tag{3.13}$$

$$a^\star = 500\,[f(X/X_n) - f(Y/Y_n)] \tag{3.14}$$

$$b^\star = 200\,[f(Y/Y_n) - f(Z/Z_n)]\,; \tag{3.15}$$

where

$$f(t) = \begin{cases} t^{1/3} & \text{if } t > (\frac{6}{29})^3 \\ \frac{1}{3}\left(\frac{29}{6}\right)^2 t + \frac{4}{29} & \text{otherwise} \end{cases}; \tag{3.16}$$

and X_n, Y_n and Z_n are the CIE XYZ tristimulus values of the reference white point. The non-linear relationships between $L*$, $a*$ and $b*$ have the role of simulate the logarithmic response of the human eye. The inverse transformation (from CIELAB to XYZ) can be expressed using the inverse of $f(t)$:

$$Y = Y_n f^{-1}\left(\tfrac{1}{116}\left(L^* + 16\right)\right) \tag{3.17}$$

$$X = X_n f^{-1}\left(\tfrac{1}{116}\left(L^* + 16\right) + \tfrac{1}{500}a^*\right) \tag{3.18}$$

$$Z = Z_n f^{-1}\left(\tfrac{1}{116}\left(L^* + 16\right) - \tfrac{1}{200}b^*\right) \tag{3.19}$$

where

$$f^{-1}(t) = \begin{cases} t^3 & \text{if } t > \frac{6}{29} \\ 3\left(\frac{6}{29}\right)^2\left(t - \frac{4}{29}\right) & \text{otherwise.} \end{cases} \tag{3.20}$$

Different color difference equations have been proposed in the literature to improve the uniformity of color distances in CIELAB. This is motivated by the fact that simple Euclidian distances do not reproduce accurately the perceptual color difference for certain areas of the color space.

3.1.2.3 LMS

LMS is a color space developed to simulate the responsivity of the three types of cone photoreceptors in human retina. Its name derives from the cone responses at **L**ong, **M**edium and **S**hort wavelengths. It is common to use the LMS color space when performing chromatic adaptation (estimating the appearance of a sample under a different illuminant), as we do in Chapter 4 of this thesis in an example based fashion.

There are many different matrix conversions intended to convert from XYZ (CIE standard observer) to LMS color space in the literature, which have been found by numerical computations intended to simulate different perceptual phenomena. These conversion matrices are usually called Chromatic Adaptation Transform (CAT) matrices. Two of the state of the art CAT matrices are CMCCAT97 and CAT02.

The CMCCAT97 matrix, which is used in CIE 1997 Color Appearance Model[3] [Luo 1998], is based on the Bradford transformation [Finlayson 2000] given by

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.8951 & 0.2664 & -0.1614 \\ -0.7502 & 1.7135 & 0.0367 \\ 0.0389 & -0.0685 & 1.0296 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \tag{3.21}$$

[3] A Color Appearance Model (CAM) represents colors considering different lighting conditions, backgrounds and other perceptual phenomena [Fairchild 2005].

The chromatic adaptation matrix CAT02, from the CIECAM02 Color Appearance Model [Moroney 2002] is given by

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.7328 & 0.4296 & -0.1624 \\ -0.7036 & 1.6975 & 0.0061 \\ 0.0030 & 0.0136 & 0.9834 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.22}$$

In Chapter 4 where we discuss the example-based Chromatic Adaptation Transform, we use the CAT02 matrix, as it is the most up to date CAT matrix recommended by CIE and has been proved efficient in comparison to previous matrices [Moroney 2002].

3.1.2.4 l$\alpha\beta$

The color space $l\alpha\beta$ was proposed by Ruderman and colleagues [Ruderman 1998] in order to achieve maximum decorrelated axes. The advantage of working in a color space with decorrelated axes is the possibility of modifying each channel of the image independently, without affecting the others. In this manner, we can perform an independent transformation for each channel and avoid to apply a complex three-dimensional transformation.

A well known technique to convert data to uncorrelated space is given by Principal Component Analysis. Rather than computing the PCA for each image separately, Rudermann and colleagues [Ruderman 1998] claimed that images given in LMS color space can be converted to $l\alpha\beta$ color space which on average decorrelates its axes with a conversion matrix. The transformation to convert an image from LMS to $l\alpha\beta$ color space is given by

$$\begin{bmatrix} l \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} logL \\ logM \\ logS \end{bmatrix} \tag{3.23}$$

This transformation converts original pixels from LMS to log space, no longer requiring color values to be positive. In, [Reinhard 2001b] it is claimed that the logarithmic representation is useful for color manipulation because of its compactness, where uniform changes in stimulus tend to be equally detectable.

3.1.2.5 sRGB

The sRGB color space, or standard RGB (Red, Green, Blue - norm REC 709), was proposed in 1996, and since then has gained widespread use in digital cameras, displays and is probably the most commonly used format to share images on the internet. The transformation from CIE XYZ to sRGB is obtained from a matrix multiplication followed by a non-linear gamma correction. The transformation from XYZ to linear sRGB is given as follows:

$$\begin{bmatrix} R_{\text{linear}} \\ G_{\text{linear}} \\ B_{\text{linear}} \end{bmatrix} = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.24}$$

.

Note that the sRGB color space was designed assuming that color intensities are observed in a display which have standard $\gamma = 2.2$. The transformation from linear RGB to sRGB, taking into account the gamma correction, is given by

$$C_{\text{srgb}} = \begin{cases} 12.92 C_{\text{linear}}, & C_{\text{linear}} \leq 0.0031308 \\ (1+a) C_{\text{linear}}^{1/2.4} - a, & C_{\text{linear}} > 0.0031308 \end{cases} \tag{3.25}$$

where $a := 0.055$; C_{linear} denotes R_{linear}, G_{linear} or B_{linear}; C_{linear} and correspondenly, C_{srgb} denotes R_{srgb}, G_{srgb} or B_{srgb}.

3.1.2.6 YUV

YUV is a space that represents color by taking into account the fact that human perception is less sensitive to chrominance than luminance. Thus, reduced bandwidth is used for chrominance components for compression purpose. The terms YUV, Y'UV, Y'CbCr, YPbPr are all related and as such can lead to ambiguity. YUV and Y'UV were initially proposed for color encoding in analog television systems, while Y'CbCr was proposed for color encoding in digital image and video standards such as MPEG and JPEG. In practice, Y denotes luminance (the perceptual sensation of light) and Y' denotes luma, which corresponds to the luminance transformed by a gamma compression.

In current image processing literature, the term YUV is commonly used to refer to images encoded using Y'CbCr, and this is the case in this manuscript. The following linear transformation is used to convert a color in sRGB to a color in YUV:

$$\begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.28886 & 0.436 \\ 0.615 & -0.51499 & -0.10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.26}$$

where Y' denotes the luma component and Cb and Cr define the two chrominance components.

3.1.3 Radiometric Calibration and Camera Pipeline

Radiometric calibration is a problem that consists in retrieving the approximate irradiances from observed images. To introduce the discussion on radiometric calibration, let's assume that we have an image $u : \Omega \to \mathbb{R}$ taken with a camera from which the response function is known. The camera response $f : \mathbb{R} \to \mathbb{R}$ converts the irradiance e (amount of light collected by the sensor) into observed image u, usually following a non-linear relation:

$$u = f(e). \tag{3.27}$$

Note that the camera response should be though as an operator designed to deliver visually pleasant images, rather than accurate light measurements. In addition, the camera response takes into account the fact that displays have a non-linear rendering (gamma correction). Although in earlier works of [Mitsunaga 1999] [Grossberg 2002], f is modelled as a scalar function producing brightness intensities (or equal responses for each color channel), more recent models of [Chakrabarti 2009] [Lin 2011] [Kim 2012] include a more complete color processing pipeline, composed of white balance transformation, gamut mapping and color space conversion (from camera specific RGB to sRGB). Different sources of noise are also inherent to the imaging pipeline, but for the sake of simplicity we ignore noise sources in the model.

In order to include color to the complete camera response model, Kim *et al.* [Kim 2012] write the observed trichromatic responses as

$$\begin{bmatrix} u(\boldsymbol{x}, R) \\ u(\boldsymbol{x}, G) \\ u(\boldsymbol{x}, B) \end{bmatrix} = \begin{bmatrix} f_r(e(\boldsymbol{x}, R)) \\ f_g(e(\boldsymbol{x}, G)) \\ f_b(e(\boldsymbol{x}, B)) \end{bmatrix}, \tag{3.28}$$

where f_R, f_G, f_B accounts for channel-wise tone responses (f being different for R, G, B color channels),

$$\begin{bmatrix} e(\boldsymbol{x}, R) \\ e(\boldsymbol{x}, G) \\ e(\boldsymbol{x}, B) \end{bmatrix} = h(\boldsymbol{T_s T_w e}), \tag{3.29}$$

$\boldsymbol{T_s}$ is a 3×3 matrix transformation that accounts for color space conversion, $\boldsymbol{T_w}$ is a diagonal 3×3 matrix accounting for white balance and $\boldsymbol{e}$ is the original irradiance vector (RAW[4] intensities) and h is a non-linear gamut mapping operator. Note that the camera spectral sensitivity is not included in this response model, so $\boldsymbol{e}$ should be seen as the irradiance responses after tristimulus integration (see Eq. 3.1 for color formation model), vignetting and signal amplification (ISO). Since most digital consumer cameras capture colors with a single sensor overlayed by a mosaic of red, green and blue color filters arranged in an array such as Bayer pattern, the response of these filters constitute the camera spectral sensitivity.

An illustration of the complete camera pipeline, mentioned above, is shown in Figure 3.6. Note that it is not in the scope of this thesis to discuss in details all the steps of the camera pipeline, but mostly the steps that influence directly in color transfer and tonal instability problems.

Radiometric calibration is a field that has the precise goal of estimating the camera response function, usually by relying on multiple images of a scene taken with varying exposures. The basis assumption for traditional radiometric calibration is that multiple registered images from the same scene (same illumination and radiances) will differ in image intensity because of changes in camera exposure.

[4]Following [Kim 2012], we consider RAW as the sensor responses after demosaicing step.

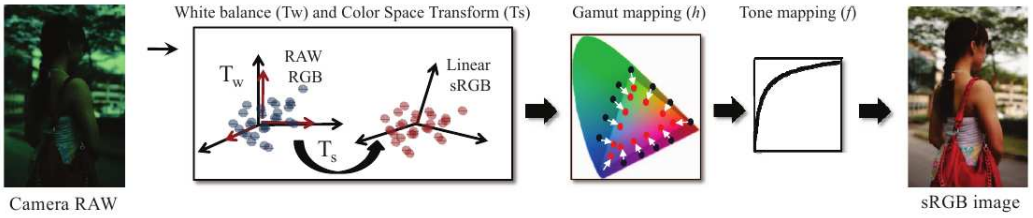


Figure 3.6: Illustration of colorimetric transformations in camera pipeline. Image adapted from [Kim 2012].

Formally, given a pair of registered images u_0 and u_1, we will have

$$\frac{f^{-1}(u_0)}{f^{-1}(u_1)} = k, \tag{3.30}$$

which implies that the two images are linearly related by an exposure ratio of k. Most works in radiometric calibration assume a prior to the response function, such as a power law (gamma curve) [Mann 1995], a polynomial curve [Mitsunaga 1999] or an eigenvector basis [Grossberg 2003]. Then, to obtain an estimate of f, corresponding patches from pair of images taken with different exposures are related by

$$u_1 = \tau_k(u_0) = f(kf^{-1}(u_1)) \tag{3.31}$$

where τ_k is the brightness transfer function (BTF) which describes the exposure change under a given camera response.

While the tone map f, the white balance matrix $\boldsymbol{T_w}$ and the color space conversion matrix $\boldsymbol{T_w}$ can all be estimated by parametric least squares fitting, the gamut mapping h needs more complex non-parametric modelling. [Lin 2011] estimates the inverse gamut mapping by radial basis function (RBF) interpolation, while [Xiong 2012] relies on bayesian regression. Note that both methods assume the existence of a training dataset of RAW-sRGB images for different cameras. In this way, they compute an interpolation of these correspondences to obtain an approximation of the camera's gamut mapping. Unfortunately, this RAW-sRGB data is not available for many consumer cameras, in special mobile phone cameras.

3.2 Considerations

In this chapter, we have introduced general concepts related to the color image formation process, color constancy and radiometric calibration. These concepts play an important role for the understanding of color transfer and color stabilization, which are respectively discussed in Chapters 4 and 5.

We have seen that colors are represented in certain vector spaces. It should be noted that the choice of a given color space often makes difference when solving the problems of color transfer, color correction and texture transfer. A proper color space should be chosen according to the specificities of the problem. For instance,

for color transfer, distances between color distributions are computed in CIELAB space for a color comparison that is perceptually relevant. For tonal stabilization, we observed that sRGB color space is more appropriate for curve fitting than CIELAB, which motivates our choice for the first.

Finally, it can be noted that a representation of an image in a chromaticity versus luminance space such as CIELAB and YUV is a simple way to split an image into color and textural/structural content. We use this technique in practice for both color and texture transfer, where color is transferred to the chromaticity channel of an image, and texture is transferred only to the luminance channel.

In Chapter 4, we continue our discussion in the context of computational color, by introducing the problem of color transfer.

Chapter 4

Color Transfer

Contents

4.1 Introduction

In the traditional techniques or photo editing softwares, an user interested in enhancing the colors of an image should tune some filter parameters or know the correct color transformation to be applied in order to reach the desired result.

In this chapter, we explore an alternative approach for color correction and enhancement, which consists in letting the user to choose examples to perform the desired operation in such a way that the original image will have the same color characteristics of the example.

Color transfer is a research field that concentrates on the color aspect of example based image processing, that is to say, the problem of transferring the color of an example image to an original input image. There are two main applications of color transfer methodologies, that have been called as colorization and recolorization (or automatic color grading). Basically, colorization is a procedure to colorize an image that was originally taken without colors. Recolorization, on the other hand, aims to rearrange the colors of an original image, given an example color palette. In this chapter, we concentrate on recolorization, thus whenever we use the term color transfer we refer to the recolorization or automatic color grading procedure.

In film post-production, *color grading* (changes in color, contrast, white balance and hues) is usually performed manually by the colorist, who could clearly benefit from automatic and consistent color transfer methods to aid his work. Similarly, editing tools for large personal photo collections could be improved with example-based algorithms, specially considering the last trend of editing sets of multi-contributor images of a given event (party, wedding, *etc*).

In order to achieve automatic color grading, the computational color transfer technique must achieve a consistent and appropriate color, brightness and contrast transformation. Artifacts and non-natural colors should be avoided, while computational space and time complexity should be kept as low as possible.

Note that color transfer can be seen as a general color correction procedure. Differently to color constancy and radiometric calibration processes (presented in Chapter 3) color transfer makes few assumptions about the physics of the camera and the scene. Nevertheless, color transfer can be also employed to compute color constancy for an input image. For that, we can simply take an image under neutral white balance and use it as example to estimate a color transformation. We present this approach for example-based color constancy in Section 4.3.1.

A common drawback of color transfer methods is the strong tendency to create undesired visual artifacts. For instance, existing noise or compression "block effects", that are initially barely noticeable, can become prominent. Hence, in order to achieve automatic color transfer, the considered method must achieve a visually-plausible and appropriate color transformation. Considering these requirements, in this chapter we propose an example-based method for automatic color transfer where the illuminant matching and the transfer of dominant colors of the images are treated separately. Moreover, our method carries out a final regularization of the color transform avoiding new parasite structures in the image. We also show that the color transformation can easily be applied to videos without any color flickering.

4.2 Related Work

[Reinhard 2001a] were pioneers in establishing the concept of color transfer, with an approach to modify the color distribution of one given original image based on the global color statistics (mean and variance) of an example image in the decorrelated color space $l\alpha\beta$. Other works have proposed global color transfer in terms of non-linear histogram matching [Neumann 2005, Papadakis 2011, Pouli 2011] or N-Dimensional Probability Distribution Function (N-PDF) transfer [Pitié 2007]. Removing the inherent artifacts due to color modifications is the main goal of some other works. For example, in [Rabin 2011] a non-local filter is studied to remove spatial artifacts.

Related work also includes [Tai 2005] in which the color transfer is defined on color segments given by an Expectation-Maximization adapted algorithm. However their color mapping is essentially different from our color mapping which is based on the Monge-Kantorovitch optimal transportation problem. In [Freedman 2010,

Ferradans 2013] the color transfer problem is also presented in terms of optimal transportation. Other works adapted the flow-based color transfer representing the colors in the image by compact signatures given by Gaussian Mixture Models [Murray 2011] or super-pixel segmentation [Wu 2013].

Another class of methods such as [HaCohen 2011] [Faridul 2013] [Hwang 2014] assumes that there are spatial correspondences to be found between the input and example image, these correspondences being used to derive a color transformation. The assumption of geometrical relationship drastically reduces the scope and genericity of the method.

Few works have introduced semantic constraints in the context of color transfer or color enhancement. In [Cusano 2012], a semantic annotation of input and example images is obtained by training a Support Vector Machines and classifying regions in the image according to a trained model. The method proposed by [Lindner 2012] performs semantic image enhancement based on correspondences between image characteristics and semantic concepts.

Finally, in parallel with the development of this work, color transfer for videos was proposed by [Bonneel 2013]. Their method relies on a temporal interpolation that minimizes the curvature of color transformations. Results are impressing for color transfer of videos, but it is not evident how to adapt it for film post-production workflow. The method is likely to produce smooth temporally varying color transformations, while colorists traditionally rely on a single color transformation per shot of film for color grading.

Despite the significant progress made since the seminal paper [Reinhard 2001a], color transfer remains a challenging problem. Indeed, we think that the current approaches are still limited in some situations (strongly dependent on the selected images) and are prone to create image artifacts. On the one hand, linear color transfer methods are robust and usually do not introduce noise. However, they do not perform well when there are several objects with different colors, since the linear transformation cannot account for the magnitude of color change that is expected. On the other hand, highly non-linear transformations seem to be more robust but at the cost of amplifying noise and introducing undesired structures when a local histogram stretching arises. Besides, all techniques may transform the image with unnatural results. For example, an object receiving non-plausible colors as a green face.

4.3 Our Approach

In this section, we present two contributions to color transfer: an example-based CAT for illuminant matching (Sec. 4.3.1) and a color transfer based on automatic color palette associations (Sec. 4.3.2). These two methods are independent and complementary, and achieve convincing results for challenging images when used together. Moreover, we show how our color transfer can optionally be constrained with semantic attributes like saliency or faces (Sec. 4.3.3).

In order to limit the aforementioned artifacts, we propose to process separately the luminance and chroma channels of the image. Basically, the luminance channel will be addressed using a novel example-based CAT, accounting for the illuminant change, while the chroma channels will be transformed using optimal transportation. In fact, we have observed a substantial improvement in our results with this approach compared to other color transfer techniques that treat jointly luminance and chroma.

4.3.1 Example-based CAT

In [Reinhard 2001a], it is mentioned that one interesting application for color transfer is to remove undesirable color cast from an image, such as the yellowish colors in photos taken under incandescent illumination. Although this description reminds the color constancy problem, as far as we know, color constancy and chromatic adaptation has not been approached in the color transfer literature. In digital photography, adjusting the lighting is known as white-balance or color-balance and is modified with respect to a standard illuminant. In this work we propose to modify the illuminant of the input image with respect of the example image illuminant.

In most cases, no information is available about the spectral power distribution of the illuminant in the scene. Hence, the solution is to estimate the color of the illuminant (the white point in the scene) based on the digital image. A simple approach to address this problem is to consider a variant of the "grey world assumption" [Huo 2006], which assumes that the mean value of a natural image tends to be a grayish color corresponding to the color of the illuminant. Formally, given an input image u and an example image v, the goal is to modify u so as to adapt its illuminant to the estimated illuminant of v. For that, we propose the following example-based CAT algorithm:

1. Estimate the white point (illuminant) of image v. *For a given value t (we set $t = 0.3$, more discussion on [Huo 2006]), the white point of an image is defined as the mean color value of all pixels such that*

$$\frac{|a^*| + |b^*|}{L^*} < t \,, \tag{4.1}$$

   where a^*, b^* and L^* denote the pixel coordinates in the CIELAB color space.

2. Estimate similarly the white point of image u.

3. Perform the chromatic adaptation transform (CAT) on u to adapt its white point to the white point of v. This transformation is described below.

4. Repeat Steps 2 and 3 until (a) the maximum number of iterations has been reached or; (b) the u white point has not changed from the previous iteration.

5. Return image u' which has the same geometry of u but with colors adapted to the illuminant of v.

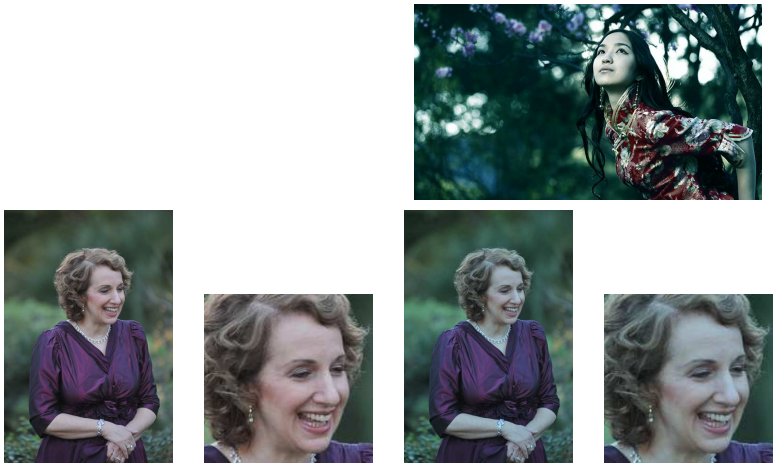


Figure 4.1: Illustration of the example-guided chromatic adaptation transform (CAT), where the illuminant of an example image is estimated and transferred to another image. Left column: input image and cropped image around the face. Right column: result of the example-guided CAT with a cold light estimated from the example image on the top. In both cases, the light cast has been estimated and transferred to the input image, specially visible on the face. Images are best viewed on the electronic version.

Now, let us describe the CAT transform (Step 3). Let (L_I, M_I, S_I) and (L_E, M_E, S_E) denote respectively the estimated white points in LMS color space. Then, the CAT linear transform is defined as:

$$M_A^{-1} \cdot \mathrm{diag}\left(L_E/L_I, M_E/M_I, S_E/S_I\right) \cdot M_A\,, \tag{4.2}$$

where M_A is a CAT matrix[1] that transforms from XYZ to LMS cone space. This transformation rescales the color values of u based on the ratio of input and example white points so that the colors of u appear to have the same illuminant of v.

The algorithm is performed iteratively, hence the user can control the desired degree of adaptation to the example image according to the maximum number of iterations parameter. Experimentally, it was assessed that no more than 30 iterations are needed for an acceptable illuminant adaptation, limiting the risk of over-adaptation.

Figure 4.1 shows a result of the example-based CAT, where an input image (left) is corrected so as to match the warm cast of an example image (middle column) or the cold cast of another example (right column).

Notice that the example-based CAT depicted here can be used either as a pre-processing step before applying chroma color transfer, or as a standalone technique to perform a smooth color transfer accounting only for the illuminant of the example image.

[1] Many CAT matrices exist in literature, such as CAT02, Bradford, CMCCAT2000, Sharp, *etc.* The state-of-the-art CAT02 transformation matrix [Moroney 2002] is used in our work.

4.3.2 Color Chroma Transfer

The intuition of our color chroma transfer method is to use optimal transportation as an efficient tool to map two color distributions approximated by their palettes, regardless of the number of colors in each set. In order to define the color chroma transfer, we propose to use a compact and robust description of the image by its set of meaningful color modes. In particular, we rely on a non-parametric color segmentation known as ACoPa (Automatic Color Palette) [Delon 2007]. This is a non-supervised technique, so the user does not need to specify the number of color modes in advance, as they are automatically extracted from the color histogram based on meaningful (*a contrario*) peaks. After extracting the set of modes from the input and example images, the color transfer based on the optimal mapping between these two sets of modes is performed (see Fig. 4.2).

More precisely, given an input image u and an example image v with its set of meaningful modes P (and Q respectively), the mode mapping function $f\colon P \to Q$ matches each input image mode with one or more example image modes. In practice, we propose a soft assignment method to compute many-to-many matches that minimizes the transportation cost between the set of modes P and Q. An effective solution for this problem comes from the Monge-Kantorovich theory of optimal transportation. Optimal transportation is a well-grounded and solid mathematical field that proves and characterizes the existence of optimal solutions for the transportation problem, minimizing the total displacement cost between two probability distributions. This displacement cost is also known as Wasserstein distance or Earth Mover's Distance (EMD) [Rubner 2000]. Let $P = \{\mathbf{p}_i\}_{i\in[1,m]}$ and $Q = \{\mathbf{q}_j\}_{j\in[1,n]}$

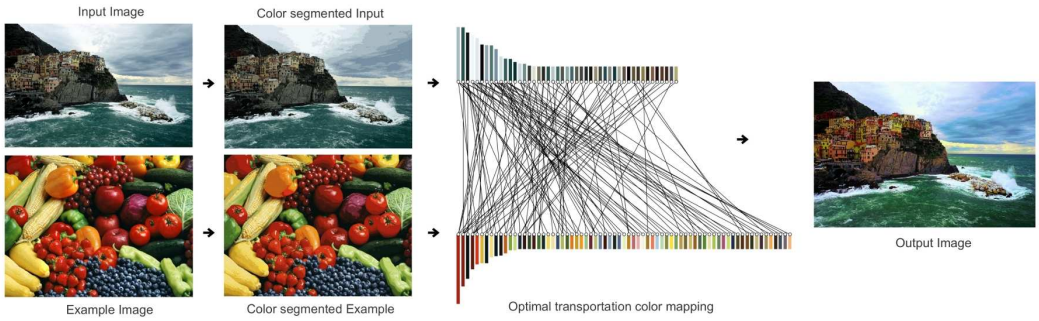


Figure 4.2: Overview of the proposed example-guided color transfer methodology. After extraction of the meaningful color palettes [Delon 2007], a color mapping is estimated as the solution of the optimal transportation problem. Finally, a smooth color transform, computed as a $3D$ thin plate spline interpolation [Bookstein 1989], generates an artifact-free image.

be the input and example signatures with m and n modes respectively, where $\mathbf{p}_i$ and $\mathbf{q}_j$ are the mode representatives. Each mode representative is a six-dimensional vector $\mathbf{p}_i = (\mu_i^l, \mu_i^a, \mu_i^b, \sigma_i^l, \sigma_i^a, \sigma_i^b)$ composed of its mean and standard deviation (both defined as three-dimensional points in the CIELAB color space).

Let $\mathbf{D} = [d_{ij}]$ be the distance matrix where d_{ij} denotes the distance between modes i and j: $d_{ij} = \|\mu_i - \mu_j\|_2 + \|\sigma_i - \sigma_j\|_2$, where μ_i and μ_j (resp. σ_i and σ_j) are the mean (resp. standard deviation) color values of u and v over the modes i and j. Thus we aim to find $\mathbf{F} = [f_{ij}]$, with f_{ij} being the flow of the assignment between i and j minimizing the cost $\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}d_{ij}$, subject to the four following constraints:

$$(1) \quad \forall j \in [1,n], \quad \sum_{i=1}^{m} f_{ij} \leqslant \frac{1}{n}\,; \qquad (2) \quad \forall i \in [1,m], \quad \sum_{j=1}^{n} f_{ij} \leqslant \frac{1}{m}\,;$$

$$(3) \quad \forall i \in [1,m], \forall j \in [1,n], \quad f_{ij} \geqslant 0\,; \qquad (4) \quad \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = 1\,.$$

In practice, the soft assignment matrix $\mathbf{F} = [f_{ij}]$ is obtained using the Simplex linear programming algorithm [Rubner 2000]. After finding $\mathbf{F}$, for each input mode $i = 1, ..., m$, an averaged and weighted mean is computed

$$\widehat{\mu}_i^k = \frac{\sum_{j=1}^{n} f_{ij}\mu_j^k}{\sum_{j=1}^{n} f_{ij}}\,, \tag{4.3}$$

where $k = \{a, b\}$ stands for chroma channels in the CIELAB color space. Based on the fact that the human visual system is more sensitive to differences in luminance than in chroma, we only use chroma channels in the color transformation, avoiding artifacts that would occur if the luminance channel was also used. Then, the color transfer between v and u is encoded giving the set of color correspondences $\Upsilon = \left\{(L_i, \mu_i^a, \mu_i^b), (L_i, \widehat{\mu}_i^a, \widehat{\mu}_i^b)\right\}_i$. Now, we have seen in practice that using Υ to apply a piecewise linear color transform to image u would create an output image with new undesired color edges at the color segment borders. Instead, we apply a *smooth* color transform to image u, computed as a $3D$ thin plate splines interpolation [Bookstein 1989] using the set of color correspondences Υ in the RGB color space. Hence, it is guaranteed that the color transform applied to u is the best color transform in terms of optimal transportation and smoothness. Thin plate splines interpolation is only used as a final regularization to reduce edge artifacts. Note that the described method could create new colors i.e. colors that are not present in the input nor in the example image. Indeed, color statistics values $\widehat{\mu}_i^k$ are computed based on the weighted average of associated color statistics from the example image (Eq. 4.3). So, these averaged values can be seen as the result of an additive color mixture which is likely to create new colors. The risk of such color mixture model is to modify the input image with false or non realistic colors. However, this risk is limited thanks to the matching between the u illuminant and the v illuminant (cf. Sec. 4.3.1). Furthermore, the mapping can be constrained with visual or semantic priors as described in the next section.

4.3.3 Semantic Constraints on Color Transfer

The color chroma transfer described above does not need any prior knowledge of the input and example images. Nonetheless the color mode mapping can be easily adapted to take into account some semantic information. The main idea is to constrain the color transfer in such a way that modes corresponding to the same semantic components of the input and example images are matched together. Given the two images u and v, let us assume that the color modes can be separated into two classes $P = \{\hat{P} \cup \tilde{P}\}$ (resp. $Q = \{\hat{Q} \cup \tilde{Q}\}$) based on a spatial knowledge as visual attention or object segmentation. Then, two different color mappings g and h are computed as solutions to the bipartite graph matching in terms of Earth Mover's distance such that:

$$g : \hat{P} \to \hat{Q}, \text{ and } h : \tilde{P} \to \tilde{Q}. \tag{4.4}$$

In order to satisfy these constraints, the optimization is split into two different transportation problems. In the following, we describe how semantic constraints as visual saliency and faces can be easily adapted to this framework.

Saliency

The saliency map is a two-dimensional representation of conspicuity (gaze points) for every pixel in an image. In this work we use the saliency map S deduced from a coherent psychovisual space proposed in [Le Meur 2006]. Given the saliency map S of an image, each color mode i is considered as salient if

$$\frac{1}{\sharp R_i} \sum_{x,y \in R_i} S(x,y) > \rho \,,$$

where R_i is the list of pixel coordinates (x, y) belonging to the color mode i and $\sharp R_i$ is its cardinal. The parameter ρ is typically set to $\rho = 0.7$, meaning that at least 70% of the pixels belonging to a color mode are salient. Finally, salient modes are mapped to salient modes (and non salient modes to non salient modes), as described in Eq. (4.4).

Faces

Face detection can be also easily incorporated in the color transfer method. The main objective is to ensure fidelity to skin tones and avoid unrealistic colors being assigned to faces and skin. Here, the popular face detector methodology[2] of Viola *et al.* [Viola 2004] is used. The face detection is performed on both images u and v. Two cases of interest are considered:

- Faces are found in u and v. In this situation, we impose that the modes extracted from faces in u are mapped with the modes extracted from faces in v to ensure skin tones transfer.

[2] implementation available in the OpenCV library. http://opencv.org/

- Faces are found in u, but not in v. In this case, colors corresponding to face and skin are not modified to ensure skin tones fidelity.

4.4 Results

In this section we present experimental results that illustrate the efficiency of our color transfer method. We strongly recommend the reader to look the figures on the digital version of the manuscript to appreciate the results. For all the experiments we use the example-based CAT followed by the color chroma transfer. We present four result sections. In Sec. 4.4.1, a comparison with five state-of-the-art color transfer methods is performed, and an objective assessment of the transformation consistency is proposed. Then, a comparison to a state-of-the-art local color transfer method is presented in Sec. 4.4.2. We also show the benefit of adding additional semantic constraints in specific situations in Sec. 4.4.4. Finally, results on video color transfer are presented in Sec. 4.4.3.

4.4.1 Evaluation of Image Color Transfer

First of all, we compare our results with five state-of-the-art global color transfer techniques from which authors have made their code available (see figures 4.3, 4.4 and 4.5): the seminal method of Reinhard [Reinhard 2001a], the N-PDF method from Pitié *et al.* followed by the regularization proposed by Rabin *et al.* [Pitié 2007, Rabin 2011]; the variational method from Papadakis *et al.* [Papadakis 2011], the histogram reshaping method described in [Pouli 2011]; and the regularized transportation method from [Ferradans 2013]. Note that for the experiment on Fig. 4.3 (Scotland landscape, firstly appeared on [Pitié 2007]), methods [Pouli 2011, Papadakis 2011] produces a result with noise amplification artifacts, while the result of [Ferradans 2013] has undesired edges on regions that were originally homogeneous on the input image. On the other hand, our method produces a result without artifacts, similarly to the result obtained by [Pitié 2007, Rabin 2011], with the advantage that we do not need to rely on post-processing image regularization that blurs the output image. For the challenging test pair of Fig. 4.4, only our method is able to adapt the low-saturated colors of the original image (Manarola, Italy on a cloudy day) to the colorful palette of the example image (fruits and vegetables) while keeping a natural and convincing result. Note that state-of-the-art methods produce results with lower color dynamics where all the houses and the rock are reddish. Finally, for Fig. 4.5, the state-of-the-art methods produce color aberrations on the sky and houses, leading to unnatural colors. Color transfer evaluation is not at all straightforward, since it depends on subjective aesthetic preferences. In addition, the purpose of color transfer can be diverse, for example content homogenization or artistic content colorization. However, we argue that color artifacts are not accepted as good results. In Fig.4.3, the halos in the sky in [Pitié 2007, Rabin 2011], the uniform reddish color transfer for the fruits in Fig. 4.4 or the inconsistent and unnatural color transfer (e.g. Fig. 4.5) are examples of not tolerable color artifacts.

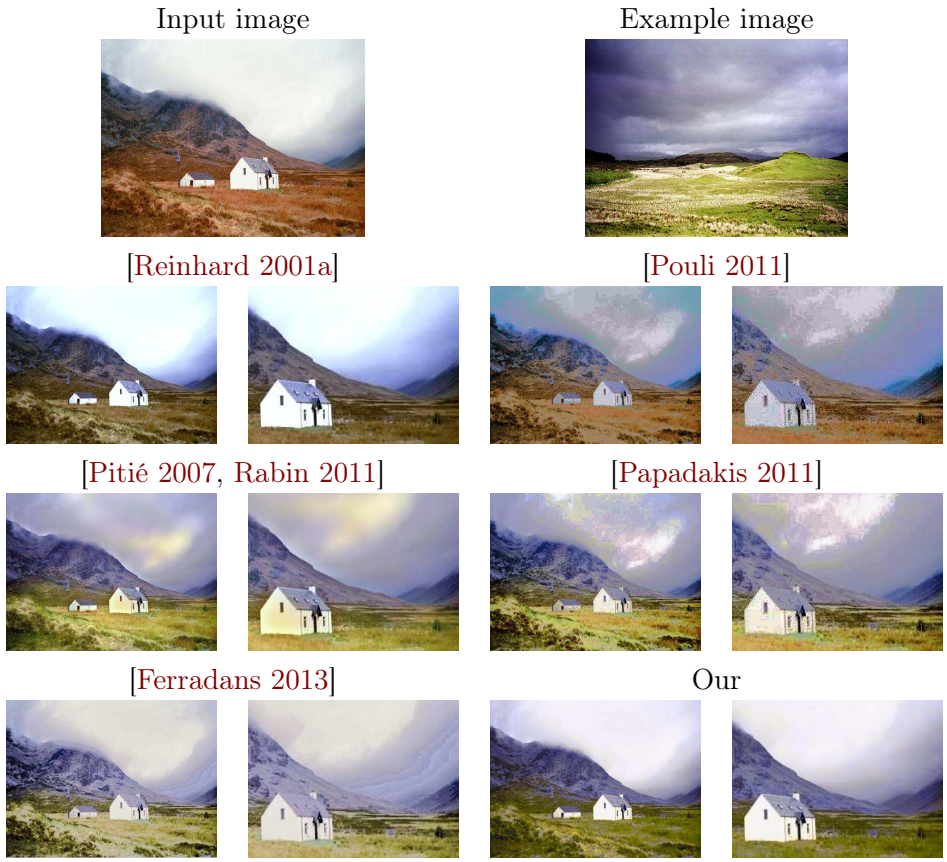


Figure 4.3: Results obtained by state-of-the-art color transfer techniques compared to our method for the Scotland image. For this image pair, a zoom shows color mapping artifacts. For instance, the whitish appearance of the house should be preserved, and banding/noise artifacts are visible in the sky. On the contrary, our method generates a visually plausible and artifact-free result. Images are best viewed in color and on the electronic version.

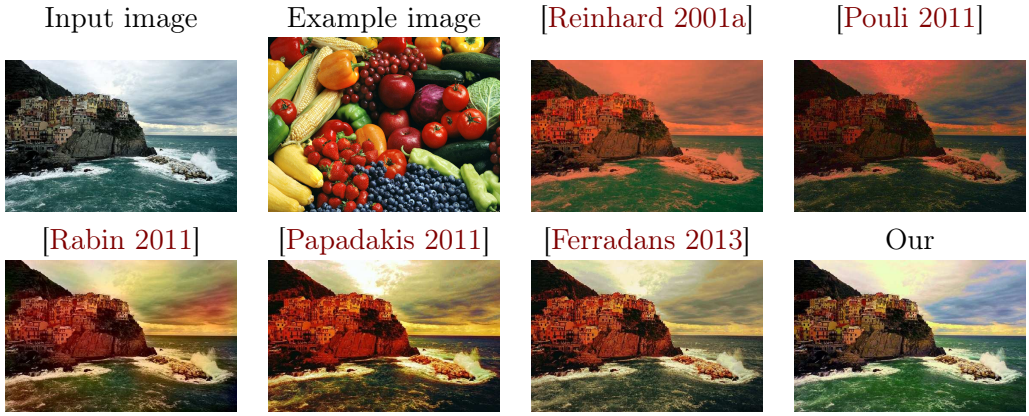


Figure 4.4: Results obtained by state-of-the-art color transfer techniques compared to our method for the Manarola/fruits pair. While state-of-the-art techniques generate inconsistent color mapping (reddish houses and halo sky), our method leads to a visually plausible and artifact-free result. Images are best viewed in color and on the electronic version.

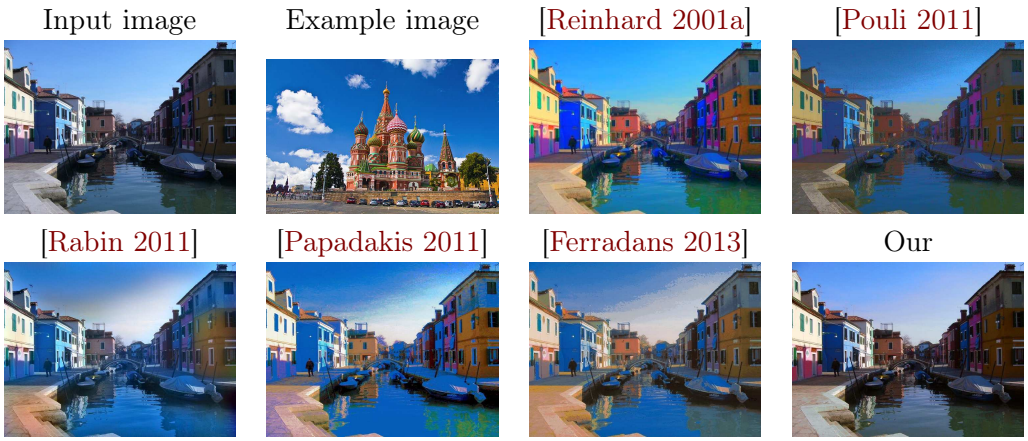


Figure 4.5: Results obtained by state-of-the-art color transfer techniques compared to our method for the Burano/moscow pair. While state-of-the-art techniques generate inconsistent color mapping (sky halo, incoherent color on the water), our method leads to a visually plausible and artifact-free result. Images are best viewed in color and on the electronic version.

Table 4.1: Comparison of the SSIM measure [Wang 2004] between input and output images for different color transfer methods, corresponding to figures 4.3, 4.4 and 4.5. A SSIM value of 1 denotes that no artifacts have been generated after color transfer. Our method creates no artifacts compared to other techniques.

	Fig. 4.3	Fig. 4.4	Fig. 4.5
[Reinhard 2001a]	0.98	0.84	0.83
[Pouli 2011]	0.87	0.56	0.76
[Rabin 2011]	0.96	0.94	0.91
[Papadakis 2011]	0.86	0.91	0.78
[Ferradans 2013]	0.68	0.87	0.80
Our	0.99	0.98	0.98

We claim that non plausible results lead to the introduction of new structures in the images. Thus, the perceptual metric Structural Similarity (SSIM) [Wang 2004] is used to assess the artifacts of color transfer, as already proposed in [Chiou 2010] and more recently in [Hwang 2014]. Since SSIM was employed with the goal of evaluating the capability of the method to produce an artifact-free result, we computed the SSIM between the luminances of the input and the output images, not taking the color into account. In Tab.4.1, we compare our method with state-of-the-art in terms of artifact generation. Results show that in all cases, our method was able to transfer the color palette while preserving the geometric structure of the image.

4.4.2 Comparison to Local Patch-based Color Transfer

In Fig. 4.6, we compare our results with the method of [HaCohen 2011], which performs color transfer with a transformation based on non-rigid patch correspondences

between the input and the example images. Both methods lead to reasonable results, and the objective comparison of methods is difficult. Although [HaCohen 2011] has better recovered the specularity of the dress, our result is less saturated (arguably more natural) on the skin, on the hair and on the background. The methods were compared on a larger set of images (results are visible in the supplementary material) and both methods produce comparable results. However, while [HaCohen 2011] assumes that the scenes are visually similar which is a very restrictive hypothesis, our framework is generic and can be used without any *a priori* of the scene. In particular, the method in [HaCohen 2011] is ineffective with the images of section 4.4.1. To sum up, our method is suitable for all types of images and in the case in which the images are very similar, our method is as good as the state-of-the-art method specifically tailored for this specific case.

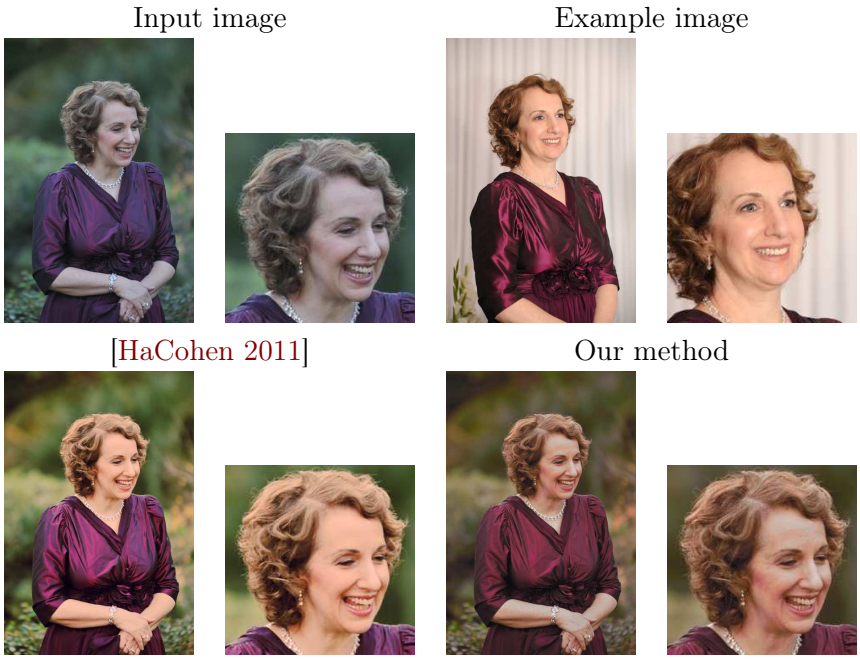


Figure 4.6: Results obtained with the color transfer technique of [HaCohen 2011] compared to our method. Although [HaCohen 2011] uses the restrictive hypothesis of spatial correspondences, we obtain a visually plausible result with a highly generic method.

4.4.3 Video Color Transfer

The color transfer method presented in this chapter can also be used for example-guided video color transfer. The extension of our technique to video is straightforward. We estimate the color transformation between the example image and a key frame of the video sequence and we encode it in a LUT (Lookup Table). In fact, the

LUT is the result of the $3D$ thin plate spline interpolation. Finally, we apply the LUT to all frames in the video sequence. Unfortunately, videos cannot be shown on this version and we invite the reader to look at the videos at the project website[3]. Results show that we obtain consistent color mappings without color flickering.

Figure 4.7: Video color transfer. Top row shows two possible example images. Second row, from left to right: the input frame chosen to perform the color transfer, and this frame recolored using the example images. For each example, the color transformation is computed and stored as a $3D$ Look-up Table (LUT). Bottoms rows show some video frames before and after color transfer using the LUT computed on the reference frame. The two corresponding videos can be seen in the supplementary material.

4.4.4 Constrained Color Transfer

Fig. 4.8 shows an experiment with a challenging test pair in terms of semantic color associations. Note that when color transfer is performed without saliency

[3] http://oriel.github.io/color_transfer

constraint, the result is semantically inconsistent. We have also tested state-of-the-art methods [Pitié 2007, Rabin 2011, Pouli 2011, Papadakis 2011, Ferradans 2013] and the color mapping was not semantically correct. On the contrary, when the saliency constraint is used, the birds in the images are detected as salient and their colors are matched accordingly. Finally, in Figure 4.9, we illustrate two cases of color transfer constrained by face detection.

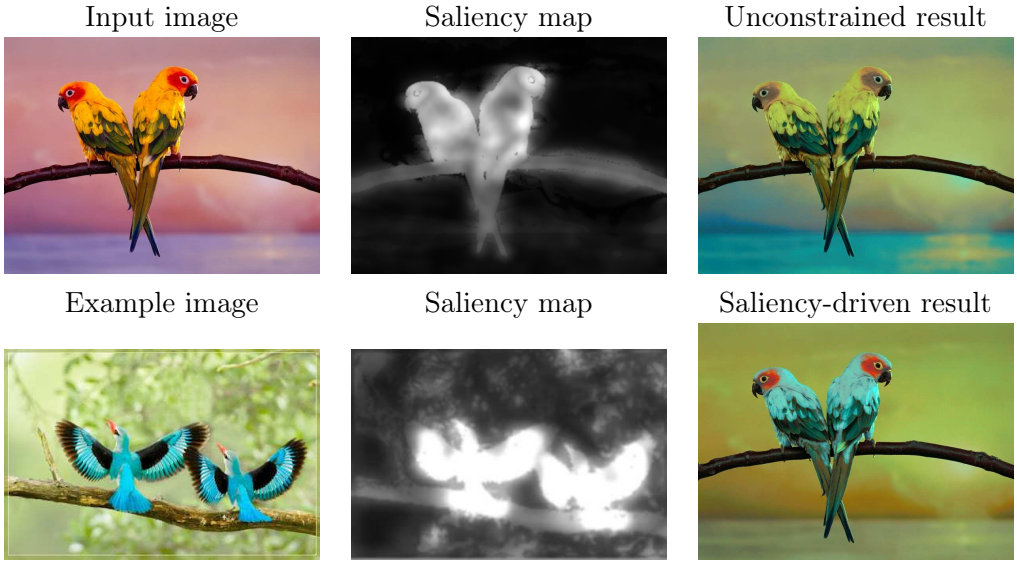


Figure 4.8: Illustration of saliency constraint: without the saliency constraint, the colors of the birds are not correctly transferred; while in the saliency-driven color transfer both the birds and the background are assigned to the expected colors.

4.5 Considerations

In this chapter, we have proposed a color transfer method that is based on global illuminant matching and optimal transport color transfer. The proposed color transfer method is automatic, does not require the input and example images to be visually similar and does not create visual artifacts as other state-of-the-art algorithms do. Our results present no visible artifacts since we limit changes in the luminance channel and regularize discontinuities in the color mapping through thin plate splines interpolation. The SSIM metric [Wang 2004] was used to objectively assess our method compared to other techniques.

We have also shown how semantic constraints can easily be considered in our framework and that our method can be applied successfully on video color transfer. The extension of color transfer to video was obtained through the application of

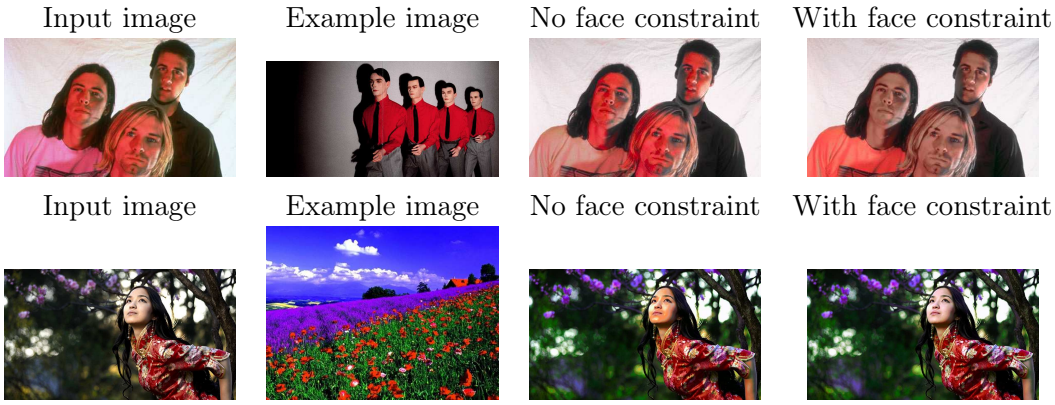


Figure 4.9: **Top part: Skin tones mapping**. From left to right: input and example images containing faces. The result without incorporating face constraint leads to undesirable reddish skin tones, while the result with the face constraint has mapped efficiently the skin tones. **Bottom part: Skin tones fidelity**. From left to right: input and example images. The result without incorporating face constraint leads to non plausible reddish skin tones, while the result with the face constraint has ensured skin tones fidelity.

a single color LUT to a sequence of images. Since color LUTs are widely used by colorists in film post-production, this approach for video color transfer has an important practical benefit.

This simple approach we presented in this chapter for video color transfer is overall effective and naturally does not produce temporal instabilities of color. In Chapter 5, we discuss the context where color instabilities are effectively observed in practice. As we will see, color instabilities are commonly produced by modern cameras, but tonal fluctuations can be efficiently addressed by performing example-based color correction.

Chapter 5

Tonal Stabilization

> *Since all models are wrong, the scientist cannot obtain a "correct" one by excessive elaboration. On the contrary, following William of Occam, he should seek an economical description of natural phenomena.*
>
> George E. P. Box, *Science and Statistics*, 1976

Contents

5.1 Introduction

In the previous chapter, we have introduced the problem of color transfer and an efficient technique that transfers the colors from an example image creating no artifacts. In this chapter, we may turn our attention to the question of tonal instability, which is one of the most common color artifacts found in modern videos.

Video tonal instability is a particular temporal artifact characterized by fluctuations in the colors of adjacent frames of a sequence. According to [Farbman 2011], in modern videos these instabilities are mainly caused by automatic settings of the camera, notably automatic white balance and automatic exposure.

Automatic white balance and automatic exposure are common features of consumer digital cameras, which are intended respectively to provide color balanced and well exposed images, while facilitating the user experience. However, these features are mostly appropriated for still images and are not stable in time, resulting in unpleasant tonal instabilities that can be perceived in videos. A notable problem with automatic white balance algorithms is their dependency on illuminant estimation, which is considered an ill-posed problem. Assumptions such as *grey world* and *max RGB* are easily violated in practice and in a context of temporal scene changes, it is likely to result in chromatic instability.

Automatic exposure, on the other hand, is a crucial feature of the camera to compensate the inherent limitations of dynamic range. However, fast exposure changes in a video footage can be unpleasant to the viewer, so a temporal smoothing of fast varying exposures can potentially enhance the perceived quality of the video. Stabilizing the exposure could also be useful for computer vision applications that rely on brightness constancy assumption (tracking, optical flow).

While automatic white balance can be simply turned off in some cases, low end cameras offer no control over setup parameters. In this case, the only alternative to avoid unpleasant tonal fluctuations is to further process the video. We note that few works in the literature have approached the problem of tonal instability in videos, and the existing solutions are limited to deal with specific types of brightness flicker or are not suited for a real time application.

In this chapter, we present a fast and parametric method to solve tonal stabilization in videos. Our main contribution is to model the tonal stabilization problem as an optimization that can be easily computed with a closed form solution. Moreover, we take dominant motion between frames into account, allowing our method to compute accurate color correspondences between temporally distant frames.

5.2 Related Work

In this section, we discuss a number of works and concepts which are related to the problem of tonal stabilization in videos. Generally speaking, tonal stabilization can be described as searching for the transformations that minimize undesired tonal variations in multiple images of a sequence. While surprisingly few works have specifically attempted to correct such color fluctuations in videos [Farbman 2011], [Wang 2014],

numerous works are motivated by similar purposes in different research communities.

Hence, we briefly review relevant works that are related to tonal stabilization. We start with radiometric calibration approaches, which takes into account operations performed in the camera pipeline in order to retrieve a physically based color calibration model between images.

Then, we present the color transfer approach, which provides color transformations between images without making explicit assumptions about the physics of the scene or of the camera. Finally, we review the literature in the problem of video tonal stabilization, from methods proposed to correct brightness flicker to more recent methods which deal with specific color instability.

5.2.1 Radiometric calibration

In theory, the radiometric calibration approach (presented in Chapter 3) would be the ideal solution to solve accurately the tonal stabilization problem, as it takes into account the complete camera color pipeline and retrieves the approximate irradiances from observed images. Nevertheless, the application of this approach to videos taken from consumer cameras poses several limitations. Note that to retrieve the camera response function, radiometric calibration needs registered images under multiple exposures. While some videos could effectively be seen as a composition of images under multiple exposures over time (such as time-lapse photography), this is not always the case. Also, a training set of RAW-sRGB image pairs is required to approximate the gamut mapping operator in the camera, but smartphone and point-and-shoot cameras usually do not provide the functionality of RAW images. Finally, the expensive estimation of camera responses is an additional difficulty to radiometric calibration of videos.

5.2.2 Color Transfer

As we have seen in Chapter 4, color transfer between images means modifying the colors of an *input* image according to the colors of an *example* image, while preserving the geometry of the original image. In principle, this goal looks similar to the one of radiometric calibration, where we normalize the colors of images to a common reference. The main difference is that color transfer methods make very few assumptions about the physics of the scene or the camera (scene illuminant or the camera response function are not explicitly estimated), being rather a general and pragmatic approach to estimate color transformations between images.

In this chapter, consider that tonal stabilization can be solved by local color transfer. Reminding that local color transfer assumes that there are spatial correspondences to be found between the input and example images, these correspondences being used to derive a color transformation [HaCohen 2011].

If this assumption reduces the scope of the method for general images, it is particularly relevant for specific user cases such as optimizing color consistency in photos from the same scene [HaCohen 2013], [Vazquez-Corral 2014]. In the case of

videos, we also expect to find numerous spatial correspondences between neighbor frames of the sequence. As we will see, the tonal stabilization algorithm presented in this chapter draws on a first raw motion estimation between frames to compute a global color correction.

In this sense, we will see that our approach for video tonal stabilization can be considered as derived from local color transfer. Moreover, similarly to general color transfer approaches, we avoid to make strong assumptions about the scene.

5.2.3 Video deflickering

While few works have attempted to correct specific color instability in videos, several methods were proposed to stabilize brightness fluctuations in gray scale videos, most of them targetting the stabilization of flicker artifact. Flickering is a high frequency variation in brightness of adjacent frames in a video, and is especially observed in old archived films. Physical flickering is caused by film degradation or aberration of exposure times. But flickering can also be observed in digital videos, when there are variations in exposure time or in the luminosity of the scene. One such example is in the case of time-lapse sequences, where each frame has a temporal sampling in the order of minutes and variations in luminosity between frames become highly noticeable.

Flickering can be either a global (spatially uniform) or a local (spatially variant) phenomenom, depending on the process that generated it. For the case of global flickering correction, [Decencière 1997] proposed a method assuming that the brightness degradation is given by an affine function

$$u_t = \alpha u_t^0 + \beta, \tag{5.1}$$

where u_t is the observed frame at time t, u_t^0 is the ideal flicker-free frame, α and β are coefficients estimated according to the assumption of mean and extreme brightness values preservation.

[van Roosmalen 1999] made a similar assumption of an affine model, but introducing a local operator

$$u_t(\boldsymbol{x}) = \alpha(\boldsymbol{x}) u_t^0(\boldsymbol{x}) + \beta(\boldsymbol{x}), \tag{5.2}$$

which varies spatially over pixel coordinates $\boldsymbol{x}$. The coefficients α and β are estimated through least squares fitting.

Affine models have the advantage of being simple and fast to estimate, but are limited to model linear brightness variations and are sensitive to the initial or reference frame. To deal with non-linear degradations, Naranjo et al [Naranjo 2000] proposed a method based on histogram specification. Each frame histogram is specified with respect to a target histogram computed as the average of neighbor frames, so that the deflicker operator is given by a monotonically increasing mapping

$$\tilde{u_t} = H_s^{-1} \circ H_t, \tag{5.3}$$

where H_t is the cumulative histogram of observed frame u_t, H_s is the target histogram of averaged neighbor frames u_s, with $s \in [t-i, t+i]$ and H_s^{-1} is approximated numerically by

$$H_s^{-1}(\alpha) = \inf\{\lambda | H_s(\lambda) \geq \alpha\}, \tag{5.4}$$

where λ is a grey level rank. Since u_t is smoothed over a set of symetric adjacent frames u_s, the method is less sensitive to the reference frame than previous affine models. Delon [Delon 2006] proposes a method relying on a similar rationale, but claiming that a direct average between neighbor histograms is not a satisfactory target distribution. The satisfactory intermediary target histogram is defined in a transport sense with the midway histogram matching [Delon 2004] between neighbor frames and performs a symmetric smoothing in a scale-time framework. Analogously to scale-space methods, the temporal smoothing is inspired by heat diffusion. In other words, the deflickering operator convolutes each rank function of the video by a gaussian kernel and is symmetric with respect to time. Further methods from [Delon 2010] and [Pitie 2004] extend the histogram matching deflicker to correct local flicker with spatially variant transformations.

Note that local flicker can be observed in old archived films, in which brightness fluctuations are mainly due to physical reasons. Hence, in this case, brightness fluctuations could vary along the image spatial coordinates. On the other hand, tonal instabilities observed in modern videos (fluctuations in camera exposure or white balance) tends to be global, under the hypothesis that camera response functions are global operators. Since we aim to correct tonal instabilities in consumer camera videos, we work under the assumption of global transformation.

We note the existence of several parametric and non-parametric methods to correct flicker artifacts in videos. However, we can observe that current deflickering methods are only suited for high frequency brightness instabilities and are not readily extended to correct color balance and exposure fluctuations.

5.2.4 Video Tonal Stabilization

The first method to deal with the video tonal stabilization problem was proposed by Farbman *et al* [Farbman 2011]. The method was proposed to compensate tonal fluctuations in digital videos caused by camera automatic settings, notably automatic white balance and automatic exposure. Then, the instability is corrected choosing one or more anchor frames, and aligning the colors of the adjacent frames with the chosen anchors.

For each frame u_t, they compute an adjustment map A_t that specifies a color adjustment for each pixel in order to obtain a desired aligned value. Then, an aligned sequence is obtained by applying each adjustment map to its frame. The goal of the method is to compute A_{t+1}, given u_t, A_t, and u_{t+1} in order to obtain a propagation of the adjustment maps along a frame sequence.

Correspondences between successive frames are computed without explicit motion compensation, by the claim that many pixels in the same spatial coordinates from two successive frames are likely to correspond to the same surfaces in the scene.

Tonal fluctuations in the luminance channel are approximated simply by a single shift parameter, where pixels belonging to the correspondence set are given by

$$R_{t/t+1} = \{\boldsymbol{x} | (L_t(\boldsymbol{x}) - \mu(L_t)) - (L_{t+1}(\boldsymbol{x}) - \mu(L_{t+1})| < \tau\}, \tag{5.5}$$

where $R_t/t+1$ denote the "robust set" of correspondences, L_t and L_{t+1} denote the luminance channel of the smoothed frames u_t and u_{t+1}, with $\mu(L)$ indicating the mean of the log luminance channel and τ is a threshold. All the remaining pixels, whose luminance changed by more than a threshold τ (the authors set empirically $\tau := 0.05$), are considered likely to have been affected by factors other than a change of parameters in the camera. For pixels belonging to the robust set ($\boldsymbol{x} \in R_{t/t+1}$), the adjustment map is computed as

$$\widehat{A}_{t+1(\boldsymbol{x})} = A_t(\boldsymbol{x}) + (u_t(\boldsymbol{x}) - u_{t+1}(\boldsymbol{x})) \tag{5.6}$$

For remaining pixels, a weighted interpolation is computed in order to achieve a global tonal transformation that is consistent for every pixel in a frame:

$$A_{t+1}(\boldsymbol{x}) = \frac{\sum_{i=1}^{N} w(\boldsymbol{x}, \boldsymbol{x}_i)\widehat{A}_{t+1}(\boldsymbol{x}_i)}{[\sum_{i=1}^{N} w(\boldsymbol{x}, \boldsymbol{x}_i)\chi_{\widehat{A}_{t+1}}(\boldsymbol{x}_i)] + \varepsilon}, \tag{5.7}$$

where $\boldsymbol{x}_i$ iterates over all pixel coordinates, $\chi_{\widehat{A}}$ is a characteristic function (one if $\widehat{A}(\boldsymbol{x}) \neq 0$; zero otherwise), $w \in \boldsymbol{W}$ is a pairwise Gaussian distance and ε is a constant close to zero. Since $\boldsymbol{W}$ is an $N \times N$ matrix, and N is large (the total number of pixels in a frame), Eq. 5.7 is extremely expensive to compute. To deal with this problem, the authors compute the interpolation using an eigenvector approximation of $\boldsymbol{W}$ through Nystrom method. Even though, the interpolation remains relatively costly and far from real time.

In summary, this method provides a reasonable solution to stabilize tonal fluctuations in videos, but at the cost of high space complexity (storing $\boldsymbol{W}$ in memory) and time complexity (interpolation step). Besides the high complexity, there are limitations in the method when we turn to the task of correcting tonal fluctuations in longer, noisy or fast motion sequences. The tonal correction performed with an adjustment map relying in anchor frames can bias all the frames to the colors of the anchors, potentionally resulting in the lost of original color dynamics. Also, the method to compute correspondences between adjacent frames in the method can be highly sensitive to noise or fast movements, and lack of accurate correspondences can result in flickering and error propagation.

A second approach, due to [Wang 2014], starts by estimating motion globally between successive frames, by relying on local features correspondences. A nine parameters affine color transformation is then used to model the exposure and white balance changes between two frames in the log domain. These affine color transformations are estimated by least squares for all neighboring frame pairs and accumulated to obtain a "color state" for each frame of the video, represented as a 4×4 matrix. To avoid data overfitting, a regularization term is added to force the affine

transformations to be close to the identity matrix. In a second step, the frame color state function is smoothed by minimizing an energy which also tends to control the deviation from the original color state. The regularization step necessitates to use PCA on the set of color states in order to avoid smoothing the different parameters of the affine matrices independently. While the results provided by the authors are visually good and much more satisfying than those of [Farbman 2011], the method is surprisingly complex to implement and requires to tune several parameters.

In the following sections, we are going to describe an alternative to perform color stabilization between frames approaching the main limitations seen in the literature. In particular, we are interested in responding adequately to three requisites: accuracy, robustness and simplicity.

5.2.4.1 Commercial tools

We are aware of existing commercial tools (Adobe After Effects, Final Cut Pro), which are able to correct specific brightness fluctuations, such as high frequency flickering commonly seen in time-lapse photography. Nevertheless, we verified limitations of these applications to correct general sequences containing tonal instabilities.

Color Stabilizer tool from Adobe After Effects and Flicker Free plugin from Final Cut Pro are based on color sampling of selected points in a reference frame, adjusting the colors neighbor frames so that the color values of selected points remain constant throughout the duration of the layer. The effect can be useful to remove flicker and equalize the exposure of footage, but it does not work if there is motion between frames.

To the best of our knowledge, existing commercial tools are based on the assumption of high-frequency flicker in nearly static sequences, with limitations to perform automatic tonal stabilization for sequences containing motion or middle frequency tonal instability.

5.3 Proposed method

In this section, we present the rationale and the main contributions of the proposed method for video tonal stabilization. First of all, we remark that our aim is to conceive a method that has the following desired properties:

1. Accuracy in modeling the color instabilities observed between frames in a video;
2. Robustness against motion, occlusion and noise;
3. Computational simplicity to be implemented in a near real time application.

We can observe that in practice the first property (model accuracy) is often in contradiction with the other properties of robustness and computational simplicity. Notably, in terms of tonal transformation, the radiometric calibration approach

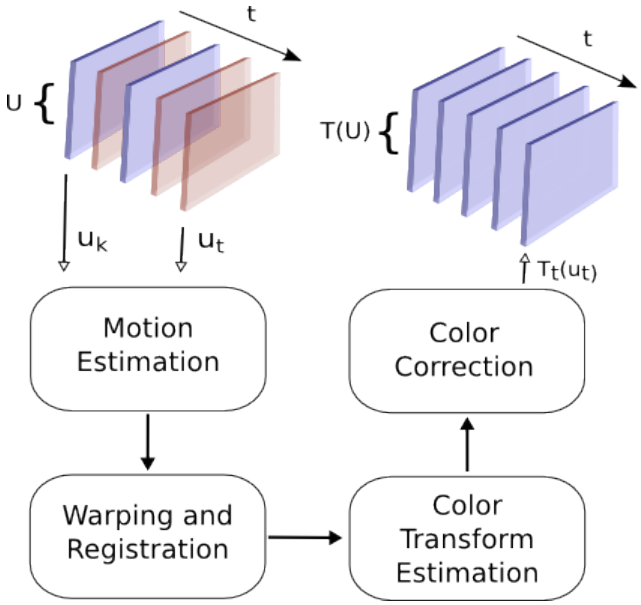


Figure 5.1: Illustration of main steps in proposed motion driven tonal stabilization.

[Kim 2012] which can be considered the most accurate model, is actually not robust against motion and occlusion, and is overly complex. Having this in mind, our method is proposed having as goal a good tradeoff between these three properties.

In addition, we note that the state-of-the-art tonal stabilization method from [Farbman 2011] do not meet the desired properties for tonal stabilization that were mentioned above. The main limitation of this method is to rely on spatial correspondences, however without applying motion compensation between the spatial coordinates of adjacent frames. Hence, the accuracy of spatial correspondences can be seriously compromised in case of fast motion between two frames.

In Figure 5.1, we present a general overview of the proposed method for tonal stabilization. In order to achieve robustness against motion and occlusion (an important limitation of [Farbman 2011]), we estimate the dominant motion between a reference keyframe u_k and the frame to be corrected u_t. Then, we register these two frames in order to compute color correspondences. Note that by means of cumulative motion, we are able to register u_t and u_k, even if they differ by several frames in time. Finally, the color correspondences are used to estimate a color transformation that is applied to correct the tonal instabilities.

The contributions of our method in comparison to state-of-the-art [Farbman 2011] can be summarized as the following:

1. Motion driven method: use of accurate color correspondences between frames obtained by dominant motion estimation and compensation.

2. Temporally longer tonal coherence, by using long term motion estimation obtained by motion accumulation.

3. Proposal of a computationally simple yet efficient parametric model for color correction.

For the application of the proposed algorithm, some assumptions need to be made regarding the sequence to be corrected and the color fluctuations we want to model. In particular, we assume that:

1. There are spatial correspondences (or redundance in content) between neighbor frames in the sequence (no scene cuts);

2. There is a global transformation which can compensate the colorimetric aberrations between the frames.

The first assumption is confirmed for every sequence composed of a single shot, as long as it does not pass through extreme variations of scene geometry (i.e: nearly total occlusion) or radiometry (i.e: huge changes in illumination or saturation). The second assumption implies that the observed color instability and consequently the camera response function are global (spatially invariant). In other words, the proposed method is not suitable for correction of local tonal instabilities such as local flicker observed in old archived films.

In the following subsections, we discuss in detail each main step (in Figure 5.1) of the proposed method. For the sake of simplicity, we start the discussion with the tonal transformation model, first assuming the simplest case of color correction between images without motion. In the sequence, we present our model to deal with the general case of tonal stabilization of sequences containing motion. Finally, we demonstrate the effectiveness of the method by experiments with real sequences and comparisons with the state-of-the-art.

5.3.1 Tonal transformation model

In this section we discuss the tonal transformation model used for correcting tonal instabilities. In particular, we consider the case of tonal instability observed in images taken with the same camera, so that tonal variations are caused specifically by the camera automatic parameters.

According to [Kim 2012], the complete color acquisition model is given by

$$\begin{bmatrix} u_R \\ u_G \\ u_B \end{bmatrix} = F\left(T_s T_w \begin{bmatrix} E_R \\ E_G \\ E_B \end{bmatrix}\right), \tag{5.8}$$

where $F : \mathbb{R}^3 \to \mathbb{R}^3$ denote the color camera response, T_s is a 3×3 matrix accounting for camera color space transform (constant over time), u is the observed intensity, E is the irradiance; T_w is a diagonal matrix accounting for changes in white balance and exposure (varying over time) and given by

$$T_w = \begin{bmatrix} \phi & 0 & 0 \\ 0 & \xi & 0 \\ 0 & 0 & \psi \end{bmatrix}. \tag{5.9}$$

Let u_0 and u_1 be two perfectly registered images taken by the same camera, differing only with respect to white balance and exposure (so that these images have identical irradiance E). Denoting $H = F(T_s)$ as the component of the camera response that is constant, then

$$u_0 = [u_{0_R}, u_{0_G}, u_{0_B}]^T = H([\phi_0 E_R, \xi_0 E_G, \psi_0 E_B]^T) \tag{5.10}$$

and

$$u_1 = [u_{1_R}, u_{1_G}, u_{1_B}]^T = H([\phi_1 E_R, \xi_1 E_G, \psi_1 E_B]^T). \tag{5.11}$$

Now, a simple approach to correct the tonal difference between u_0 and u_1 is to transform the colors of u_0 to have the same tonal characteristics of u_1, so that

$$H^{-1}(u_0) = \begin{bmatrix} \frac{\phi_0}{\phi_1} & 0 & 0 \\ 0 & \frac{\xi_0}{\xi_1} & 0 \\ 0 & 0 & \frac{\psi_0}{\psi_1} \end{bmatrix} H^{-1}(u_1) \tag{5.12}$$

$$u_0 = H\left(\begin{bmatrix} \frac{\phi_0}{\phi_1} & 0 & 0 \\ 0 & \frac{\xi_0}{\xi_1} & 0 \\ 0 & 0 & \frac{\psi_0}{\psi_1} \end{bmatrix} H^{-1}(u_1) \right). \tag{5.13}$$

Hence, in theory we can achieve tonal stabilization between images u_0 and u_1 with a simple diagonal transformation performed in the camera sensor space (given by non-linear transformations H and H^{-1}). This tonal stabilization model is inspired by radiometric calibration [Kim 2012] and [Xiong 2012] as an accurate procedure to perform camera color transfer when irradiances $E = [E_R, E_G, E_B]$ are known in the form of RAW images, allowing an estimate of H. However, for the problem of tonal stabilization, we are faced with videos taken with low-cost cameras, from which we cannot make the usual assumptions that are necessary to compute radiometric calibration. The assumption of multiple exposures from the same scene, which is required to estimate the camera response function may not be valid for some sequences, and RAW-sRGB correspondences are also not available in practice.

Reminding the desired properties we have listed for video tonal stabilization, we could say that while accurate, the radiometric calibration model is overly complex and not general to be applied for tonal stabilization of sequences from which we do not know the irradiances. The question we want to answer is: how could we approximate this model, when the only information we have are the intensities observed in u_0 and u_1?

While the observed images do not provide enough information to derive the exact color transformation that normalize their tonal characteristics, we claim in this chapter that an effective solution for this problem comes from a tonal intensity mapping (such as a brightness or color transfer function), which can be computed by parametric or non-parametric estimation methods. In the sequence, we discuss the pros and cons of each estimation approach and provide the motivation for our choice.

5.3.1.1 Non-parametric or parametric color transformation

Non-parametric color transformation models do not make explicit assumptions on the type of transformation, allowing to model non-linear transformations, but at the risk of lack of regularity that would demand post-processing regularization.

Some notable examples of non-parametric color transformations are weighted interpolation and histogram specification. As we have previously discussed, a weighted interpolation such as suggested in Farbman's tonal stabilization method [Farbman 2011] has the drawback of being computationally complex both in terms of memory requirements and processing time. We note that a color interpolation such as proposed by [Farbman 2011] is in fact a global transformation that is similar to a histogram specification, the main difference being that the interpolation is computed from spatial correspondences, while the histogram specification is computed from intensity cumulative histograms.

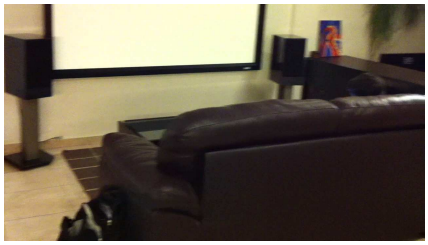
Classical histogram specification could be an efficient alternative to solve the problem of tonal stabilization (channel-wise specication requires only $O(n \log n)$ computations, where n is the number of pixels in an image). However, there are well known limitations of histogram specification. Indeed, it can lead to contrast stretching and quantization artifacts that would need post-processing [Rabin 2010], and range extrapolation of the transformation is not always possible, in special when dealing with color. Take for example the transformations illustrated in Figure 5.2, where we have channel-wise histogram specification. Note that the red and blue transformation curves in Figure 5.2 are affected by sudden jumps, which turns out to produce strong artifacts in the resulting image after transformation.

On the other hand, parametric models assume that the transformation can be modelled by a given function (linear, affine, polynomial, etc), so the problem is solved by estimating the coefficients of the transformation. While not very flexible to model any form of transformation, parametric models have the important advantage of being expressed by smooth and regular functions, well defined for the whole color range, so that extrapolation is not a problem. Furthermore, since the transformation is described by few parameters, it reduces the risk of oscillation in time.

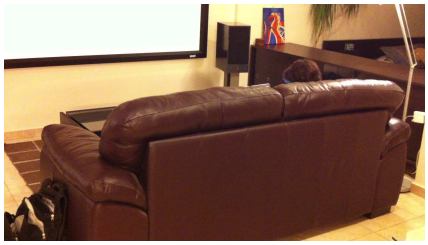
We note that most white balance algorithms implemented in digital cameras adjust the channel scaling with a simple parametric model, which is a Von Kries diagonal transformation[1] performed in RAW images [Kim 2012]. However, as we have seen in the discussion of tonal transformation model, a diagonal model applied to sRGB images is not able to model non-linearities inherent to the camera response.

We want to stress that we do not have enough information to derive the exact tonal transformation model for color stabilization - be it a parametric or non-parametric transformation. Hence, we search for a tonal transformation model that is simple enough to be fastly computed, and accurate enough to produce a visually pleasant tonal stabilized sequence. After performing experiments with different parametric and non-parametric models (histogram specification, splines interpola-

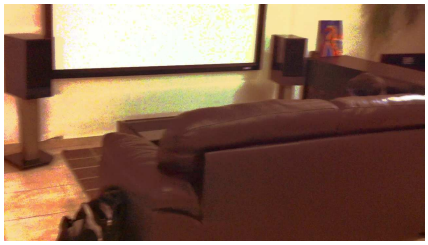
[1] In practice, some camera white balance algorithms compensate only the red and blue channels, leaving the green channel untouched.



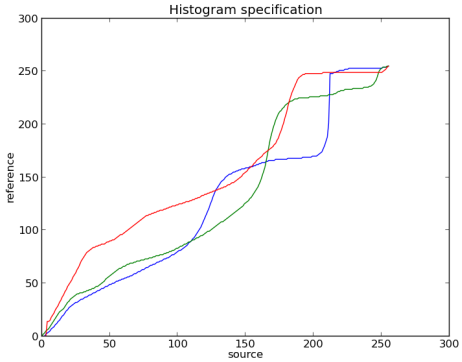
Source image



Reference image



Result: Source specified to reference



Histogram specification curves

Figure 5.2: An illustration of channel-wise histogram specification mapping and results. Note that the mapping is irregular with sudden jumps, which results in strong artifacts in the transformed image.

tion, piece-wise linear function, diagonal model) the power law transformation has shown to best fit the criteria mentioned above.

5.3.1.2 Power law color transformation

For the sake of simplicity, we discuss in this subsection the proposed tonal transformation model assuming that we want to correct tonal instability in sequences containing no motion. The general case of sequences containing motion is approached in Sec. 5.3.1.2 and Sec. 5.3.2.

Our assumption to correct non-linear tonal instabilities is that exposure differences between frames can be approximated by an exponential factor, while white balance correction can be approximated by diagonal color re-scaling. We observed that a parametric power law model is successful in jointly meeting these assumptions. Formally, let u_k be a reference image and u_t an image to be corrected, assuming the images are perfectly registered, a power law relationship between u_t and u_k is written as a function of the form

$$u_k(\boldsymbol{x}, c) = T(u_t) = \alpha_c u_t(\boldsymbol{x}, c)^{\gamma_c}, \tag{5.14}$$

where $c = \{R, G, B\}$ denotes the image color channels, $\boldsymbol{x} \in \Omega$ denotes the spatial coordinates over the domain $\Omega \subseteq \mathbb{R}^2$. Our problem now is to estimate the optimal

coefficients α_c, γ_c such that we minimize the mean square error

$$\arg\min_{\alpha_c,\gamma_c} \sum_{\boldsymbol{x}\in\Omega} (u_k(\boldsymbol{x},c) - \alpha_c u_t(\boldsymbol{x},c)^{\gamma_c})^2. \tag{5.15}$$

The minimization of non-linear Eq. 5.15 over α_c and γ_c has not an analytical solution. However a possible approximation consist in taking the logarithm in Eq. 5.14 to have

$$\log u_k(\boldsymbol{x},c) = \gamma_c \log u_t(\boldsymbol{x},c) + \log \alpha_c \tag{5.16}$$

so that it can be solved by least squares fitting as an affine function defined in the logarithmic domain:

$$\arg\min_{\hat{\alpha_c},\gamma_c} \underbrace{\sum_{\boldsymbol{x}\in\Omega} (\log u_k(\boldsymbol{x},c) - (\gamma_c \log u_t(\boldsymbol{x},c) + \hat{\alpha_c}))^2}_{E} \tag{5.17}$$

where $\hat{\alpha_c} = \log \alpha_c$. Now, we can solve Eq. 5.17 by setting

$$\frac{\partial E}{\partial \hat{\alpha_c}} = \frac{\partial E}{\partial \gamma_c} = 0 \tag{5.18}$$

to derive the well known analytical solution to univariate linear regression:

$$\gamma_c = \frac{Cov\,(\log u_t, \log u_k)}{Var\,(\log u_t)}, \tag{5.19}$$

$$\hat{\alpha_c} = \overline{\log u}_{k(c)} - \gamma_c \overline{\log u}_{t(c)}, \tag{5.20}$$

$$\alpha_c = \exp(\hat{\alpha_c}), \tag{5.21}$$

where $\overline{u}$ denotes the empirical mean of image u. This solution to obtain the coefficients α_c and γ_c has some desirable properties: it is computationally simple and exact, guaranteed to converge in $O(n)$ iterations (linear in the number of n correspondent points, $n = \sharp\Omega$). As a remark, we note that minimizing Eq. 5.15 is evidently not equivalent to minimize 5.17. We know that when fitting an affine function in the logarithmic domain, the loss function E also becomes logarithmic, meaning that residuals computed from low values will tend to have more weight than residuals computed from high values. For our application of color correction, this imply that the estimation can be specially sensitive to the presence of outliers in dark colors. Even though the analytical solution is fast and exact (for non-linear error), for higher regression accuracy in terms of linear mean squared error, the solution can be alternatively computed with a numerical method such as gradient descent [Levenberg 1944, Marquardt 1963].

It should be noted that the model we propose in Eq. 5.14 is quite similar to the one proposed in the work of color stabilization in [Vazquez-Corral 2014], where the tonal transformation is also modeled as a combination of a linear term and a power term. In particular, the tonal tranformation model proposed by [Vazquez-Corral 2014] is given by

$$T(u(\boldsymbol{x})) := (\boldsymbol{M}u(\boldsymbol{x})^{\gamma_1})^{\frac{1}{\gamma_2}}, \tag{5.22}$$

where $u(\boldsymbol{x})$ is a color intensity, $\boldsymbol{M}$ is a 3×3 matrix and γ_1, γ_2 are the estimated gamma correction parameters from the original and reference images. Differently to [Vazquez-Corral 2014], we do not consider the γ_c coefficient in our model to be equivalent to the parameter γ in the gamma correction of the camera imaging pipeline [Kim 2012]. We rather assume that the power γ_c in our model approximates possible non-linear color changes between the images of a sequence.

Another difference between the two models is that [Vazquez-Corral 2014] includes a full 3×3 matrix in the tonal transformation, while our model is separable over color channels. Therefore, the model of [Vazquez-Corral 2014] can take into account channel correlations and possible color space conversions that can take place when correcting images taken from different cameras. Nevertheless, in our case we consider sequences taken entirely with the same camera, and as we show in our experiments, our tonal transformation model is effective in practice with straightforward optimization in comparison to the model proposed in [Vazquez-Corral 2014].

Finally, we note that a power law model for color transformation is commonly used for color grading in film post-production. The ASC CDL[2] (American Society of Cinematographers Color Decision List) is a format for exchange of color grading parameters between equipment and software from different manufacturers. The format is defined by three parameters slope (α), offset (β) and power (γ), which are independently applied for each color channel:

$$T(u) = (\alpha u + \beta)^{\gamma} \ . \tag{5.23}$$

This transformation is usually applied in a color space specific to the color grading software (for example YRGB color space in DaVinci Resolve[3]). Comparing to ASC CDL, our parametric model is similarly based on power and slope coefficients, without offset, which in advantage allow us to compute the optimal parameters with an analytical expression.

5.3.2 Motion and temporal coherence model

While we started our discussion with the assumption of perfectly registered images, it is evident that in practice, movement is observed in the majority of sequences. Motion estimation is employed in this chapter to guarantee tonal stabilization by taking into account the movement not only between a pair of frames, but also between several frames in a sequence.

There is an extensive literature on motion estimation methods, some examples are dominant global motion estimation [Odobez 1995], dense optical flow [Black 1996], sparse feature tracking [Shi 1993]. We claim that for the present task of estimating tonal transformations driven by motion based correspondences, it is desirable to

[2] http://en.wikipedia.org/wiki/ASC_CDL
[3] https://www.blackmagicdesign.com/products/davinciresolve/color

have a dense set of correspondences, so that we take advantage of correspondences between homogeneous intensity areas to estimate accurate color transformations.

In particular, we rely on dominant motion estimation between frames, mostly motivated by a tradeoff: dominant motion is computationally simpler (potentially computed in real time) in comparison to dense optical flow such as [Liu 2008]. However, dominant motion does not provide pixel-wise accuracy. We nevertheless note that dominant motion usually accounts for camera motion, and in our experience, tonal instabilities seen in videos are normally correlated with the movement of the camera. In contrast to tasks that depend heavily on accurate motion (i.e. video motion stabilization), we do not need a highly accurate motion description in order to estimate a color transformation that compensates tonal differences between frames.

Denoting $u_t : \Omega \to \mathbb{R}^3$ and $u_k : \Omega \to \mathbb{R}^3$ as two neighbour frames in a sequence such that $t = k + 1$, we can assume that u_t and u_k depict the same scene, differing only by a small spatial displacement. Then, the 2D motion between these frames can be described by a global transformation A, such that $u_k(\widehat{\Omega})$ and $u_t(A(\Omega))$ denotes the registration (motion compensated alignment) of u_k and u_t, where $\widehat{\Omega}_k \subseteq \Omega_k$ is a subset of spatial coordinates in u_k. More specifically, we represent A as a matrix that accounts for affine warping, which can be considered a good tradeoff between complexity and representativeness, taking into account scale, translation and rotation transformations between frames. Then we have

$$A(\boldsymbol{x}) = \begin{bmatrix} u(\boldsymbol{x}) \\ v(\boldsymbol{x}) \end{bmatrix} \tag{5.24}$$

$$\begin{aligned} u(\boldsymbol{x}) &= a_1 + a_2 x_1 + a_3 x_2 \\ v(\boldsymbol{x}) &= a_4 + a_5 x_1 + a_6 x_2, \end{aligned} \tag{5.25}$$

where $\boldsymbol{x} = (x_1, x_2)$ denotes the original pixel coordinates, $A(\boldsymbol{x})$ is the affine flow vector modeled at point $\boldsymbol{x}$ and $(a_{1,\ldots,6})$ are the estimated motion coefficients. We estimate the coefficients based on robust parametric motion estimation method from Odobez and Bouthemy [Odobez 1995]. Their method computes the optimal affine motion coefficients in terms of spatio-temporal gradients by Iteratively Reweighted Least Squares (IRLS) with M-estimator loss function (Tukey's biweight). Such loss function is known to be more robust against motion outliers than usual quadratic error. Their method also takes into account a brightness offset as a simple way of relaxing the brightness constancy assumption (which states that pixel intensities from the same object do not change over time) to deal with minor changes in scene illumination.

We have considered the case of motion estimation between neighbor frames u_t and u_k, for $t = k + 1$, but we want to generalize the approach for the case of an arbitrary k differing to t by several frames. In particular, for video tonal stabilization, we would like to take advantage of motion estimation between several frames in order to guarantee longer tonal coherence. However, long term motion estimation is a challenging problem and methods based on spatio-temporal gradients cannot deal with direct large motion estimation between frames. An usual workaround to deal with larger displacement is to estimate motion from multiple image resolutions, but

even though, the multiresolution estimation is inaccurate in estimating large motion between several frames. In other words, reliable parametric dominant motion estimation based on [Odobez 1995] is limited only for interframe motion.

In practice, a simple cumulation of interframe motions is used as an approximation of long term motion, which can be used for the estimation of tonal transformations. Formally, assuming $t >> k$ (the keyframe is in the "past") and $s = (t-k)-1$ being the temporal scale for which the scenes in u_t and u_k are overlapped, the acummulated affine motion from u_t to u_k is given by

$$A_{t,k} = A_{t,t-1} \circ A_{t-1,t-2} \circ ... \circ A_{t-s,k}, \tag{5.26}$$

where $A_{t,k}$ denotes the motion coefficients estimated from frame u_t to u_k. Having an estimate of $A_{t,k}$, we can warp u_t to u_k in order to get a registered pair of images with known motion compensated correspondent points, which are defined by

$$\Omega_{t,k} = \{(\boldsymbol{x}, \boldsymbol{y}) \mid \boldsymbol{x} \in A_{t,k}(\Omega_t), \boldsymbol{y} \in \widehat{\Omega}_k\}, \tag{5.27}$$

where $\widehat{\Omega}_k \subseteq \Omega_k$. Nevertheless, it must be reminded that the motion estimation is a rough global approximation and is likely to contain errors due to occlusions, non-dominant (object) motion, or simply innacurate coefficients in $A_{t,k}$. Hence, we need to discard motion outliers to guarantee an accurate color transformation. One approach for that is to compute a difference map between the aligned images, and consider that values higher than a threshold on this difference map will correspond to the outliers. However, the difference map being based on the residual of intensity values, we should note that intensity differences are not reliable under brightness and color changes between frames.

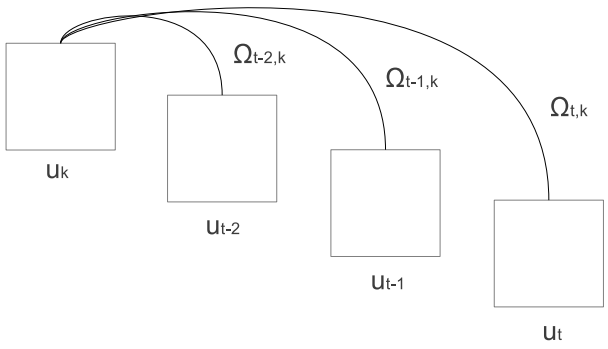


Figure 5.3: Warping and correction based on keyframe u_k. In this model, for each frame to be corrected u_t, an affine warping gives a set of spatial correspondences namely $\Omega_{t,k}$, from each a colorimetric transformation is estimated.

Thus, we first compute a rough radiometric compensation of tonal differences between the aligned images as a measure to reduce the risk of confusing motion outliers with tonal differences. In the sequence, we can compute a difference map

from the corrected warped frame, which will discard the motion outliers and keep the colorimetric differences - which are essential to estimate the color transformation.

Formally, the outlier removal approach can be summarized as the following. Let $\widehat{\Omega}_{t,k}$ be the set of correspondent spatial coordinates (motion overlap) shared between two frames u_t and u_k. $\widehat{\Omega}_{t,k}$ is computed by accumulating frame to frame motions - we warp u_t to align it to the keyframe u_k in order to have $u_k(\widehat{\Omega}_k)$ registered to $u_t(A_{t,k}(\Omega_t))$. Since $\widehat{\Omega}_{t,k}$ contains motion outliers, we will reject outlier data, but first we account for possible tonal differences between the aligned frames, so that these differences are not taken as outliers. Given $(\boldsymbol{x}, \boldsymbol{y}) \in \widehat{\Omega}_{t,k}$, the tonal differences between aligned current frame and keyframe are compensated by a simple mean value shift:

$$\tilde{u}_t(\boldsymbol{x}, c) = (u_t(\boldsymbol{x}, c) - \mu(u_t(c)) + \mu(u_k(c)) \tag{5.28}$$

Finally, we will have a set of corresponding spatial coordinates filtered by motion outliers, defined by

$$\Omega_{t,k} = \{(\boldsymbol{x}, \boldsymbol{y}) \in \widehat{\Omega}_{t,k} \mid \frac{1}{3} \sum_{c \in C} [u_k(\boldsymbol{y}, c) \ - \tilde{u}_t(\boldsymbol{x}, c)]^2 < \sigma\}, \tag{5.29}$$

where σ is the empirical noise, which can be an estimation (with a noise estimation method such as [Colom 2013]) or an approximation of the noise variance in u_t and u_k.

Based on the set of spatial correspondences between temporally distant frames $\Omega_{t,k}$, we are able to estimate temporally coherent tonal transformations, so that we can compensate tonal instabilities. By taking long term motion into account, we enforce that tonal coherency is not lost from frame to frame.

5.3.3 Motion driven tonal stabilization

We can start the discussion on motion driven tonal stabilization by first considering an ideal symmetric operator. This transformation has the desired property of being invariant with respect to the time direction, avoiding bias to the colors of the keyframe. This definition leads us to a symmetric scale-time correction similar to the operator proposed by [Delon 2006]:

$$S_t(u_t) = \sum_{i=-s}^{s} \lambda_i T_i(u_t), \tag{5.30}$$

where s is the temporal scale of the correction, T_i is a tonal transformation weighted by λ_i, assuming that λ_i is a gaussian weighting intended to give more importance to transformations estimated from frames that are temporally close to u_t. This operator can be seen as a *temporal smoothing* which computes the tonal stabilization of u_t as a combination of several weighted transformations. In practice, the S_t operator requires the estimation of $2s$ transformations for every frame to be corrected, which is computationally expensive, and even if s is set to be small, the correction then risks to be not sufficiently effective. This approach fits well for high frequency

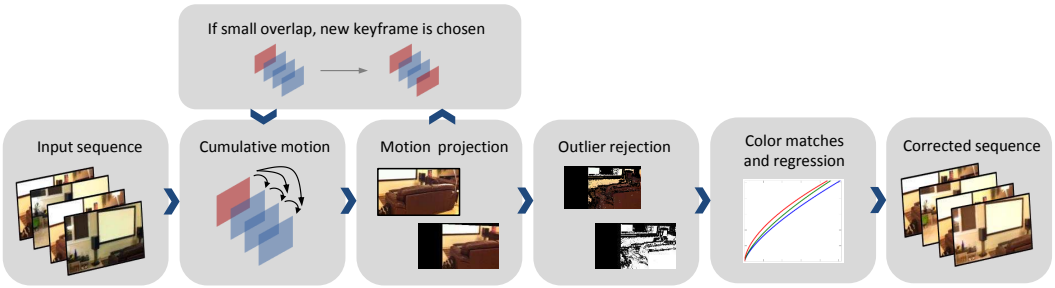


Figure 5.4: Flow chart of proposed motion driven tonal stabilization algorithm.

flickering stabilization, because flicker can be filtered with an operator defined for a limited temporal scale. On the other hand, tonal fluctuations caused by camera parameters could need larger temporal scales to be properly corrected.

We can nevertheless propose a faster and yet efficient alternative to operator S_t where less computations are required to correct each frame. In particular, we also want to control undesired estimation bias and drift throught weighted transformations. For the sake of simplicity, we can assume that the starting point for the sequential tonal stabilization is the first frame of the sequence, then the solution for tonal stabilization can be seen as a *temporal prediction*, where we predict the correct tonal appearance of u_t based on previously known tonal states. This is typically the case of an application for sequential on-the-fly correction, for instance to compensate tonal fluctuations of a live camera in a video conference. Even for sequential tonal stabilization, the symmetric property can be approximated by combining forward and backward corrections.

In Algorithm 1, we present the proposed sequential motion driven tonal stabilization. For each frame u_t, we want to find a triplet of RGB transformations defined as $T_t(u_t)$ which minimizes the tonal differences between u_t and u_k. Let $M(u_t, u_k)$ denote a function that takes two frames as parameters, computes their motion estimation, warping and outlier rejection, producing as output reliable spatial correspondences. So, the tonal transformation is based on a regression over the set of data points given by the coordinates $\Omega_{t,k} = M(u_t, u_k)$.

The steps illustrated in Figure 5.4 (motion estimation, warping, color transform estimation, color correction) are repeated for all the following u_{t+m} frames, until $\sharp\Omega_{t,k} < \omega \times n$, where $\sharp\Omega_{t,k}$ denotes the cardinality of the corresponding set. When this condition is met, it means that the cardinality of the overlapped region between u_t and u_k is no longer large enough to allow for an accurate color estimation. In this case, the keyframe u_k is updated to u_{t-1}.

In contrast to Farbman's method [Farbman 2011], which propagate transformations from frame to frame, our method guarantees longer tonal coherency between the temporal neighborhood of a keyframe. In other words, we propagate the tonal transformations from keyframe to keyframe, so that the accumulation of tonal error is controlled by using a larger temporal scale

An important aspect of the video tonal stabilization problem is that complete

Algorithm 1 Motion driven tonal stabilization

Input: Sequence of frames $u_t \in U, t = \{1, ..., D\}$
Output: Tonal stabilized sequence $T_t(u_t), t = \{1, ..., D\}$

$k \Leftarrow 1$ # Initialize keyframe index
$t \Leftarrow k + 1$ # Initialize current index
$T_1(u_1) = u_1$ # First output frame is not transformed
while $t \leq D$ **do**
  $\Omega_{t,k} \Leftarrow M(u_t, u_k)$ # Compute motion based correspondences
  if $\sharp\Omega_{t,k} \geq \omega \times n$ **then** # If there are enough correspondences:
    for $c \Leftarrow \{R, G, B\}$ **do** # Perform tonal correction
      $\alpha_c, \gamma_c \Leftarrow$ min. Eq. 5.15
      $\widehat{u}_{t(c)} \Leftarrow \alpha_c {u_{t(c)}}^{\gamma_c}$
      $T_t(u_{t(c)}) \Leftarrow \lambda \widehat{u}_{t(c)} + (1 - \lambda) u_{t(c)}$
    end for
    $t \Leftarrow t + 1$
  else # If there are not enough correspondences:
    if $k < t - 1$ **then**
      $k \Leftarrow t - 1$ # Update keyframe
      $u_k \Leftarrow T_{t-1}(u_{t-1})$
    else
      $T_t(u_t) \Leftarrow T_{t-1}(u_t)$
      $t \Leftarrow t + 1$
      $k \Leftarrow t + 1$
    end if
  end if
end while

temporal preservation of tonal appearance is not always desired, due to the camera inherent dynamic range limitations. In fact, tonal instabilities caused by camera automatic exposure can be perceptually disturbing, but if huge changes occurs in camera exposure, the variation of tonal appearance should be kept to some degree, so that we avoid overexposure. In order to deal with this aspect, we can perform temporally weighted color transformations, or aditionally we can increase the dynamic range of the sequence in time.

5.3.4 Temporal weighting

As a regularization concern, we can ensure that the transformation $T_t(u_t)$ does not deviate largely from the original content of u_t by applying a weight λ:

$$T_t(u_{t_{(c)}}) = \lambda(\alpha_c(u_{t_{(c)}}^{\gamma_c})) + (1 - \lambda)u_{t_{(c)}}. \tag{5.31}$$

A similar weighted correction is used in [van Roosmalen 1999] where it is proposed to fix $\lambda := 0.85$ as a forgetting factor for recursive deflickering. We claim

that the weighting λ could vary over time, in function of the temporal distance or in function of the motion between u_t and u_k, assuming that frames that are closer in content to the keyframe should receive higher weight in the tonal correction. Since we know the affine motion parameters $A_{t,k}$ that warps u_t to u_k, we can actually compute a rough spatial distance from these two frames and write

$$\lambda = \exp(-\lambda_0 \frac{||V_{uk}||}{p}), \tag{5.32}$$

where $||V_{uk}||$ denotes the norm of the dominant motion vector V_{uk}, p is the maximum spatial displacement (number of rows + number of columns in the image), λ_0 is the exponential decay rate (in practice we set $\lambda_0 := 0.5$). Another possibility is to weigth the correction in function of the temporal distance between u_t and u_k:

$$\lambda = \exp(-\lambda_0 \frac{|t-k|}{D}), \tag{5.33}$$

where D is the video duration (number of frames in the sequence). In this sense, the idea is to decrease the influence of frames which have large motion displacement from the current frame. A remark of interest, is that work done by [Hirai 2010a, Hirai 2010b] in the field of color perception have shown that chromatic and contrast sensitivity functions decreases exponentially when the velocity of stimuli increases. Therefore, we could claim that the motion dependent λ has, to some degree, a perceptual motivation.

5.3.5 Temporal dynamic range

In our experiments we observed that without any temporal weighting, we can guarantee strict tonal stabilization throughout the entire sequence, no matter if strong luminance changes occur. The result is visually pleasant for sequences in which luminance variation is smooth, however, when correcting sequences with significant changes in exposition (ex.: from very dark to very bright environments), we observed saturation and clipping in the final result. In order to deal with this problem, we can work with a higher dynamic range, so that we do not have to clip color intensities larger than $2^8 - 1 = 255$ (maximum intensity value for each color channel in 8 bits images).

We can allow larger intensities by working with 16 bits images, so that intensities larger than 255 do not need to be clipped after color transformation. Then, we have as result a sequence that has an increased dynamic range over time, and the sequence could actually be visualized without losing intensity information in an appropriated high dynamic range display.

However, in practice, we need to convert the sequence back to 8 bits in order to display it in standard low range displays. Instead of clipping all the intensities which extrapolate the limit, we can alternatively apply a tone mapping operator of choice to render a low dynamic range image. In particular, we have made experiments with a logarithmic tone map operator. Given an intensity value i, and the maximum intensity value of the whole sequence z, a log tone mapping operator m is given by

$$m(i) = 255 \left(\frac{log(1 + \frac{i}{z})}{log(2)} \right). \tag{5.34}$$

In Figure 5.5 we illustrate the potential problem of intensity clipping when applying tonal stabilization and the effects of attenuating it with a temporal tone map operator or with a temporal weighting.

5.3.6 Additional implementation details

In order to reduce the influence of noise outliers in the estimation of tonal transformation, smoothing is applied to u_t and u_k. Note that this step is not necessary for well exposed sequences where tonal instability is mainly due to white balance fluctuations, nevertheless smoothing is recommended when working with sequences strongly affected by noise.

We observed that more computationally expensive smoothing approaches (such as bilateral filtering) do not produce noticeable differences. Thus, we rather perform smoothing through subsampling. We downscale the original frames to (120 pixels wide) for both motion estimation and color transform estimation, which in turn do not produce noticeable loss in tonal stabilization accuracy. Furthermore, instead of applying the power law color transformation to correct the full original frame composed of N pixels, we build one lookup table (LUT) per color channel, and then we compute the power law independently for each LUT. This reduces the number of power law computations from $3 \times N$ (more than 16 million for 4k video resolution) to only $3 \times 256 = 768$.

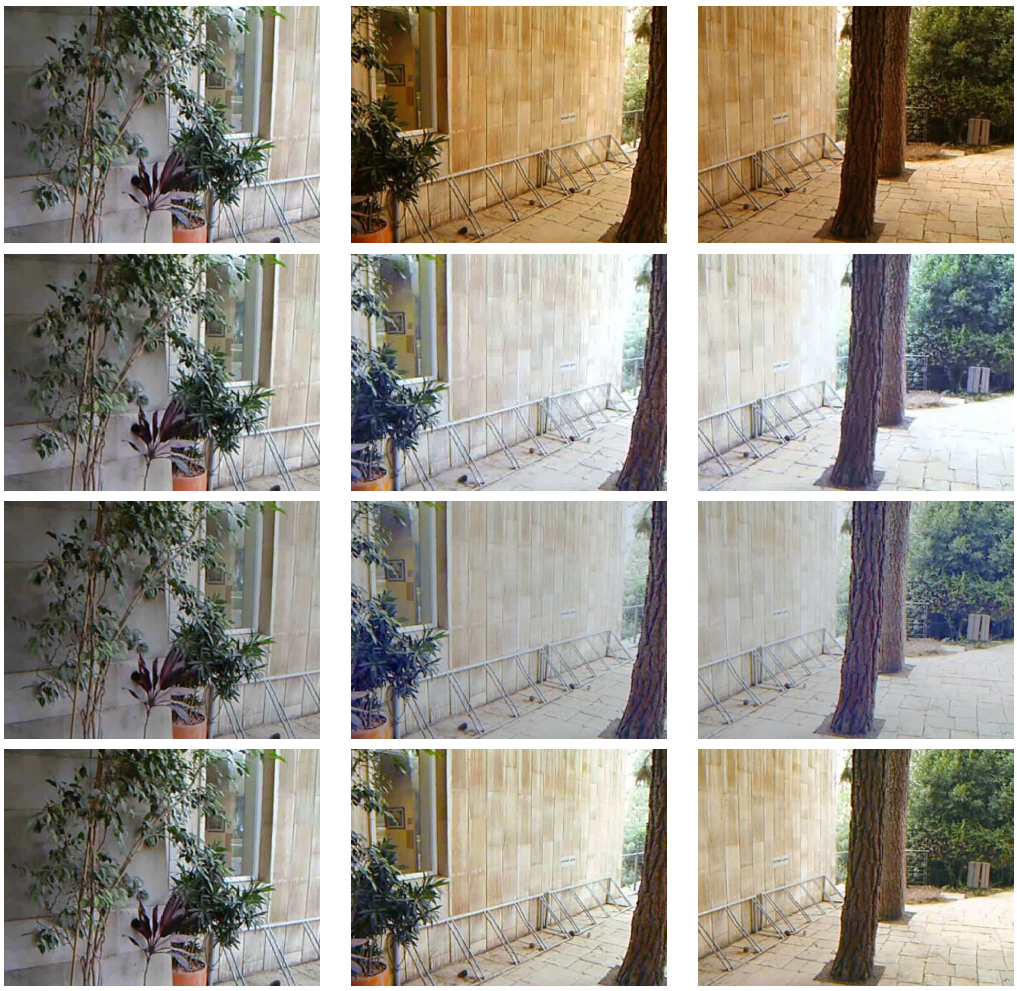


Figure 5.5: Illustration of tonal stabilization in sequence "entrance". **First row**: Frames extracted from the original sequence. **Second row**: stabilizing with $\lambda := 1$ (no temporal weighting) and without tonemap ensures tonal coherence between all the frames, but at the cost of clipping intensities. **Third row**: Tonal stabilization with a temporal tone mapping, where tonal appearance is preserved and clipping do not occurs, however, the sequence is overall darker. **Bottom row**: Tonal stabilization with temporal weighting (λ decreases exponentially in function of motion). While temporal weighting reduces the strict tonal preservation, it may be argued that it produces a visually pleasant result by preserving some degree of the original colors.

5.4 Results and discussion

In this section we study some properties of the proposed tonal stabilization method and we experiment the algorithm with sequences containing real tonal fluctuations.

First, we study the goodness of fit of the proposed power law model, and we compare our model to the state-of-the art parametric tonal transformation proposed by [Vazquez-Corral 2014]. In the sequence, we present a discussion on the influence of parameter settings in the tonal stabilization results.

Then, we show the experimental results obtained with our tonal stabilization algorithm and we compare them with state-of-the-art results. On the one hand, qualitative evaluation based on visual inspection is performed and, on the other hand, quantitative results measuring the amount of tonal variation in the resulting sequence and the fidelity to the original sequence are provided. Both, quantitative and qualitative evaluation prove that the proposed algorithm is accurate and robust with all the tested sequences independently of the amount of tonal instabilities or motion.

5.4.1 Influence of parameters

As a first step towards experimental analysis of the proposed method, we study the influence of the algorithm parameters in the results. Our method has three parameters: ω can be seen as a geometric similarity threshold between u_t and u_k; σ, can be seen as the equivalent radiometric similarity threshold, and λ is the temporal weighting factor.

We take as case study two sequences ("graycard" and "sofa" - courtesy of [Farbman 2011]), from which a patch with homogeneous color can be succesfully tracked over time during the whole sequence. Firstly we fix $\sigma := 15$ and $\lambda := 1$ (no temporal weighting) to study the effects of the parameter ω on tonal stabilization results. We remind that ω is a threshold that accounts for the minimum percentual overlap area between two frames from which a color transformation can be estimated. It can be considered the most important parameter in our method, since it is directly linked to the rate of keyframe update. Secondly, $\omega := 0.25$ is fixed, to study the effect of applying the λ temporal weighting. In this scenario, the value of λ decreases exponentially over time when the motion or temporal difference from current frame to keyframe increases.

In Figure 5.6, we show the temporal intensity variation of a patch extracted from "graycard" sequence. The mean intensity variation in RGB is shown for the original sequence, and the sequences stabilized with ω varying between 0.25, 0.5 and 0.75. Although we could expect that such a wide variation in the values of ω would produce noticeable differences, we observe that for "graycard" sequence the value of this parameter has small influence on the result. The tonal fluctuations observed in the original sequence are efficiently stabilized in all the cases, apart for slight tonal variations that can be attributed to the noise variance. Nevertheless, if we observe carefully, it can be noted that with $\omega := 0.75$, tonal errors are more prominent, with

a slight drift, than when we set $\omega := 0.25$; which arguably produces the best result.

In Figure 5.7, we show the temporal intensity variation of a patch extracted from "sofa" sequence. Note that in the original sequence we observe wild tonal oscillations, and different settings for parameter ω have a more visible influence on the result than in the sequence "graycard". When this sequence is corrected with $\lambda := 1$, the tonal fluctuations are overall stabilized, however, clipping is produced in green and red channels. On the other hand when the temporal weighting λ is applied, clipping is reduced by preserving a limited degree of the tonal variations observed in the original sequence.

In practice, we have observed that the optimal value for ω depends on the accuracy of the motion estimation. If the motion is not accurate, a greater value for ω can be preferable, so that the motion estimation error is less accumulated with time. In general, we have observed that $\omega := 0.25$ leads to stable color transformations in most cases.

Finally, in Figure 5.8 we fix $\omega := 0.25$ and $\lambda := 1$ to observe the effects of varying the value of σ in the tonal stabilization of sequence "graycard". Note that when applying a correction with $\sigma := 30$ to "graycard" sequence the color transform estimation is more affected to outliers, producing less accurate tonal stabilization, than when we set $\sigma := 5$.

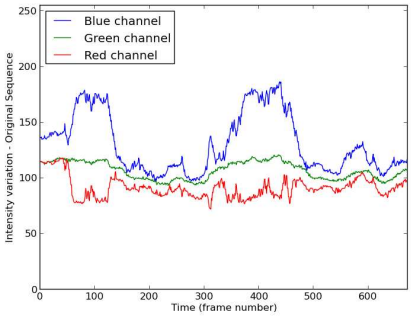
5.4.2 Goodness of fit

In order to evaluate the accuracy of our power law model, we have estimated the mean R^2 (coefficient of determination) along color channels:

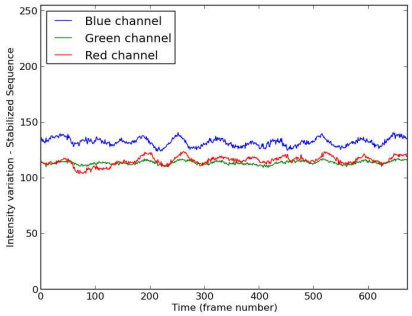
$$R^2 = \frac{1}{3}\sum_c \left(1 - \frac{\sum_{\boldsymbol{x}\in\Omega_p}(\log u_k^c(\boldsymbol{x}) - T_t^c(\log u_t^c(\boldsymbol{x})))^2}{\sum_{\boldsymbol{x}\in\Omega_p}(\log u_k^c(\boldsymbol{x}) - \overline{\log u_k^c})^2}\right). \tag{5.35}$$

where Ω_p is the set of points selected from a color chart in the image. In particular, we consider images captured with a smartphone from the same scene containing a Macbeth color chart (see Fig. 5.9). Each picture is adjusted (using the camera settings) to have a different exposure or white balance (WB), so that we can analyze the tonal changes by studying the color transfer function between the reference picture (sunlight WB, medium exposure) and the other ones. More specifically, we use the median color value of each color in the color chart to estimate a power law transformation. In Fig. 5.10, we plot the functional relationship (in logarithmic domain) between the colors extracted from the color chart of the reference picture and the correspondent colors from the other pictures. As an indication of goodness of fit, the computed R^2 value is shown for each plot (the closer is R^2 to 1, the better the observed tonal transformation fits the model). The coefficient is larger than 0.9 for all the computed regressions, which shows that the relationship between reference and test color intensities is approximately linear in a logarithmic scale and in general the model fits the data.

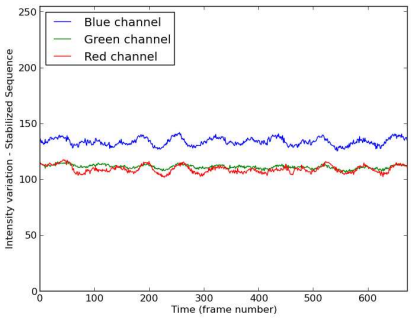
Note that the images with the color chart are useful to evaluate our model but they are not enough challenging to compare our results with other methods in the



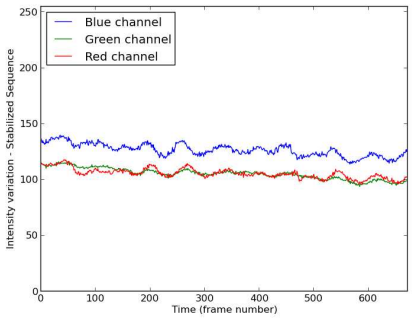
Original Sequence



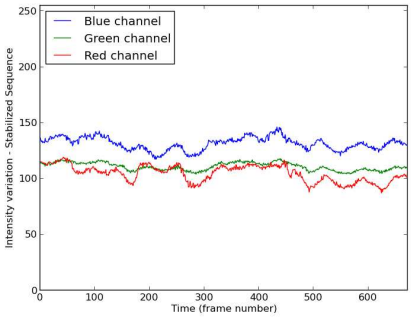
Stabilized: $\omega := 0.25$, $\lambda := 1$



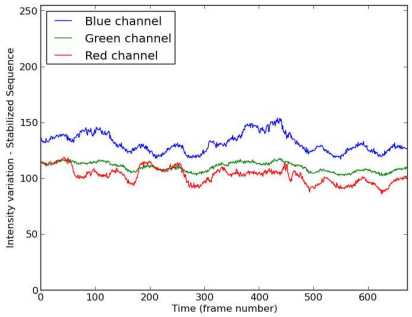
Stabilized: $\omega := 0.5$, $\lambda := 1$



Stabilized: $\omega := 0.75$, $\lambda := 1$

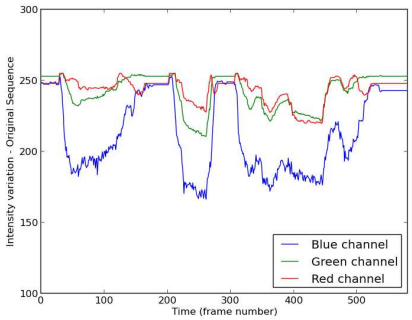


$\omega := 0.25$, motion based λ

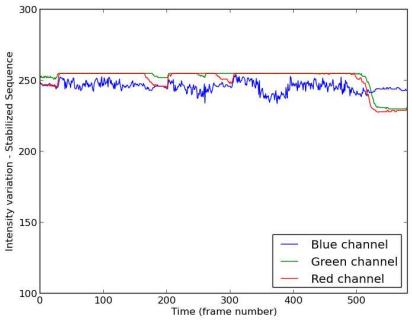


$\omega := 0.25$, time based λ

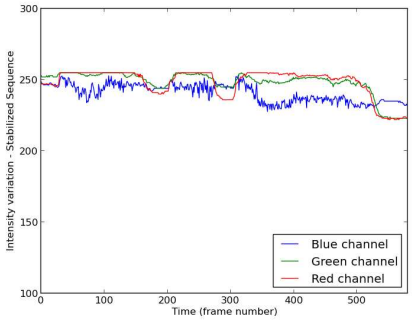
Figure 5.6: Influence of parameters ω and λ on tonal stabilization of sequence "graycard". In each plot, we present the temporal variation of RGB coordinates from a tracked homogeneous gray patch. Note that different settings for parameter ω with $\lambda := 1$ produce small influence on the result, and the sequence is efficiently stabilized in all the cases (apart for slight intensity fluctuations that can be attributed to image noise). When applying a λ weighting based on time or motion difference to keyframe, some degree of the original intensity variations are maintained.



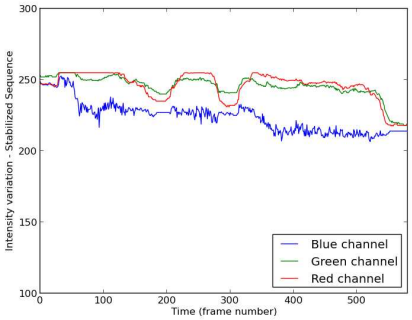
Original Sequence



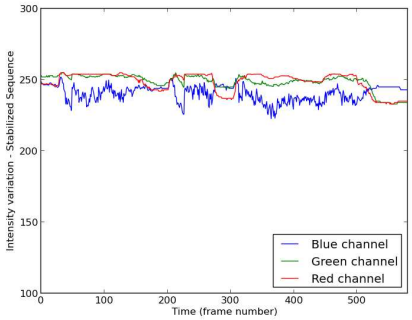
Stabilized with $\omega := 0.25$



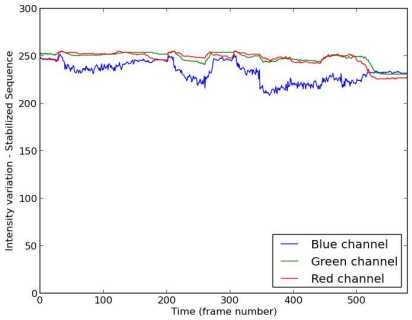
Stabilized with $\omega := 0.5$



Stabilized with $\omega := 0.75$

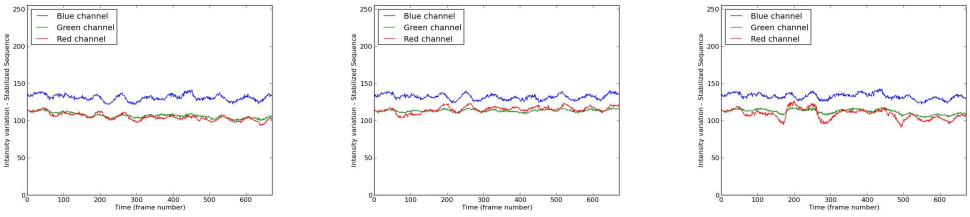


Motion based λ



Time based λ

Figure 5.7: Influence of parameters ω and λ on tonal stabilization of sequence "sofa". Note that this sequence suffer from severe tonal instability, in particular in the blue color channel, which makes it challenging to have an accurate tonal stabilization. Nevertheless, with $\omega := 0.25$ we can still obtain a stabilized result, at the cost of clipping red and green coordinates. We can observe that increasing the value of ω decreases the tonal preservation. When applying a λ weighting based on time or motion difference to keyframe, some degree of the original intensity variations are maintained and channel clipping is reduced.



$\omega := 0.25$, $\sigma := 5$ $\omega := 0.25$, $\sigma := 15$ $\omega := 0.25$, $\sigma := 30$

Figure 5.8: Influence of parameter σ on tonal stabilization of sequence "graycard". In each plot, we display the mean intensity variation of RGB channels from a tracked gray patch. Note that when applying a correction with $\sigma := 30$, the color transform estimation is more affected to outliers, producing less accurate tonal stabilization, than when we set $\sigma := 5$.

case of videos. In fact, all methods are comparable with this data since the presence of all the colors in the chart help the camera automatic white balance algorithm to work properly, without producing strong tonal instabilities.

Finally, we note that the tonal fluctuations caused by automatic camera settings in videos are far less intense than the tonal changes presented in Fig. 5.9, which were produced by manually adjusting the camera settings.

5.4.3 Qualitative evaluation

In practice, our method has been tested on 18 different video sequences. While some sequences have been kindly provided by the authors of [Farbman 2011], we have completed our dataset with video sequences acquired with smart phones from different manufacturers. Complete video sequences (originals and results) are available at the project website[4]. We strongly recommend the reader to look at the electronic version of the paper and the videos in our website to appreciate the results.

First, we have considered video sequences in which camera motion is not complicated as in the sequence "sofa" (see Fig. 5.11). In the original sequence one same object appears with different colors (e.g., sofa) while in our resulting sequence all colors are stable.

We present our tonal stabilization result for a sequence with fast motion in Figure 5.12. The video, taken while driving in a highway, is particularly challenging because of the driving speed, the fast motion of objects, the rain droplets falling and the wiper blade movements.

We note that camera zoom is challenging to be estimated by our dominant motion model. Nevertheless, our tonal stabilization method compensates innacurate motion estimation with the keyframe update approach. We have observed in practice that for sequences where the registration is not accurate, the number of motion

[4] http://oriel.github.io/tonal_stabilization.html

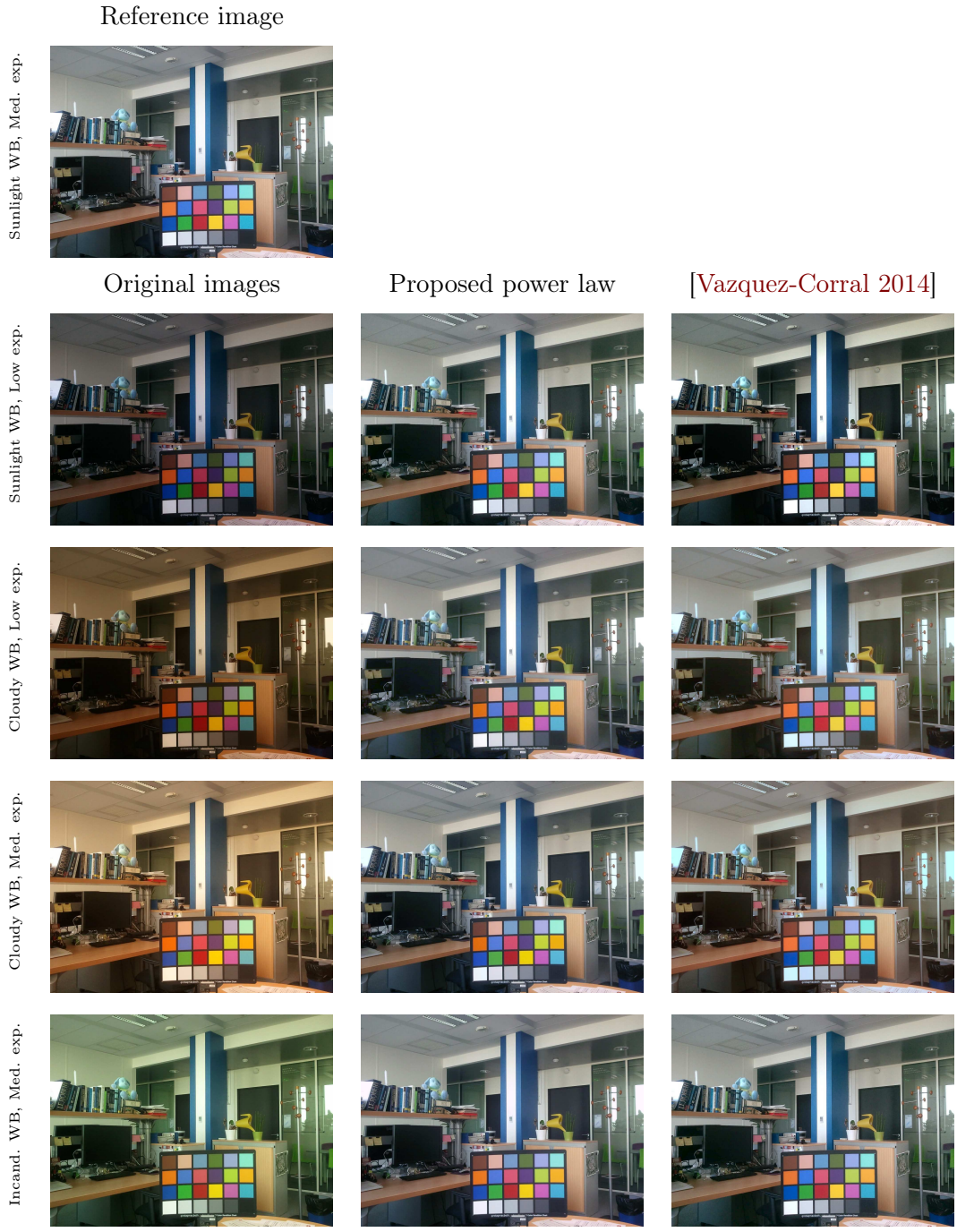


Figure 5.9: In this experience, a sequence of approximately registered images is taken from the same scene, and the color transformation is estimated from correspondent points. **On the first row**, a reference image (keyframe u_k) is shown on the first column. **From the second to fifth rows**, on the left column, tonally unstable images (u_t) taken with different exposures and white balance are shown. On the middle column, the color corrected images with our model are shown, and on the right column, it is shown the color corrected images with model [Vazquez-Corral 2014] for comparison. It can be noted that tonal instability is largely reduced when images are corrected with both models, but our model is less computationally demanding.

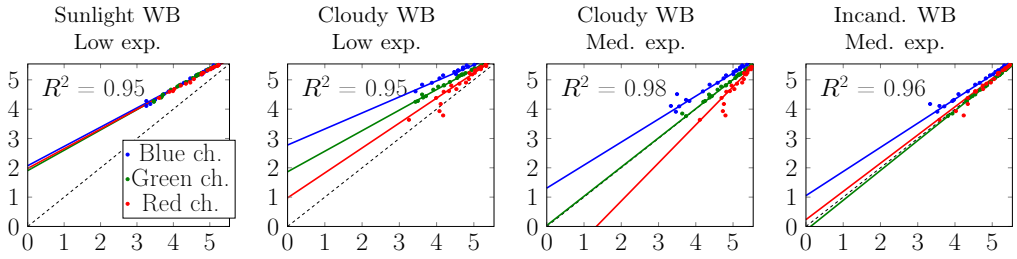


Figure 5.10: The goodness of fit of our model is analysed in this figure. The extracted points from the color chart and the estimated regression lines are plotted in the logarithmic domain. The dashed black line corresponds to the identity, the x-axis corresponds to $\log u_t^c(\Omega_p)$, while the y-axis corresponds to $\log u_k^c(\Omega_p)$ for the plotted points and $\log T_t^c$ for the plotted lines, where Ω_p is the set of color chart coordinates. The regression line has a close fit to the color points, reminding that R^2 values which are close to 1 are an indication of a good fit.

outliers increases and keyframe updates are triggered with more frequence. For example, in the driving sequence in Fig. 5.12 we can see objects changing in scale, nevertheless our model still correct tonal instabilities in the sequence and does not generate artifacts.

In order to evaluate our results with respect to state-of-the-art tonal stabilization methods, we have considered the methods of Farbman et al. [Farbman 2011] and Wang et al. [Wang 2014]. In our comparison, the results from [Wang 2014] have been provided by the authors while the results from [Farbman 2011] come from our implementation of their method, which is coherent with the results published in their paper and website.

In particular, Fig. 5.13 compares our results with the sequence "building" that has been acquired with a Samsung Galaxy S smart phone. Note that the results from [Farbman 2011] are not perfectly stabilized due to the important camera motion of the sequence, and the results from [Wang 2014] are stabilized but the sequence is much whiter and parts of it are completely saturated which is not visually pleasant. Fig. 5.14 compares with the sequence "graycard" the same algorithms which turn out to have the same behavior in terms of remnant tonal variation for [Farbman 2011] and wash-out look (white) for [Wang 2014]. On the contrary, in the two sequences "building" and "graycard", our results are both stable and color coherent with a good dynamic range.

5.4.4 Quantitative evaluation

5.4.4.1 Comparison of color transformation models

In order to evaluate our power law color transformation, we compare it to the parametric color transformation proposed by [Vazquez-Corral 2014], given by Eq. 5.22. Our parameter estimation of model [Vazquez-Corral 2014] relies on gradi-

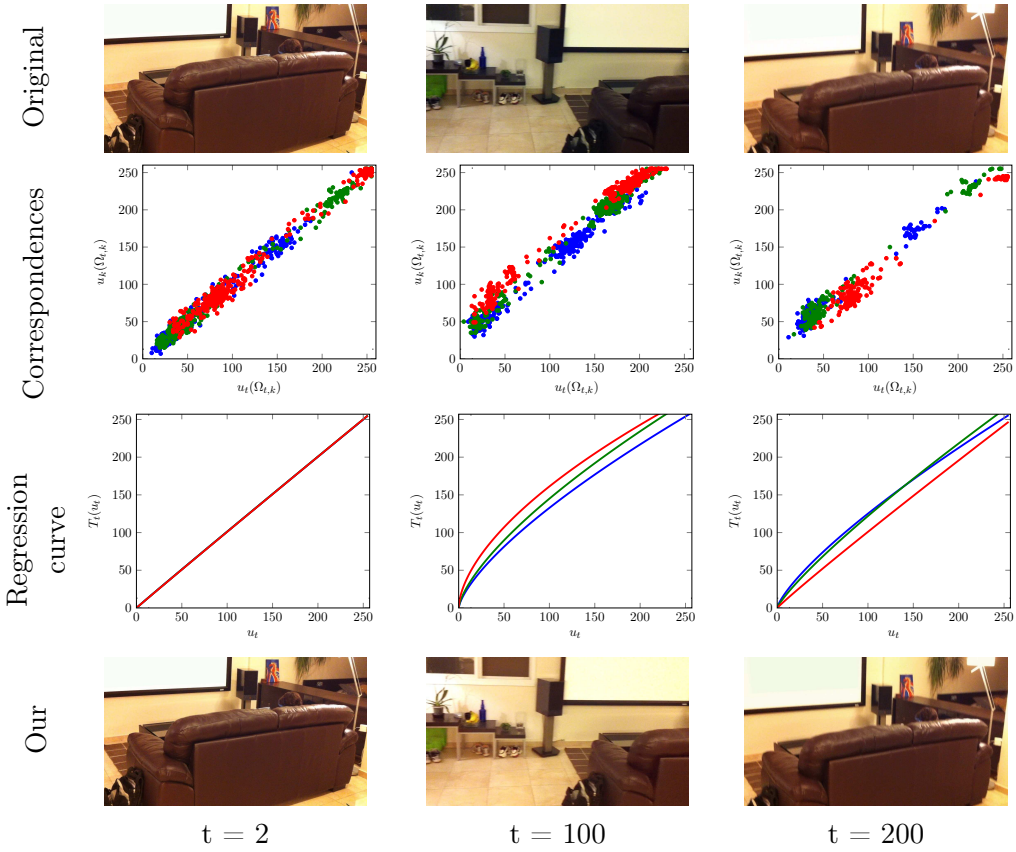


Figure 5.11: Tonal stabilization of the sequence "sofa". Top row: frames extracted from the original sequence, t = (2, 100, 200, 400). Second row: plot of point correspondences between the original frame u_t and the keyframe u_k. Third row: estimated power law tonal transformation for each frame. Bottom row: same frames from top row, after tonal stabilization with our method. Note that objects appearing with different colors in the original sequence have the same color in our results. The color of the plotted points and curves correspond to the sRGB color channel (Red, Green, Blue) of the image.

Figure 5.12: Tonal stabilization of the sequence "driving". Top row: frames extracted from the original sequence, t = (350, 375, 400, 425). Second row: same frames from top row, after tonal stabilization with our method. This video, taken while driving in a highway, is particularly challenging because of the driving speed, the fast motion of objects, the rain droplets falling and the wiper blade movements. Still, our method produces satisfactory tonal stabilization for this sequence, with no artifact creation.

ent descent, as a practical approach to solve the non-linear least squares problem. We note that our implementation differs from the original work, which employs a 2-stage parameter estimation solving for γ_1 and γ_2 in an exhaustive approach and for M by Singular Value Decomposition. We made experiments with the tonal transformation model proposed in [Vazquez-Corral 2014] based on trustworthy color correspondences between the reference and test images and estimating all the coefficients of the model by gradient descent. Images are aligned by homography using corespondences from SIFT and RANSAC, and to guarantee that no outliers are used for coefficient estimation, the color correspondences used to compare our method to [Vazquez-Corral 2014] are taken from the mean 24 colors of the Macbeth color chart, in particular, we use the same test images and camera configurations as shown in Figure 5.9.

In Figure 5.15, we illustrate the PSNR distribution for each color in the Macbeth colorchart for the camera configuration *Sunlight WB, Low exp.*, comparing our color transformation model to the parametric method of [Vazquez-Corral 2014] and the non-parametric "PDF transfer" method of [Pitié 2007].

According to our experiments with four different white balance and exposure configurations, summarized in Table 5.1, for three of the shooting configurations the color correction model proposed by [Vazquez-Corral 2014] is in average more accurate than our model (in terms of PSNR), while for one test images our model is in average more accurate. In particular, for this set of four images, we observed that [Vazquez-Corral 2014] performed better when white balance changed between reference and test images, while our power law performed better when only exposure changed between reference and test images. However, we note that the number of test images is not large enough to draw a solid conclusion about the two models. Furthermore, it is well known that PSNR is a limited metric, which does not always

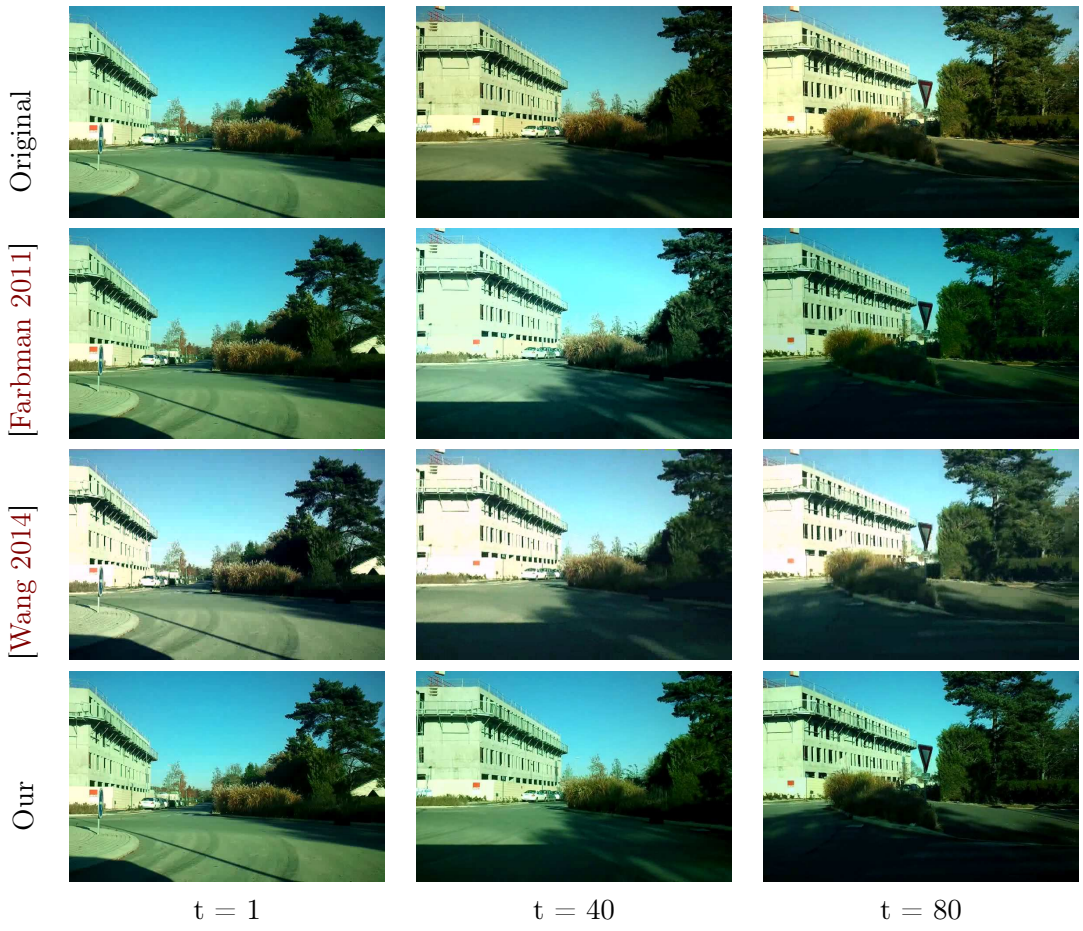


Figure 5.13: Comparison of tonal stabilization for the sequence "building" with the methods of [Farbman 2011] and [Wang 2014]. This figure shows three frames of the sequence (t=1,40,80). Our algorithm is able to stabilize tonal variations without generating artifacts, while the results from [Farbman 2011] are not perfectly stabilized and the results of [Wang 2014] tend to saturate parts of the building.

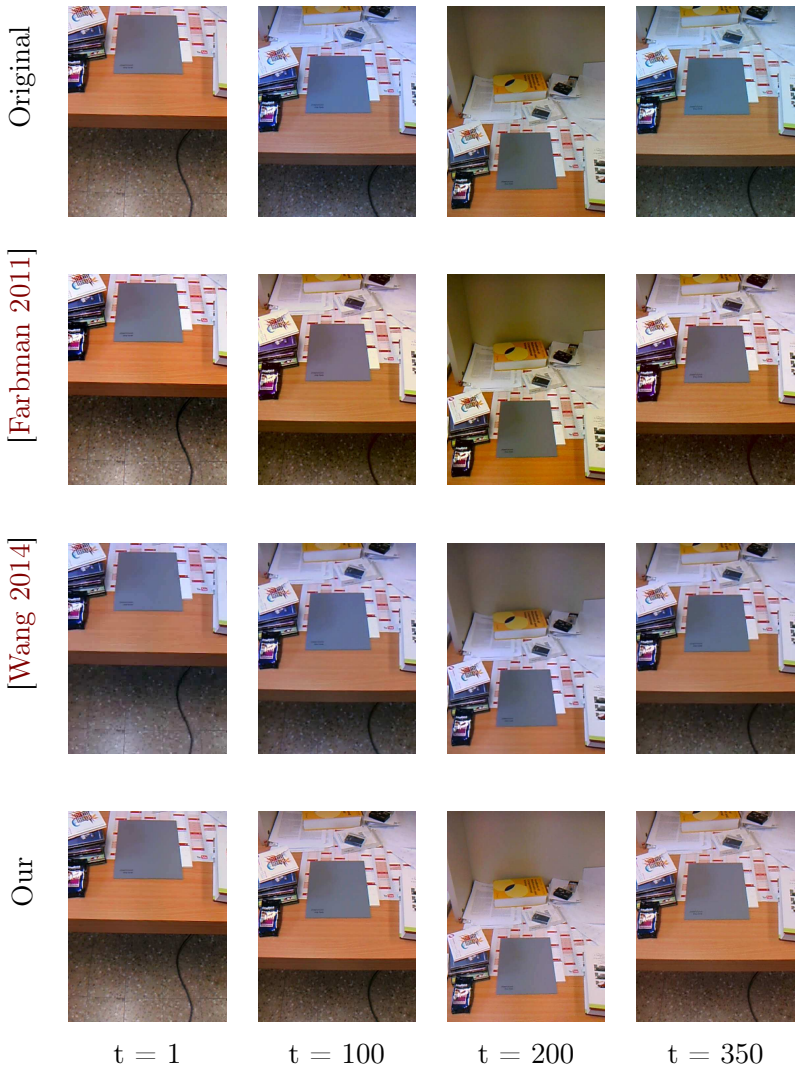


Figure 5.14: Tonal stabilization of the sequence "graycard". First row: frames extracted from the original sequence (t=1, 100, 200, 350). Second row: results from [Farbman 2011]. Notice the yellowish color for t=200. Third row: results from [Wang 2014] with a wash-out appearance. Bottom row: our stabilized results without any visual artifact.

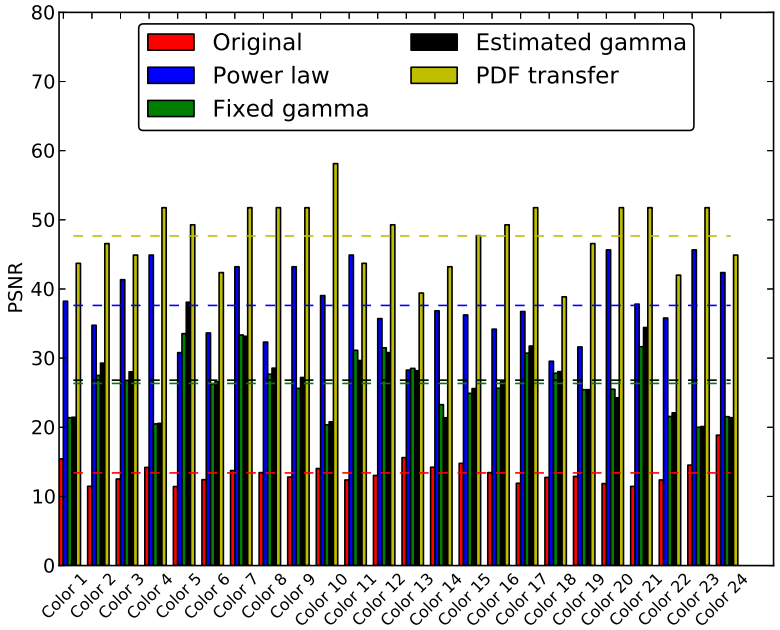


Figure 5.15: Quantitative results: **Sunlight WB, Low exp.** In order to evaluate our power law color transformation (plotted in blue), here we compare it to two other methods: the parametric color transformation proposed by [Vazquez-Corral 2014] either by fixed γ (plotted in green) and estimated γ (plotted in black); and the non-parametric N-dimensional PDF (Probability Density Function) transfer proposed by [Pitié 2007] (plotted in yellow). Plotted bars correspond to the PSNR between the reference and the corrected images, from the same experience illustrated in Figure 5.11. Different groups of bars correspond to each of the 24 colors of the Macbetch colorchart, dashed lines correspond to the mean PSNR over all 24 colors. Note that the PDF transfer method of [Pitié 2007] matches the 3D color histograms of images and is expected to be very accurate for the test images in our experiment, but at the cost of high computational complexity.

Table 5.1: Comparison of different color transformation models. Mean PSNR is computed as described in [Vazquez-Corral 2014]. For PSNR computation, we use the color correspondences taken from the 24 colors of the Macbeth color chart, based on reference *Sunlight WB, Low exp.* and test images as shown in Figure 5.9. We can observe that for this set of images, our transformation model has higher PSNR when there is only exposure changes while [Vazquez-Corral 2014] has has higher PSNR for white balance (WB) changes. Note that PSNR values does not always correlate well with human perception of image quality. Overall, both models have comparable accuracy, while our model is much less computationally demanding.

	Original	Our Power Law	[Vazquez-Corral 2014]
Sunlight WB, Low exp.	13.40	**37.62**	36.41
Cloudy WB, Low exp.	12.46	28.84	**31.12**
Cloudy WB, Med. exp.	18.68	29.27	**30.94**
Incand. WB, Med. exp.	23.73	34.39	**37.56**

correlate well with human perception of image quality.

We note that the model proposed by [Vazquez-Corral 2014] is more complete and arguably closer to the radiometric calibration pipeline [Kim 2012] than our model. Nevertheless, we observed in practice that the estimation of optimal coefficients of model [Vazquez-Corral 2014] is less obvious than ours. This can be noted in the experiment shown in Table 5.2, where we present a runtime comparison between the proposed power law model and the model proposed by [Vazquez-Corral 2014]. For a fair comparison, in this experiment we estimate the coefficients for both models with the same gradient descent optimization, in addition to estimating our model parameters by analytic expression. Color correspondences used for the comparison are taken from the mean 24 colors of the Macbeth color chart, based on test images and camera configurations as shown in Figure 5.9, with *Sunlight WB, Med. exp.* as reference image. It can be observed that the processing time required to estimate the coefficients of our model is approximately 100 times lower than [Vazquez-Corral 2014] when computed by analytic expression, and 10 times lower when computed with gradient descent.

In conclusion, we may argue that both our model and [Vazquez-Corral 2014] can be seen as practical approximations to the unknown inverse camera tonal transformation studied in [Kim 2012], where model proposed by [Vazquez-Corral 2014] seems to be best targeted for color stabilization among photographies taken with arbitrary cameras, while our model seems to be best targeted for video tonal stabilization, where a sequence of images is typically taken from the same camera and lower computational complexity is an important requirement.

5.4.4.2 Comparison of video tonal stabilization methods

In an effort to quantitatively assess the performance of our algorithm we propose to study the tonal variation of a homogeneous patch with respect to the reference (first)

Table 5.2: Runtime comparison between proposed power law and model by [Vazquez-Corral 2014] (Eq. 5.22), denoted as V.C. in the table. For a fair comparison, in this experiment we estimate the coefficients for our model by analytic expression (Eq. 5.17) and also with the same gradient descent (G.D.) optimization we used to estimate the model parameters of [Vazquez-Corral 2014]. In both cases our model is much faster. Color correspondences used for the comparison are taken from the mean 24 colors of the Macbeth color chart, based on test images and camera configurations as shown in Figure 5.9

	Power law (Eq. 5.17)	Power law (G.D.)	V.C (G.D.)
Fig. 5.9, 2nd row	**0.00034s**	0.0028s	0.028s
Fig. 5.9, 3rd row	**0.00059s**	0.0022s	0.043s
Fig. 5.9, 4th row	**0.00038s**	0.0037s	0.031s
Fig. 5.9, 5th row	**0.00062s**	0.0032s	0.040s

frame through the sequence. This is, considering the resulting sequence, we compute the color differences (in CIELAB color space) between a homogeneous patch in the reference frame P_0 and its corresponding patches through the resulting sequence P_t, $t = 1, \ldots, D$. Ideally, the patch tonal variation remains constant and equal to zero. However, this criterion is not sufficient to evaluate a tonal stabilization algorithm. For instance, a resulting sequence of completely homogeneous color frames would satisfy this criterion but would not be a good (pleasant) result. Because of this reason we also study the fidelity to the original sequence by computing the color difference between the same aforementioned patches P_t and the same patches on the original sequence P_t^o, for all $t = \{1, \ldots, D\}$. A resulting sequence with a large deviation from the original sequence would produce undesired artifacts. With these two criteria being defined we consider a tonal stabilized sequence being a good result when the patch tonal variation is as constant and small as possible and *at the same time* the fidelity to the original sequence is as much preserved as possible.

For sequences not containing a fixed color chart, the two error curves (tonal variation and fidelity to original) can be computed, provided the video sequence has a homogeneous patch, as it is the case for the sequence "building" or the sequence "greycard". Fig. 5.16 shows the error curves for these two sequences. We observe that the patch tonal variation is reduced with our method and the method of [Wang 2014] when compared to the patch tonal variation of the original sequence but this is not the case for the results of [Farbman 2011]. Also, our method produces the closest results in terms of color fidelity to the original sequence. Notice that for the sequence "graycard" the fidelity to original is smaller for [Wang 2014] between t=75 and t=150. We explain this behavior because we choose the first frame as reference. Indeed we obtain a smaller fidelity to original for the first frames (from t=1 to t=75), but then, when there is a big instability of the original sequence for these frames (see red curve of the tonal variation) our algorithm compensates this

difference. As we have explained in Fig. 5.5 we believe that our weighting strategy provides more natural and artifact-free results.

5.4.5 Computational time

Besides the qualitative and quantitative evaluation our method is also more efficient in comparison to the state-of-the-art. Our prototype implementation in Python processes a $1920 \cdot 1080$ resolution video in a rate of 11 frames per second considering image reading and writing and 20 frames per second without reading and writing[5]. On the contrary, the C++ implementation of [Wang 2014] processes 1 frame per second (depending on video length) and a Python implementation of [Farbman 2011] processes 0.6 frames per second. We believe that an optimized implementation of our method could approach real time processing which is a major advantage and proves the feasibility of embedding robust tonal stabilization algorithms on smart phones.

[5]Processed by Intel(R) Core(TM) i5-3340M CPU @ 2.70GHz, 8GB RAM

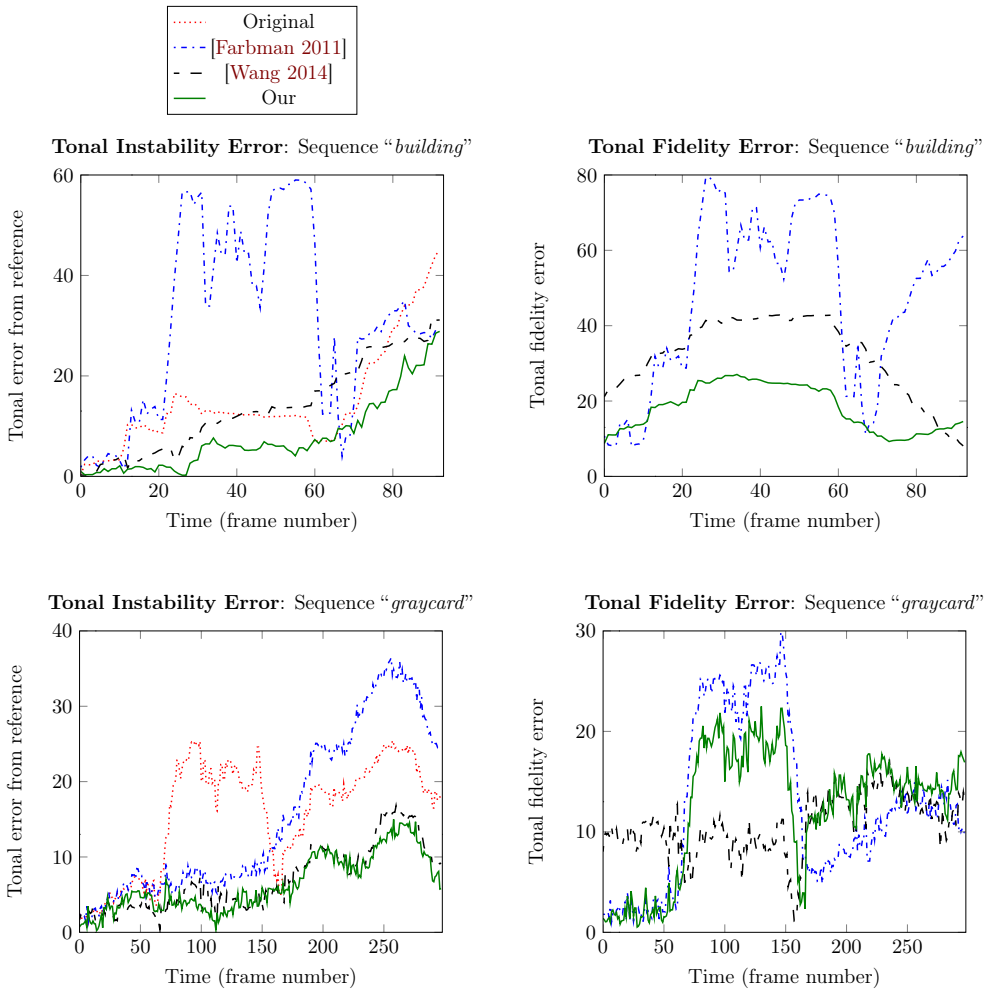


Figure 5.16: Quantitative evaluation of the sequence "building" (top) and "graycard" (bottom). For each sequence, we show (i) the tonal instability error, computed as the color distance of a tracked patch to the reference (first) frame, and (ii) the tonal fidelity error, computed as the color distance at each instant, between the corrected frame and the original frame, which indicates the degree of fidelity between the original and the corrected sequence. The color distances are computed as the euclidian distance in perceptual color space CIELAB. Overall, our method compares favorably with the methods of [Farbman 2011] and [Wang 2014], both in terms of *reduction of tonal instability* as in terms of *fidelity to original colors*.

5.5 Limitations and Perspectives

Besides the effectiveness of our method to model and compensate tonal instability, we note the following limitations that could be approached in future work:

- *Scenes with complex motion*: Since we rely on 2D dominant affine motion, the accuracy of spatial correspondences can be affected in sequences containing more complex motion (i.e. motion parallax caused by non-trivial depth variation in the scene). As we claimed in Section 5.3.1.2, a rough affine registration is usually enough to compute tonal compensation, as long as we can reject the motion outliers and capture the color correspondences in overlapping homogeneous regions. But clearly, our method would benefit from a more accurate motion estimation (such as local camera paths) to be applied to more challenging sequences.

- *Temporal preservation of achromatic colors* under severe underexposure: In some cases, the tonal compensation of underexposed frames (using a well exposed keyframe as reference frame) can produce inconsistent tonal mapping for achromatic colors. In other words, underexposed objects which are originally achromatic can be assigned to chromatic colors after tonal stabilization. For example, if a red image region at time t becomes suddenly dark (nearly black) at time $t+1$, in order to compensate for this region, the estimated color transformation may inconsistently transform a black object at $t+1$ to also have a reddish chromaticity. This inconsistency can occur because the estimated tonal transformation cannot differ image intensities which are originally dark from intensities that become dark due to underexposure. Local color transformations would possibly be a solution to deal with this ambiguity.

- *Optimal coefficient filtering*: The estimated coefficients of the power law color transformation may oscillate from a frame to the next, if image noise is too severe or motion estimation is highly innacurate. Applying the temporal weighting as an exponential forgetting (parameter λ) reduces this risk. But a possible extension would be to analyse and filter the coefficient transformations based on optimal adaptive filtering approaches, such as the Kalman Filter or the Particle filter.

5.6 Considerations

In this chapter, we have proposed an efficient tonal stabilization method, aided by motion estimation and using a power law tonal transformation to model color changes in videos. We have shown that a simple six-parameters color transformation model is enough to provide tonal stabilization caused by automatic camera parameters, without the need to rely on any prior knowledge about the camera model.

In contrast to the state of the art, the proposed algorithm is robust for sequences containing motion, it reduces tonal error accumulation by means of long-term tonal propagation, and it does not require high space and time computational complexity to be executed.

In addition, one of the main advantages of the proposed method is that it could be applied in practice as an online algorithm, that has potential for real time video processing applications such as tonal compensation for video conferences or for live broadcast.

Finally, we note that the presented method for tonal stabilization can be seen as an example-based approach, as keyframes are taken as example images from which color transformations are estimated. Hence, our tonal stabilization can be interpreted as a color transfer method, where the characteristics to be transferred are the white balance and the exposure of keyframes.

Chapter 6

Style Transfer

Contents

6.1 Introduction

In the previous chapters, we have taken color as a feature of interest to be mimicked from example images. In this chapter, we turn our interest to style, which is in itself difficult to define, since it can be seen as a combination of different visual features.

Style transfer is the task of transforming an image in such a way that it mimics the style of a given example. This class of computational methods are of special interest in film post-production and graphics, where one could generate different renditions of the same scene under different "style parameters" [Kyprianidis 2013] [Durand 2002].

The upcoming film "Loving Vincent" [1] can be seen as a remarkable example of the practical possibilities of style transfer. This endeavor is claimed to be the first fully painted feature film to be made, and includes an average of 12 oil paintings per second of video. Note that more than 100 painters were involved in the production, which is a painstaking work that could be facilitated by style transfer techniques.

Patch-based methods have been widely employed to solve problems such as texture synthesis [Efros 2001], inpainting [Criminisi 2004], and super-resolution [Freeman 2002] with state-of-the-art performance. These non-local and non-parametric approaches draw on the principle of self-similarity in natural images: similar patches (sub-images) are expected to be found at different locations of a single image.

Despite the practical success of patch-based methods for inverse problems, the patch dimensionality remains a sensitive parameter to tune in these algorithms. For instance, to obtain a coherent patch-based texture synthesis, patches should have approximately the same dimensionality of the dominant pattern in the example texture. The problem of patch dimensionality is equally crucial for example-based style transfer. In this case, we are given as example an image containing a mixture of style and content. Hence, patch dimensions should be large enough to represent the patterns that characterize the example style, while small enough to forbid the synthesis of content structures present in the example image. We propose a style transfer method that is able to meet these requirements by means of an adaptive patch partition. Fig. 6.6 illustrates our method.

In this chapter, we suggest that a correct style transfer can be thought as a local transfer of texture and a global transfer of color. A robust method for local texture transfer must capture the style while preserving the image structure, and this can be achieved with a spatially adaptive image partition.

Moreover, we show that a relevant partition must incorporate a prediction of how well an image portion of the source image will be matched to the example style image. That naturally leads to an example-based partition, where the partition is bound to the coupling between the source image and the example image.

Finally, we show that an adaptation of image style transfer for videos requires some strategy to guarantee temporal coherence. We show that temporally coherent stylization can be achieved through the combination of motion based warping and style resynthesis for areas where optical flow is not reliable.

6.2 Related Work

Style transfer can be related to texture [Efros 2001] and color transfer [Reinhard 2001a, Frigo 2014]. Texture transfer can be seen as a special case of texture synthesis, where example-based texture generation is constrained by the geometry of an original image. Style transfer, for this part, can be seen as a composition of texture and color transfer, where style is transferred from an example to an original image, being modeled as a combination of texture and color. Recent methods modeling

[1] http://join.lovingvincent.com/

style transfer in a color context include [Shih 2014], where the style of head shots is mimicked through local image statistics and [Shih 2013] where the daytime of an image is transformed relying on examples. In this work, we approach style mainly from the textural rather than the color aspect.

6.2.1 Markov Random Fields in Computer Vision

In a Markov chain, a sequence of one-dimensional random variables $\boldsymbol{X} = (X_1, ..., X_N)$ has a joint distribution given by the conditional probability $P(X_i|X_{i-1}, X_{i-2}, ..., X_1)$. In other words, the probability of an event X_i is conditioned by the probability of previous events $(X_{i-1}, X_{i-2}, ..., X_1)$ [Blake 2011]. Under a first-order Markov chain, we have a simplified assumption

$$P(X_i|X_{i-1}, X_{i-1}, ..., X_1) = P(X_i|X_{i-1}), \tag{6.1}$$

which means that $P(X_i)$ has the "memoryless" property of only being conditioned to its previous event $P(X_{i-1})$. Markov Random Fields (MRF) is a generalization of the Markov chains to two-dimensional random variables, being a suitable formalism to model different inverse problems in image processing and computer vision, such as denoising [Geman 1984], super-resolution or optical flow estimation [Freeman 2000].

When modeling vision problems by Markov Random Fields, the "memoryless" property of Markov chains is generalized to a neighborhood smoothness prior. A piecewise smoothness prior is common for modeling natural images [Weiss 2007], where an observed pixel intensity is conditioned to its neighbor pixels, thus it is generally expected that neighbor pixels have similar intensity.

According to [Freeman 2000], modeling image reconstruction problems with MRF can be illustrated considering a relationship between an *observed image*, and a *latent scene* - the hidden variables we want to infer. Then, the question we want to answer is: what is the underlying scene that most likely explains the observed image? For answering the question, we need to sample the most probable "latent scene" from the MRF modeling the problem.

In computer vision, a MRF is usually represented as a probabilistic graphical model in the form of an undirected graph $G = (V, E)$, where vertices V corresponds to the set of problem variables, and edges E correspond to the modeled joint probabilities between the model variables. An illustration of Markov Random Field model for digital images is shown in Figure 6.1.

Solving a Markov network involves a learning phase, where the parameters of the network connections are learned from training data, and an inference phase, when the scene corresponding to particular image data is estimated [Freeman 2000]. For a Markov random field, the joint probability over the scenes x and images y can be written as:

$$P(x_1, ..., x_N, y_1, ..., y_N) = \prod_{(i,j)} \psi(x_i, x_j) \prod_k \phi(x_k, y_k), \tag{6.2}$$

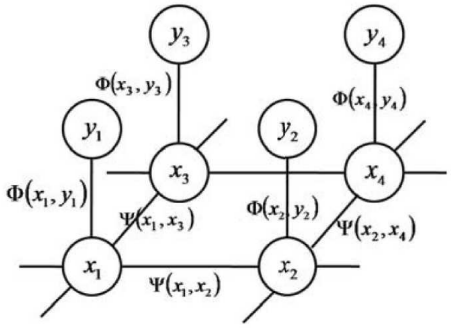


Figure 6.1: An example of simple Markov Network for vision problems. Each node in the graph describes an image or scene unit. Observed variables y_i have underlying scene explanations x_i, which are computed by probabilistic inference. Image courtesy of [Freeman 2000].

where ψ and ϕ are compatibility functions (edge potentials) learned from training data, (i, j) denote neighboring nodes i, j and N is the number of image/scene nodes. Except for toy examples, computing a Maximum A Posteriori (MAP) inference to solve directly Eq. 6.2 is prohibitive due to the high dimensionality of the scene variables, but approximate solutions can be found by iterative algorithms.

One popular algorithm for approximate inference is the belief propagation [Weiss 1997]. The algorithm can compute an exact solution for tree-structured probabilistic graphs, but only an approximate solution to graphs containing loops. Loopy belief propagation is a popular adaptation of the original algorithm to deal with graphs containing loops, such as the image representations in vision problems.

In each iteration of the algorithm, neighboring variables update their likelihoods by *message passing* and after a number of iterations, the marginal probabilities (beliefs) of all the variables can be approximately determined [Freeman 2000].

6.2.2 Texture Synthesis

Texture synthesis by non-parametric sampling is inspired by the Markov model of natural language [Shannon 1948], where text generation is posed as sampling from a statistical model of letter sequences (n-grams) taken from an example text. In an analogous manner, non-parametric texture synthesis relies on sampling pixels directly from an example texture. It became a popular approach for texture synthesis [Efros 1999] and for texture transfer [Efros 2001], [Hertzmann 2001], [Zhang 2013] due to convincing representation of either non-structural and structural textures.

In the literature of texture synthesis and transfer, we find two main approaches to compute non-parametric sampling from an image based Markov Random Field (MRF), which we call here respectively as the *greedy* and the *iterative* strategies. The first strategy considers texture synthesis as the minimization of a greedy heuristic cost function, performing sampling by neighborhood matching to obtain a local

solution. The non-parametric texture synthesis method of [Efros 1999] takes a pixel to be synthesized by random sampling from a pool of candidate pixels selected from an example texture. This process is illustrated in Figure 6.2. A similar approach was extended to patch-based texture synthesis and also for texture transfer in [Efros 2001].

The patch sampling approach for texture synthesis, popularized by [Efros 2001] and [Liang 2001], has the advantage of being much faster than pixel-based synthesis (methods popularized by [Efros 1999] and [Hertzmann 2001]). As the patch becomes the unit of synthesis, the search space for neighborhood matching is reduced, but a post-processing step is required for patch border smoothing. To produce seamless patch blending, [Liang 2001] employed alpha blending (linear interpolation), while [Efros 1999] proposed to compute optimal boundary cut between patches, which was called image quilting.

Differently to the approach popularized by [Efros 1999], we follow in this chapter an *iterative* strategy, inspired by [Freeman 2000] and [Wang 2009]. The iterative approach considers an explicit probability density modeling of the sampling problem and computes an approximate Maximum a Posteriori (MAP) solution through an iterative optimization method. In this chapter, we adopt the Loopy Belief Propagation algorithm [Weiss 1997], which is an efficient and simple method for MRF inference.

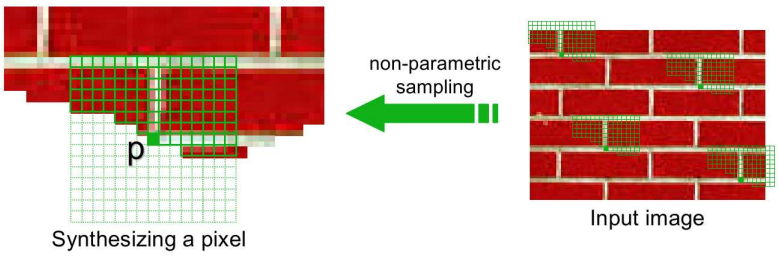


Figure 6.2: Illustration of texture synthesis by non-parametric sampling. A given pixel p is synthesized by random sampling from a pool of candidate pixels from an example texture. Candidates pixels have similar neighborhood to p and are found by neighborhood template matching. This process can be seen as a greedy approximation to MRF sampling. Image courtesy of [Efros 1999].

6.2.3 Texture and Style transfer

Style transfer can be computed in a *supervised* or *unsupervised* fashion. One of the first methods to propose supervised style transfer posed the problem as computing an "image analogy" given by $A : A' :: B : B'$ [Hertzmann 2001], implying that an input image B should be related to a stylized image B' the same way as image A is related to A', with A and A' known. In this method, inspired by the texture transfer of [Ashikhmin 2001], a pixel to be synthesized in image B' is directly selected from an example stylized image A', by minimizing a cost function that takes into account the

Figure 6.3: Illustration of image analogies. Image courtesy of [Hertzmann 2001].

similarity between B and A and the preservation of local neighborhoods in A'. The image analogies approach was extended for supervised stylization of animations in the work of [Bénard 2013], where the problem of temporal coherence is investigated and neighborhood matching is accelerated inspired by the randomized "PatchMatch" approach of [Barnes 2010]. A recent work has shown that patch correspondences can be further accelerated with hash tables in [Barnes 2015] and applications for style transfer are presented. We note that the supervised approach needs a registered pair of example images A and A' from which it is possible to learn a style transformation, however this pair of images is hardly available in practice. In this chapter we rather consider an unsupervised approach.

There are few works dealing with unsupervised style transfer in the literature, the closest to our method being [Rosales 2003], [Cheng 2008], [Okura 2015] and [Zhang 2013]. Still borrowing from the image analogies notation, we can consider that the *unsupervised* scenario assumes that only an example image A' and an original image B are given. In [Rosales 2003] the authors describe a Bayesian technique for inferring the most likely output image from the input image and the exemplar image. The prior on the output image $P(B')$ is a patch-based MRF obtained from the input image. The authors in [Zhang 2013] decompose original and example images into three additive components: draft, paint and edge. In our approach, the input image is not decomposed into additive parts, as we rather consider a spatial decomposition. Moreover, we note that in both [Rosales 2003] and [Zhang 2013], a MRF is defined for image patches disposed over a regular grid, which is not the case in our approach, where we consider an example-based adaptive image partition.

The work of [Okura 2015] shares some similarities with the approach we propose in this chapter, in the sense that example-based appearance manipulation is seen as a combination of color and texture transfer. In their approach, they evaluate the success of color transfer in reproducing the appearance of the example image, and texture transfer is performed only if color transfer is not sufficient. A typical example of appearance manipulation given in their paper is to render a picture taken in a season to look like it was taken in another season. Note that for this approach to work, input and example images should be ideally from the same scene.

Finally, the work of [Gatys 2015] proposed a "neural style transfer" technique using deep Convolutional Neural Networks (CNN) to separate and recombine the content and the style of two images. Their main idea is to represent style by cor-

relations between features from different layers of a CNN, and to represent content by feature responses in higher layers of a CNN. Their method produces impressive results for style transfer, but has the drawback of high computational complexity. This work received a great attention in vision community and resulted in products such as "DeepArt" [2] and an accelerated smartphone application "Prisma"[3].

Recently, neural style transfer for videos has been proposed by [Ruder 2016], where temporal coherence is enforced to the stylization by using optical flow guidance. Our approach for temporal coherence is similar in spirit to [Ruder 2016]: propagate style by motion warping where optical flow is reliable, and resynthesizing style where optical flow is not reliable.

It should be noted that image and video style transfer based on deep CNN's differs considerably from our approach, since it assumes a pre-trained neural network architecture. Although results of neural style transfer are mostly excellent, as remarked by [Fišer 2016], the stylization by neural networks tends to be unpredictable in practice. On the other hand, patch based approaches may be advantageous for a more predictable stylization that better preserves the main structures in the original image.

6.3 Image Style Transfer

According to the primal sketch theory of visual perception [Marr 1982], an image may be seen as a composition of *structures*: an ensemble of noticeable primitives or tokens; and *textures*: an ensemble with no distinct primitives in pre-attentive vision. Inspired by this principle, [en Guo 2003] presented a generative model for natural images that operates guided by these two different image components, that they called as *sketchable* and *non-sketchable* parts.

In this work, we adopt a similar view for example-based style synthesis. Our main motivation comes from the observation that the visual elements accounting for distinctive painting styles in fine arts are often anisotropic with respect to scale. In other words, details corresponding to the geometry (or the sketchable part) of a scene are often painted carefully with fine brushwork, while the scene non-sketchable part is sometimes painted with rougher brushes, where brushwork style is usually more distinct. Obviously, this observation holds more importantly for some particular artistic styles such as impressionism and post-impressionism than other painting styles such as realism.

We remind that in texture transfer, pixel-based models have assumed neighborhoods with regular size, and patch-based methods similarly assume an image decomposition into patches in a regular grid. As we illustrate in Fig. 6.4, a regular grid assumption is problematic for style transfer. In general, if the patches in a regular grid are small (for instance of size 8×8), we achieve a realistic reconstruction of the original image, but the style of the example image is hardly noticeable. On

[2] https://deepart.io/
[3] http://prisma-ai.com/

the other hand, for larger patch size, the style from the example can be noticed in the reconstructed image, however the fine geometry of the original image is not correctly reconstructed.

In order to overcome this limitation, we propose a method that takes into account the scale problem in stylization. In the following subsections, we give a formal definition for unsupervised style transfer and our proposed solution to the problem.

6.3.1 Problem definition

Let $u : \Omega_u \to \mathbb{R}^3$ be an input image and $v : \Omega_v \to \mathbb{R}^3$ an example style image. Style transfer can be posed as finding a correspondence map $\varphi : \Omega_u \to \Omega_v$ which assigns to each point $\boldsymbol{x} \in \Omega_u$ in the original image domain a corresponding point $\varphi(\boldsymbol{x}) \in \Omega_v$ in the example image domain. The output image can then be defined as $\hat{u} = v(\varphi)$.

In order to capture the style of v while preserving the structures of u, the correspondence map φ should ideally be a piecewise constant translation map on a partition $R = \{R_i\}_{i=1}^n$ of Ω_u. In practice, the partition R should depend on the geometrical content of u, while ensuring the existence of good correspondences between u and v over each region R_i. Note that we assume that images u and v are not from the same scene, thus matching stylized and non-stylized images such as proposed by [Russell 2011] does not apply to our problem. To achieve a convincing style transfer, regularity is also required at the boundary between neighboring correspondent regions.

All these requirements could be expressed in a unique non convex energy depending on both R and φ and requiring an alternating optimization strategy. For the sake of simplicity, our approach rather considers these sub-problems independently, following the four steps below:

1. Split and match: compute an adaptive partition R of Ω_u (Sec. 6.3.2);
2. Optimization: Search for the optimal map φ (Sec. 6.3.3);
3. Bilinear blending between neighboring regions and reconstruction of $\hat{u}$ (Sec. 6.3.4);
4. Global color and contrast matching (Sec. 6.3.5).

In the split and match step, we split Ω_u into a quadtree R which takes into account both the geometry of u and the ability for these regions to have good matches in v. At the same time, we compute for each region R_i a reduced set of candidate regions in v. The search of the optimal map φ is then seen as a graph labeling problem, where the nodes of the graph are the regions R_i. We denote $L_i = \{l_{i_c}\}_{c=1}^C$ the set of candidate labels for the region R_i, the label $l_{i_c} \in \Omega_v$ being a patch coordinate in image v. This probabilistic labeling problem is solved by belief propagation, followed by bilinear blending for the final reconstruction. We note that $\hat{u}$ is reconstructed by texture transfer only in luminance (Y channel in YUV color space).

Finally, we suggest applying the global color transfer method [Frigo 2014] proposed in Chapter 4 in the chrominance channel to capture the color style, and a contrast transformation to match the global contrast of the example image.

6.3.2 Split and Match adaptive partition

As we claim throughout this chapter, decomposing an image into a suitable partition has a considerable impact in the quality of patch-based style synthesis. We propose a simple yet effective approach based on a modified version of the classic Split and Merge decomposition [Horowitz 1974]. In the classic algorithm, the local variance of a quadtree cell decides whether a cell will be split into four cells. Here we propose a "Split and Match" example-guided decomposition, where the stopping criteria for quadtree splitting depends also on the patch similarity between the input and example images.

In our representation, a region R_i is a square of Ω_u, of size $\tau_i \times \tau_i$. We denote by $\boldsymbol{x}_i$ its center and we denote indifferently by $u(R_i)$ or $p^u_{\boldsymbol{x}_i}$ the patch of size τ_i^2 centered at $\boldsymbol{x}_i$.

The decomposition starts with one single region $R_1 := \Omega_u$. Each region R_i of the partition is split into four equal squares, each one of size $(\frac{\tau_i}{2})^2$, until a patch in the example image v matches $u(R_i)$ with some degree of accuracy.

Since quadtree patches can have arbitrary size, we use normalized distances for patch comparison. More precisely, the distance between two patches $p^u_{\boldsymbol{x}_i}$ and p^v_y of the same size τ_i^2 is defined as

$$d[p^u_{\boldsymbol{x}_i}, p^v_y] = \frac{||p^u_{\boldsymbol{x}_i} - p^v_y||^2}{\tau_i^2}. \tag{6.3}$$

Now, if y_i is the best correspondence of $\boldsymbol{x}_i$ in v at this scale τ_i:

$$y_i := \arg\min_y \; d[p^u_{\boldsymbol{x}_i}, p^v_y], \tag{6.4}$$

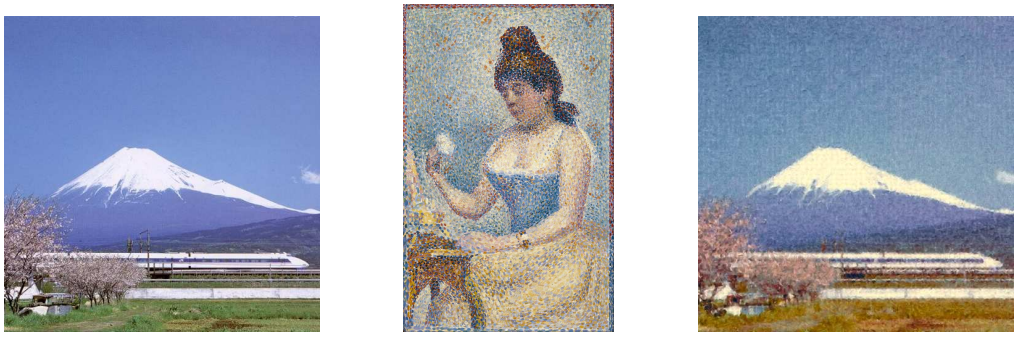
the region R_i is split in four regions if the following condition is satisfied

$$\begin{aligned} &\zeta(p^u_{\boldsymbol{x}_i}, p^v_{y_i}) = \\ &\big(\sigma_i + \; d[p^u_{\boldsymbol{x}_i}, p^v_{y_i}] > \omega \;\; \textbf{and} \;\; \tau_i > \Upsilon_0\big) \;\; \textbf{or} \;\; \tau_i > \Upsilon_1, \end{aligned} \tag{6.5}$$

where $\sigma_i = \sqrt{Var(p^u_{\boldsymbol{x}_i})}$ is the standard deviation of $p^u_{\boldsymbol{x}_i}$, ω is a similarity threshold (fixed to $\omega := 15$ in practice), Υ_0 is the minimum patch size and Υ_1 the maximum patch size allowed in the quadtree (respectively fixed to 8^2 and 256^2).

Observe that R_i is not encouraged to be split if there is at least one patch p^v_y which is similar enough to $p^u_{\boldsymbol{x}_i}$, unless the standard deviation of the patch σ_i is large.

Eventually, for every "leaf node" of the quadtree (nodes for which the splitting condition in Eq. (6.5) is not satisfied), a set of K candidate labels $L_i = \{l_{i_k}\}_{c=1}^{C}$ is selected for R_i by computing a spatially constrained K-nearest neighbors (k-NN) $\{p^v_{l_{i_k}}\}_{c=1}^{C}$ of $p^u_{\boldsymbol{x}_i}$ in v. A spatial constraint $|l_{i_c} - l_{i_{c+1}}| > \chi$ (with $\chi := \frac{\tau_i}{2}$ in practice),



Original | Example | Regular grid, $\tau_i := 8$

Regular grid, $\tau_i := 32$ | Variance-based quadtree | Our method

Figure 6.4: Comparison of partitioning strategies for style transfer. Small patches ($\tau_i := 8$) do not manage to capture the style, while large patches ($\tau_i := 32$) do not preserve the image structures. The adaptive partition based on image variance exhibits an artifact due to a large image portion that cannot be correctly matched. On the contrary, the example-based partition finds good matches for all parts.

requires that two candidate patch labels (pixel locations) l_{i_c} and $l_{i_{c+1}}$ are sufficiently distant from each other and encourages label variety. The whole split and match step is summarized in Algorithm 2.

Algorithm 2 "Split and Match" patch decomposition

Input: Images: u, v; parameters: Υ_0, Υ_1, ω
Output: Set of regions $R = \{R_i\}_{i=1}^{n}$, set of candidate labels $L = \{L_i\}_{i=1}^{n}$

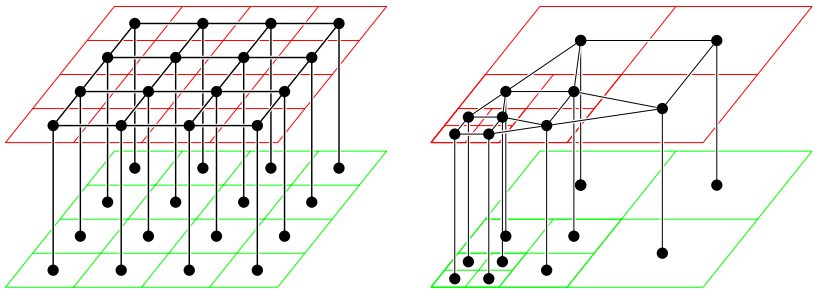
Initialization: $R_1 \leftarrow \{\Omega_u\}$
for every region $R_i \in R$ **do**
  $\boldsymbol{x}_i \leftarrow$ center of R_i
  $\sigma_i \leftarrow \sqrt{Var(p^u_{\boldsymbol{x}_i})}$
  Compute $y_i = \arg\min_y \ d[p^u_{\boldsymbol{x}_i}, p^v_y]$
  if $\zeta(p^u_{\boldsymbol{x}_i}, p^v_{y_i})$ is true **then**
    Split R_i into four:
    $m \leftarrow \sharp R - 1$
    $R \leftarrow \{R \setminus R_i\} \cup \{R_{m+1}, ..., R_{m+4}\}$
  else
    Compute spatially constrained k-NN:
    $L_i \leftarrow \{l_{i_c}\}_{c=1}^{C}$ with $|l_{i_c} - l_{i_{c+1}}| > \chi$
  end if
end for

In Fig. 6.4, we show the interest of adopting an example-based adaptive image partition. Note that for 8^2 patch dimensionality, the pointillist texture feature is not captured, while for 32^2 patch dimensionality the style is better captured at the cost of having poor reconstruction of structures present in the original image. When a classic adaptive partition is used (based on the variance of the original image), style transfer is reasonably achieved, but entire structures are also copied from the example image (the woman's face in the painting). On the other hand, when the "Split and Match" adaptive partition is used, it leads to a convincing synthesis of the example style, while structures in the original image are well preserved.

6.3.3 Markov Random Fields modeling

In the spirit of [Freeman 2000], we consider here a problem formulation for style transfer as a patch-based Markov Random Field. Within this formalism, the problem of example-based style transfer can be solved by computing the Maximum a Posteriori from a well chosen joint probability distribution on all image units (quadtree patch labels in our model).

Usually, patch-based MRF models such as in [Freeman 2000] are computed over a graph in a regular grid, as illustrated in Fig. 6.5a. In this work, we rather propose a MRF model over an adaptive partition, as shown in Fig. 6.5b. Nevertheless, the neighborhood definition in the proposed quadtree MRF is analogous to a 4-neighborhood in a regular grid.



a) Regular grid b) Adaptive quadtree

Figure 6.5: a) MRF for low-level vision problems over a regular grid. Nodes in the bottom layer represent image units from the observed scene, while nodes in the top layer represent hidden image units that we search to estimate through inference. The vertical edges represent data fidelity terms, while the horizontal edges represent pairwise compatibility terms. b) MRF over an adaptive image partition.

As discussed in Sec. 6.3.2, for a quadtree patch $p^u_{\boldsymbol{x}_i}$, we first compute a set of K candidate labels $L_i = \{l_{i_k}\}_{k=1}^K$ as a strategy to reduce the dimensionality of the labeling problem. We consider now an inference model to compute the most likely set of label assignments for all the patches in R, where labels represent patch correspondences between u and v. More precisely, we search for the set of label assignments $\hat{L} = \{\hat{l}_i\}_{i=1}^n$ maximizing the probability density

$$P(L) = \frac{1}{Z} \prod_i \phi(l_i) \prod_{(i,j)\in\mathcal{N}} \psi(l_i, l_j), \tag{6.6}$$

where Z is a normalization constant, ϕ is the data fidelity term

$$\phi(l_i) = exp(-d[p^u_{x_i}, p^v_{l_i}]\lambda_d) \tag{6.7}$$

and $\psi(l_i, l_j)$ is a pairwise compatibility term between neighboring nodes i and j ($(i,j) \in \mathcal{N}$ means that R_i and R_j are neighbors in Ω_u)

$$\psi(l_i, l_j) = exp(-d[\bar{p}^v_{l_i}, \bar{p}^v_{l_j}]\lambda_s + |l_i - l_j|^2\lambda_r), \tag{6.8}$$

with λ_d, λ_s and λ_r three positive weights (respectively fixed to 2, 2 and 1 in all experiments). This function ψ is composed of a smoothness term and a term penalizing label repetitions.

In patch-based MRFs, the compatibility term ensures that neighbor candidate patches are similar in their overlapping region. To define this properly, we first extend each region R_i of the partition R by $\tau_i\theta$ in each direction (θ is an overlapping ratio set to 0.5 in practice). This permits to define an overlap between two neighboring extended regions $\bar{R}_i$ and $\bar{R}_j$. The term $d[\bar{p}^v_{l_i}, \bar{p}^v_{l_j}]$ in $\psi(l_i, l_j)$ is the distance between the corresponding extended patches in v over this intersection $\bar{R}_i \cap \bar{R}_j$.

While we search for smooth intensity transitions in the overlapping part of neighbor candidate patches, we also aim to penalize two neighbor nodes to have exactly the same label, thus we encourage $|l_i - l_j|^2$ to be large as a strategy to boost local synthesis variety.

Note that computing an exact MAP inference to solve directly Eq. (6.6) is an *intractable combinatorial problem* due to the high dimensionality of image based graphical models, but approximate solutions can be found by iterative algorithms. We adopt in this work the *Loopy Belief Propagation* method [Weiss 1997] [Pearl 1988].

In practice, we do not maximize the density (6.2) but rather minimize its negative logarithm for computational convenience. Converting the MAP inference into an energy minimization problem has two implementation advantages: it avoids the computation of exponentials and allows to represent energies with integer type.

Thus, we search to minimize

$$E(L) = \sum_i \phi'(l_i) + \sum_{(i,j)\in\mathcal{N}} \psi'(l_i, l_j), \tag{6.9}$$

where $\phi'(l_i) = -\log \phi(l_i)$ and $\psi'(l_i, l_j) = -\log \psi(l_i, l_j)$. In practice, the normalization constant Z can be dropped when passing from a MAP to an energy formulation. The method we employ to minimize Eq. 6.9 is presented in Algorithm 3.

Algorithm 3 Loopy belief propagation for style transfer

$\forall(i,j) \in \{1, ..., n\}$ initialize message $m_{ij} \leftarrow 0$
while stopping condition is not achieved **do**
  for each region $R_i \in R$ **do**
    $\boldsymbol{x}_i \leftarrow$ center of R_i
    for each candidate label $l_i \in L_i$ **do**
      $\phi'(l_i) \Leftarrow d[p^u_{\boldsymbol{x}_i}, p^v_{l_i}]\lambda_d$
      for each node j which is neighbor to i **do**
        for each candidate label $l_j \in L_j$ **do**
          $\psi'(l_i, l_j) = d[\bar{p}^v_{l_i}, \bar{p}^v_{l_j}]\lambda_s - |l_i - l_j|^2\lambda_r$
        end for
      end for
    end for
    Update the incoming message at node i:
    $m_{i,j}(l_i) = \min_{[l_i \in L_i]} \psi'(l_i, l_j) + \phi'(l_i) + \sum_{k\in\mathcal{N}_i\setminus\{j\}} m_{k,j}(l_i)$
    Update the optimal label at this iteration:
    $\hat{l}_i = \arg\min_{[l_i\in L_i]} \phi'(l_i) + \sum_{j\in\mathcal{N}_i} m_{i,j}(l_i)$
  end for
end while

In most of our experiments with loopy belief propagation, we observed that a few iterations are enough to achieve convergence. Hence, we set the maximum number iterations to 10 as stopping condition for Algorithm 3.

Finally, after we have computed the set of optimal labels, a patch in the reconstructed image $\hat{u}$ with estimated label $\hat{l}_i$ is given by $p^{\hat{u}}_{\boldsymbol{x}_i} = p^{v}_{\hat{l}_i}$.

6.3.4 Bilinear blending

Although we compute label correspondences that are likely to be coherent across overlapping regions, seams can still be noted in the reconstructed image $\hat{u}$ across the quadtree patch boundaries. In order to remove visible seams we apply an effective method inspired on linear alpha blending. Note that in an overlapping quadtree, a variable number of patches may overlap. Then, a pixel $\boldsymbol{x}$ in the final reconstructed image $\tilde{u}(\boldsymbol{x})$ is defined as a linear combination of the S overlapping patch intensities at $\boldsymbol{x}$:

$$\tilde{u}(\boldsymbol{x}) = \sum_{s=1}^{S} \alpha_s(\boldsymbol{x})\, \bar{p}^{\hat{u}}_{\boldsymbol{x}_s}(\boldsymbol{x})\,, \text{ where } \alpha_s(\boldsymbol{x}) = \frac{\delta(\boldsymbol{x}, \partial\bar{p}^{\hat{u}}_{\boldsymbol{x}_s})}{\sum_{s=1}^{S} \delta(\boldsymbol{x}, \partial\bar{p}^{\hat{u}}_{\boldsymbol{x}_s})} \tag{6.10}$$

is the weighting factor and $\delta(\boldsymbol{x}, \partial\bar{p}^{\hat{u}}_{\boldsymbol{x}_s})$ is the normalized closest distance between pixel $\boldsymbol{x}$ and the patch border $\partial\bar{p}^{\hat{u}}_{\boldsymbol{x}_s}$:

$$\delta(\boldsymbol{x}, \partial\bar{p}^{\hat{u}}_{\boldsymbol{x}_s}) = \frac{|\boldsymbol{x} - \partial\bar{p}^{\hat{u}}_{\boldsymbol{x}_s}|^2}{\tau_s^2}. \tag{6.11}$$

This blending strategy ensures smooth transitions between neighbor patches at a low computational cost.

6.3.5 Global color and contrast transfer

We have described in the previous subsections our strategy for texture transfer through an adaptive patch-based approach. Now, we consider that color and contrast are two features in style that may be consistently modeled as global transformations.

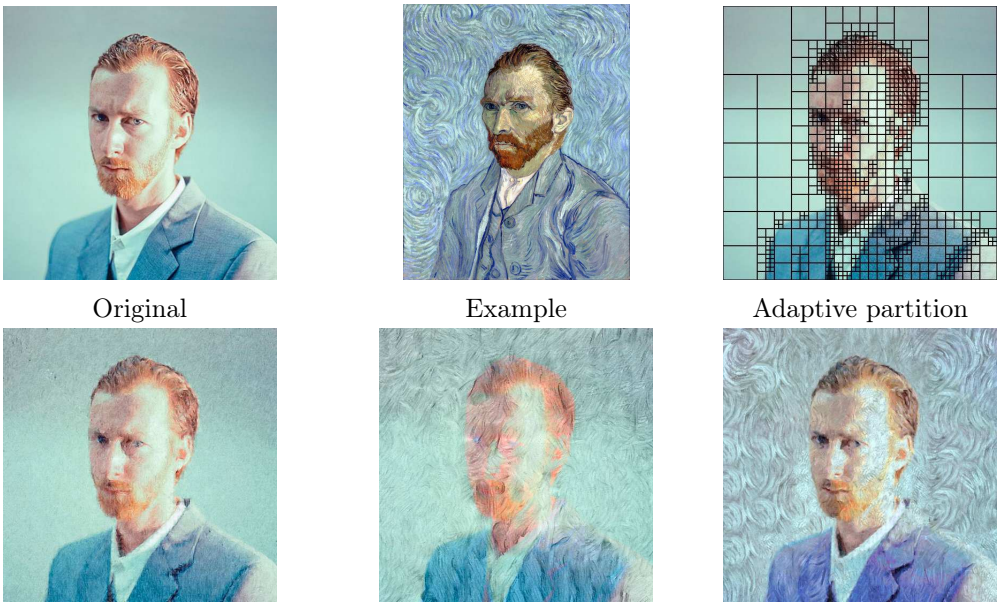
That said, we apply the color transfer method proposed in our previous work [Frigo 2014] to match consistently the color palettes of the original and example images. This color transformation is combined with a global contrast transformation achieved by a parametric histogram specification. In particular, classical histogram specification between images may lead to visual artifacts, thus we approximate the histogram specification curve to a power law model through least squares fitting. That ensures that the contrast is transferred globally without creating artifacts.

6.3.6 Experiments

We present here a number of experiments performed with our method. In Fig. 6.7, we present a comparison of our algorithm and two state-of-the-art style transfer methods. The first method, called PatchTable [Barnes 2015], is originally applied for supervised style transfer, but since we do not have a pair of example unfiltered and filtered images (as used in [Hertzmann 2001]), we apply their method[4] in an

[4] We use the code provided in the author's page

unsupervised setting by assuming that the unfiltered and filtered images are the same. In their result, Van Gogh's typical brushwork is transferred at some point, but the complete painting style is poorly recreated. Also the overall structures of the original image are lost. The second method is the Neural Artistic Style [Gatys 2015], based on CNN. While the color palette of the example image is well preserved, the texture is not well recreated in their results. For example, the texture in the sky differs considerably from the snail shapes in Van Gogh's painting. Our result outperforms state-of-the-art methods, capturing the local image texture, color and contrast.



Original | Example | Adaptive partition

Result: regular MRF, $\tau_i := 8$ | Result: regular MRF, $\tau_i := 32$ | Result: adaptive MRF

Figure 6.6: Illustration of *unsupervised* style transfer based on MRF models over different spatial partitions. It can be seen that a regular partition either does not capture the style when patches have small size ($\tau_i := 8$) or does not preserve structures when patches have large size ($\tau_i := 32$). On the other hand, an adaptive partition results in a convincing and structure-preserving style transfer.

Finally, in Fig. 6.8, we show style transfer with our method for two different painting styles. With our adaptive quadtree partition each original image has a different partition particularly adapted to the given example. Thanks to this strategy, our algorithm accomplishes to transfer the appearance independently of the scale texture of the example or the geometry of the original image. We believe that a good style transfer implies to reproduce the texture and the color palette of the example image, as we have already mentioned. However, other artistic choices are possible depending on the desired results. For example, some application may request to only transfer texture and keep the colors of the original image. Fig. 6.9 and Fig.

Original and Example

Our method

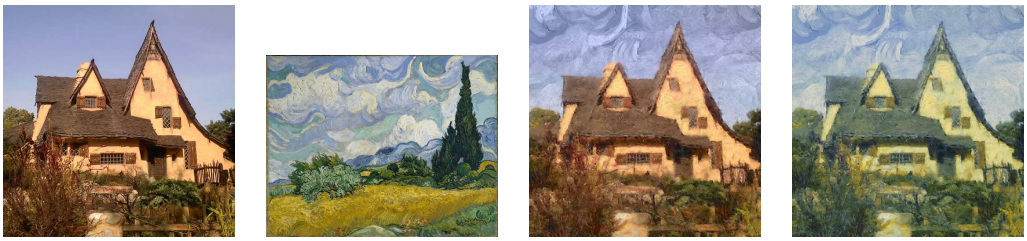
Unsupervised Patch table [Barnes 2015]

Neural Artistic Style [Gatys 2015]

Figure 6.7: Comparison with state-of-the-art. It can be observed that our method captures the prominent texture and color features from Van Gogh's painting, with an overall accurate reconstruction of buildings. The method of [Barnes 2015] captures partially the brushwork texture from the painting, but the buildings are not well reconstructed. The method of [Gatys 2015] captures accurately the painting colors, however it does not reconstruct all main structures in the original image (buildings in bottom left), and the brushwork textures are not noticeable in the result image, which has a rather blurry effect in the sky.

Figure 6.8: Illustration of our example-based style transfer for different painting styles. Example images from van Gogh's and Seurat's are on the top and original images are on the left. Our algorithm transfers successfully the style for different texture scales and it preserves the image geometry of the original images.

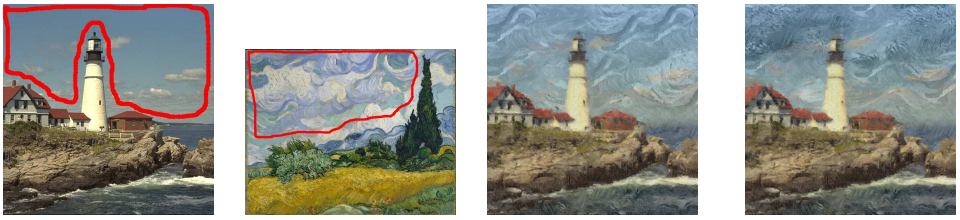
C.6 show our results when transferring or not the global color.



Source image Example image Texture transfer Texture + color

Figure 6.9: Illustration of only texture versus texture and color transfer. Both results are interesting depending on the artistic choice.

In the experience of Figure 6.10, we show a possible application of user interactivity, in which we add semantic value to style transfer. For example, an user may want that the style in the sky of an example image is transferred to the sky of an input image. In practice, we take into account the matching regions selected by the user, by adapting our optimal labeling problem. More precisely, we add a penalization cost to the data term computer over all patches belonging to non-circled regions.



Input image Example image With constraint Without constraint

Figure 6.10: Texture transfer guided by user interactivity. In this experiment, we take into account user strokes to guide region matching. Regions circled by user in the input and in the example images are enforced to be matched for style transfer. In the example, an user manually selects the sky in the input and in the example image. It can be noted that style transfer without user constraints may transfer to the sky of resulting image some textures that are present in the fields in the example image. On the other hand, when user guide is taking into account, the texture in the sky of result is mostly the same as the texture of the sky's painting. We only perform texture transfer in this example, without color transfer.

Finally, an illustration of the effects of Belief propagation in style transfer is presented in Fig. 6.11, and additional results of our method applied with different original and example images can be seen in Figs C.2 to C.5

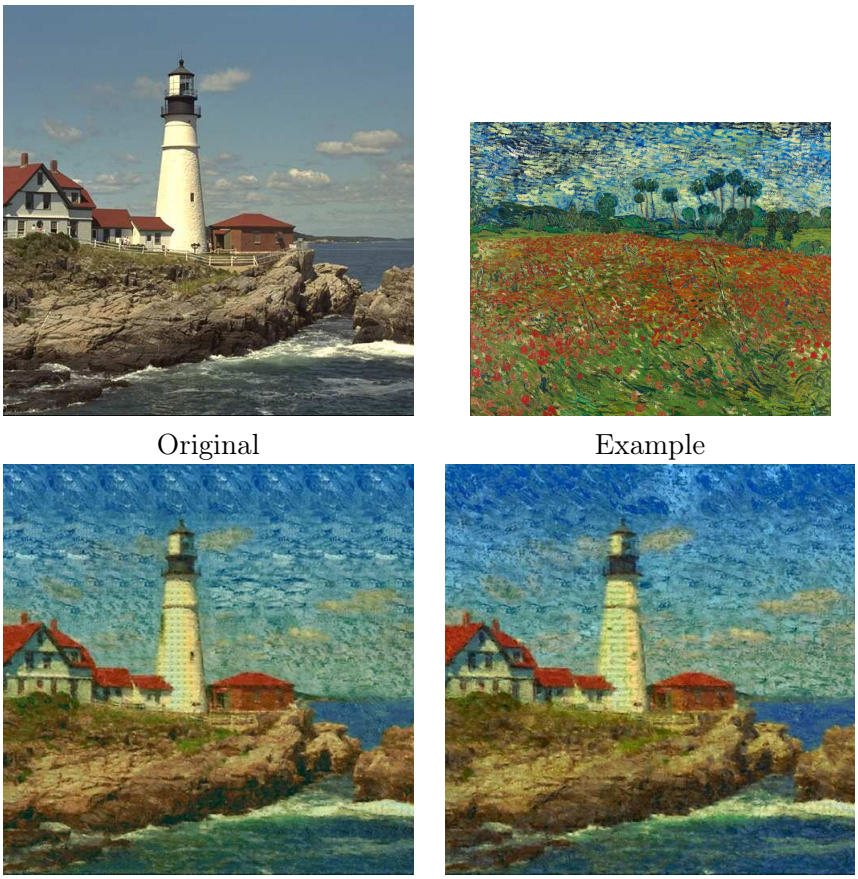


Figure 6.11: Illustration of solving style transfer by Belief Propagation. The image on the bottom left is reconstructed by taking independently the best matching patches in the example image. It can be seen that the bottom left image contains considerable patch repetitions, notably in the top part of the image. In the bottom right image, we solve a patch labelling problem by belief propagation, considering a smoothness term and a term penalizing label repetitions. It can be seen in the bottom right image that our method synthesizes a stylized image that could be argued as looking natural, since it is more regular and non-repetitive than the image in bottom left.

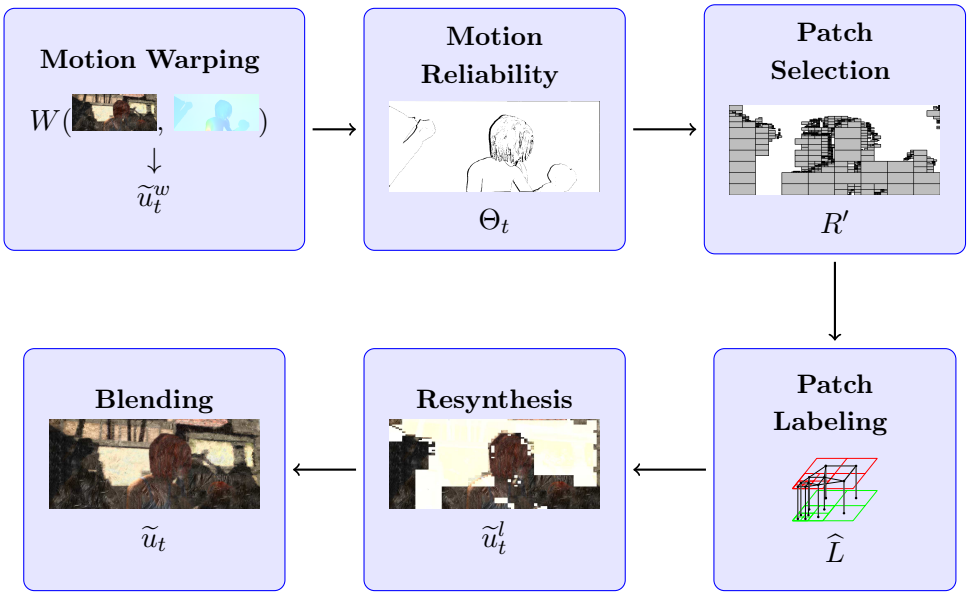


Figure 6.12: Overview of our proposed method for Temporal Style Propagation (TSP). In this flowchart, we present the main steps performed at each iteration of style propagation at time t. First, **motion warping** is applied to the previously stylized frame $\widetilde{u}_{t^\star}$, resulting in output $\widetilde{u}_t^w$. Second, **motion reliability** is computed by taking into account the optical flow accuracy and occlusion map. A pixel with coordinate $\boldsymbol{x}$ is considered unreliable if $\Theta_t(\boldsymbol{x}) = 0$ (pictured in black in the flowchart). **Patch selection** takes all patches lying on coordinates of non-reliable motion as patches to be resynthesized. In the sequence, patches are **resynthesized** by style transfer, solving an optimal labeling problem and obtaining relabeled image $\widetilde{u}_t^l$. Finally, the resulting style propagation $\widetilde{u}_t$ is obtained by **blending** $\widetilde{u}_t^w$ with $\widetilde{u}_t^l$.

6.4 Video Style Transfer

In Section 6.3, we presented our approach for unsupervised style transfer of images. A naive approach to extend our style transfer method to videos would consist in simply applying an independent stylization to each frame of a sequence. However, strong texture flickering occurs using this approach, even if neighbor frames share most of its geometry and color.

We propose a temporally coherent stylization which takes into account the optical flow of a scene to propagate textures. We consider a keyframe-based approach, where we first apply the algorithm described in Section 6.3 to a set of chosen keyframes, and then the stylization is propagated to neighbor frames. We set a "keyframe rate" r in such a way that we have one keyframe for each second of video (typically we have $r := 25$ for a video with 25 frames per second).

Our strategy for video style transfer consists in performing Temporal Style Propagations (TSP) for "chunks of frames", which are delimited by a "left keyframe" with index k and a "right keyframe" with index $k + r$. Let $U = \{u_t\}_{t=1}^{D}$ be the

sequence of original frames with $u_t : \Omega \to \mathbb{R}^3$ a color image over a discrete domain $\Omega = \{1, .., M\} \times \{1, .., N\}$; and $\widetilde{U} = \{\widetilde{u}_t\}_{t=1}^{D}$ the sequence we wish to obtain from video style transfer of U, where D is the duration of the sequence (number of frames). For a given set of frames $U_{k,k+r} = \{u_t\}_{t=k+1}^{k+r-1}$, we first propagate the textures from stylized left keyframe $\widetilde{u}_k$ in forward direction, then we propagate the textures from $\widetilde{u}_{k+r}$ in backward direction and finally we blend the forward and backward stylized frames by linear interpolation. This stylization by chunks is repeated until the end of sequence, as we summarize in Algorithm 4.

Our method for temporal style propagation is computed as a combination of optical flow warping and optimal patch labeling solved by loopy belief propagation. We note that our formulation of patch candidate selection for temporal style propagation differs to the one used for image style transfer. From the fact that style propagation is based on a stylized keyframe, u_k and $\widetilde{u_k}$ are known, and style transfer can be conducted in the supervised case, in the spirit of [Hertzmann 2001]. In particular, patch sampling for temporal style propagation is performed by nearest neighbor search between patches from u_t and patches from u_k, and the coordinates of best matches are used to sample patches from stylized keyframe $\widetilde{u}_k$.

In the following sections, we give more details of our approach for video style transfer.

Algorithm 4 Video Style Transfer

Input: U, v; parameters: r
Output: $\widetilde{U}$

$k \leftarrow 1$
while $k \leq D$ **do**
  Compute Style Transfer (ST) for u_k and u_{k+r}
  $\widetilde{u}_k \leftarrow ST(u_k, v)$
  $\widetilde{u}_{k+r} \leftarrow ST(u_{k+r}, v)$
  Temporal Style Propagation in forward direction:
  $\varepsilon \leftarrow 1$
  $\widetilde{U}^f_{k,k+r} \leftarrow \text{TSP}(U_{k,k+r}, \widetilde{u}_k, k, \varepsilon)$
  Temporal Style Propagation in backward direction:
  $\varepsilon \leftarrow -1$
  $\widetilde{U}^b_{k,k+r} \leftarrow \text{TSP}(U_{k,k+r}, \widetilde{u}_{k+r}, k+r, \varepsilon)$
  Forward-backward blending
  $\widetilde{U}_{k,k+r} \leftarrow \alpha \widetilde{U}^f_{k,k+r} + (1-\alpha)\, \widetilde{U}^b_{k,k+r}$
  $k \leftarrow k + r$
end while

6.4.1 Temporal Style Propagation

We assume in this section that $\widetilde{u}_k$, resulting from style transfer applied to keyframe u_k, is known, where k is the index of a keyframe. To achieve temporal style propa-

gation, our aim is to compute $\widetilde{u}_t$ from the original frame u_t, by imposing temporal coherence between $\widetilde{u}_t$ and the previously stylized frame $\widetilde{u}_{t^\star}$. Note that the formalism for temporal style propagation introduced in this section applies for both forward and backward style propagation. When computing forward propagation at time t, the previously known frame has index $t^\star := t - 1$, while in the backward case, we have $t^\star := t + 1$.

The motion field from frame $u_{t^\star}$ to frame u_t is written as an offset map $\Delta_{t^\star,t}$, and $u_t^w := W(u_{t^\star}, \Delta_{t^\star,t})$ denotes image $u_{t^\star}$ warped by optical flow, where W is a warping operation which consists of shifting Ω by $\Delta_{t^\star,t}$ and applying bicubic interpolation for non integer coordinates. Optical flow is estimated using "DeepFlow", a state-of-the-art method proposed by [Weinzaepfel 2013], which combines a variational approach with descriptor matching to obtain a dense offset map from a pair of images.

In order to compute $\widetilde{u}_t$, we can consider $\widetilde{u}_t^w := W(\widetilde{u}_{t^\star}, \Delta_{t^\star,t})$ as an initial estimate. Evidently, warping the previously stylized frame $\widetilde{u}_{t^\star}$ by optical flow encourages temporal coherence between neighbor frames $\widetilde{u}_{t^\star}$ and $\widetilde{u}_t$. In practice, optical flow estimation is not always reliable, and motion vectors are not determined for some areas (occlusions, domain boundaries). Thus, motion warping alone is not sufficient to achieve temporally coherent style propagation.

Our key idea to address the problem mentioned above is to recompute style transfer only for the regions where optical flow is not reliable. To determine such regions, we compute a motion accuracy and an occlusion map, and we combine them into a single reliability map. An overview of our approach is depicted in Figure 6.12.

Motion accuracy map can be computed by taking the absolute error between the original frame and the motion warped frame. We compute a binary accuracy map given by

$$A_t(\boldsymbol{x}) = \begin{cases} 1, & \text{if } |u_t(\boldsymbol{x}) - u_t^w(\boldsymbol{x})| < \tau_e, \\ 0, & \text{otherwise.} \end{cases}$$

where τ_e is the error tolerance, set in practice to 25. According to [Sundaram 2010], a simple and effective approach to compute occlusions is to take the residual between the forward and the backward optical flow:

$$O_t(\boldsymbol{x}) = \begin{cases} 1, & \text{if } |\Delta_{t^\star,t}(\boldsymbol{x}) + \Delta_{t,t^\star}(\boldsymbol{x})|^2 < \tau_m, \\ 0, & \text{otherwise.} \end{cases}$$

where $\tau_m := 0.01(|\Delta_{t^\star,t}|^2 + |\Delta_{t+1,t}|^2) + 0.05$ as proposed by [Sundaram 2010]. Finally, the optical flow reliability map is given by

$$\Theta_t = A_t \odot O_t, \tag{6.12}$$

where $\odot$ denotes the Hadamard (element-wise) matrix product. We consider that if $\Theta_t(\boldsymbol{x}) = 0$, optical flow is not reliable at $\boldsymbol{x}$, and style transfer needs to be recomputed for this coordinate. Thus, the set of coordinates to be resynthesized is given by

$$\chi_t = \left\{ \boldsymbol{x} \in \Omega \,\middle|\, \Theta_t(\boldsymbol{x}) = 0 \right\}. \tag{6.13}$$

In order to synthesize style for the set of coordinates χ_t, we rely on style transfer by adaptive patch sampling, in similar fashion to the labeling problem we describe in Section 6.3. We first compute an adaptive patch partition $R = \{R_i\}_{i=1}^{n}$ for image u_t, then we divide the partition in two sets of patches $R = R^{'} \cup R^{''}$, where $R^{'}$ contains the set of patches to be relabeled, these patches lying on coordinates contained in χ_t and $R^{''}$ contains the patches which are considered as already labeled (style is inferred from motion propagation). Formally, a quadtree patch R_i is added to R' if $R_i \cap \chi_t \neq \emptyset$. In other words, if one or more pixels in a patch have no reliable optical flow, the entire patch will be included in R'.

Now, an important difference in approach between our image style transfer and our video style transfer, is that for the later, from the fact that we know a stylized keyframe, we can pose the problem as a supervised style transfer, in similar fashion to image analogies [Hertzmann 2001]. We know both u_k and $\widetilde{u}_k$, which can be seen as an example of "style transformation", then given u_t, we search for $\widetilde{u}_t$. In other words, for patches $u_t(R_i^{'})$ for all $R_i^{'} \in R^{'}$, we synthesize style taking the keyframe $\widetilde{u}_k$ as an example image. On the other hand, for patches $u_t(R_i^{''})$ for all $R_i^{''} \in R^{''}$ we can rely only on motion warping from $\widetilde{u}_t$.

In particular, it is worthy noting that in the first iteration of the algorithm, we have $t^{\star} := k$, thus $\widetilde{u}_{t^{\star}} = \widetilde{u}_k$, meaning that we only have the stylized keyframe $\widetilde{u}_k$ as example. From the second iteration, we have both stylized keyframe $\widetilde{u}_k$ and the previously stylized frame $\widetilde{u}_{t^{\star}}$ as two different sources for propagating style. We illustrate the first three iterations of our method for forward style propagation in Figure 6.13.

Let $\widehat{L}$ be the set of optimal labels that we want to give to the set of patches $R^{'}$. Then, we formulate the labeling problem as searching for the optimal patch assignment from the style image, given a set of candidate labels. Candidate labels are computed by nearest neighbor search, but differently to the approach described in Section 6.3 where we match input u to example v, for videos we rely on the keyframe as example, thus we match u_t to u_k to search for candidate labels.

Formally, we search for the label set $\widehat{L} = \{\widehat{l_i}\}_{i=1}^{n}$ minimizing the energy

$$E(L) = \lambda_d E_d(L) + \lambda_s E_s(L) + \lambda_t E_t(L), \tag{6.14}$$

where E_d corresponds to the data fidelity term, E_s is the spatial smoothness term, and E_t is the temporal coherence term and $\lambda_d, \lambda_s, \lambda_t$ are the correspondent weights of each term. Similarly to our approach for image style transfer, we minimize the energy in Eq. 6.14 with Loopy Belief Propagation. The three terms of Eq. 6.14 are defined as follows:

1. The **data fidelity** term is given by

$$E_d(L) = \sum_{i=1}^{n'} d[p_{l_i}^{u_k}, p_{\boldsymbol{x}_i}^{u_t}], \tag{6.15}$$

where n' is the number of regions in R'. We remind that $p_{\boldsymbol{x}_i}^{u_t}$ denotes a patch from image u_t centered at $\boldsymbol{x}_i$ and is equivalent to write $u_t(R_i^{'})$. We consider that if patches $p_{\boldsymbol{x}_i}^{u_t}$ and $p_{l_i}^{u_k}$ are similar in structure, then patch $p_{\boldsymbol{x}_i}^{\widetilde{u}_t}$ should be similar in style to $p_{l_i}^{\widetilde{u}_k}$. It can be noted that this data term differs from the one used in our image style transfer. For videos, we perform patch matching between non-stylized images u_t and u_k, while in Section 6.3.3 we proposed patch matching between non-stylized image u and example style image v. This is motivated by the fact that u_t is expected to have considerable overlapping content with u_k, which facilitates matching patches between these images.

2. The **spatial smoothness** term is given by

$$E_s(L) = \sum_{(i,j)\in\mathcal{N}} d[\bar{p}_{l_i}^{\widetilde{u}_k}, \bar{p}_{l_j}^{\widetilde{u}_k}]. \tag{6.16}$$

Exactly in the same manner as we propose for image style transfer, we consider that neighbor patches $\bar{p}_{l_i}^{\widetilde{u}_k}$ and $\bar{p}_{l_j}^{\widetilde{u}_k}$ should be similar in their overlapping area, so that we encourage smooth transitions between stylized patches. We remind that $\bar{p}$ denotes the extended patch p, which overlaps neighbor patches.

3. The **temporal coherence** term is given by

$$E_t(L) = \sum_{i=1}^{n'} d[p_{\boldsymbol{x}_i}^{u_t^c}, p_{l_i}^{\widetilde{u}_k}], \tag{6.17}$$

where u_t^c is a combination of two images: $\widetilde{u}_t^w$ (stylized frame $\widetilde{u}_{t^\star}$ warped by optical flow) and u_t (the current frame to be stylized):

$$u_t^c := [\Theta_t \odot \widetilde{u}_{t^\star}^w] + [(1 - \Theta_t) \odot u_t] \tag{6.18}$$

The idea of temporal coherence cost is to encourage, for the coordinates where optical flow is reliable (given by binary map Θ_t), that the intensities of a stylized patch at time t remain similar to the warped intensities from time $t^\star$.

After estimating a set of optimal labels $\widehat{L}$, we compute an image $\widetilde{u}_t^l$, with the bilinear patch blending described in Section 6.3.4. Finally, we obtain a reconstructed image

$$\widetilde{u}_t := G_t \odot \widetilde{u}_t^w + (1 - G_t) \odot \widetilde{u}_t^l, \tag{6.19}$$

where $G_t := G^\sigma * \Theta_t$ is the optical flow reliability map convolved by a gaussian kernel G^σ with standard deviation σ, for spatial blending between the motion warped image $\widetilde{u}_t^w$ and the labeled image $\widetilde{u}_t^l$. Our method for temporal style propagation is summarized in Algorithm 5.

Algorithm 5 Temporal Style Propagation (TSP)

Input: $(U_{k,k+r}, \widetilde{u}_k, k, \varepsilon)$
Output: $\widetilde{U}_{k,k+r}$

$t \leftarrow k + \varepsilon$
while $t \neq k + r$ **do**
    $t^\star \leftarrow t - \varepsilon$
    $\Delta_{t^\star,t} \leftarrow$ optical flow between $u_{t^\star}$ and u_t

    Warp $u_{t^\star}$ and $\widetilde{u}_{t^\star}$:
    $u_t^w \leftarrow W(u_{t^\star}, \Delta_{t^\star,t})$
    $\widetilde{u}_t^w \leftarrow W(\widetilde{u}_{t^\star}, \Delta_{t^\star,t})$

    Compute motion accuracy map:
    $A_t(\boldsymbol{x}) \leftarrow |u_t(\boldsymbol{x}) - u_t^w(\boldsymbol{x})| < \tau_e$
    Compute occlusion map:
    $O_t(\boldsymbol{x}) \leftarrow |\Delta_{t^\star,t}(\boldsymbol{x}) + \Delta_{t,t^\star}(\boldsymbol{x})|^2 < \tau_m$
    Compute optical flow reliability map:
    $\Theta_t \leftarrow A_t \odot O_t$
    Set of coordinates to be resynthesized:
    $\chi_t \leftarrow \left\{ (\boldsymbol{x}) \in \Omega \,\middle|\, \Theta_t(\boldsymbol{x}) = 0 \right\}$

    Select set of patches R' to label by style transfer:
    $R' \leftarrow \{\}$
    for every region $R_i \in R$ **do**
        if $R_i \cap \chi_t \neq \emptyset$ **then**
            $R' \leftarrow R' \cup R_i$
        end if
    end for
    Compute optimal labels $\widehat{L}$ for patches in domain R':
    $\widehat{L} \leftarrow \min E(L)$
    Compute $\widetilde{u}_t^l$
    $\widetilde{u}_t \leftarrow \widetilde{u}_t^l$ blended with $\widetilde{u}_t^w$

    $t \leftarrow t + \varepsilon$
end while

6.4.2 Forward-backward blending

We described in section 6.4.1 our algorithm for temporal style propagation, which is applied for a set of frames $\{u_{k+1}, u_{k+2}, ..., u_{k+r-1}\}$ delimited by a "left keyframe" u_k and "right keyframe" u_{k+r}. For each set of frames, temporal style propagation is performed in a forward and in a backward pass. We denote the stylized sequence resulting from forward style propagation as $\widetilde{U}^f_{k,k+r} = \{\widetilde{u}^f_t\}^{k+r-1}_{t=k+1}$ and the sequence resulting from backward propagation as $\widetilde{U}^b_{k,k+r} = \{\widetilde{u}^b_t\}^{k+r-1}_{t=k+1}$.

We can finally compute $\widetilde{u}_t$ as a blend of the forward and backward pass

$$\widetilde{u}_t = \alpha_t \widetilde{u}^f_t + (1 - \alpha_t)\widetilde{u}^b_t, \tag{6.20}$$

where α_t is the linear temporal weighting factor given by

$$\alpha_t = \frac{k + r - t}{k + r - k}. \tag{6.21}$$

6.4.3 Keyframe coherence

In the previous sections, we assumed that keyframes are stylized independently as a first step of our video style transfer method. In practice, an independent stylization of keyframes will still result in a temporally varying style. This temporal variation of style tends to be smooth, as a result of the forward-backward temporal interpolation described above. This can be pleasant for some observer since textures will change smoothly without flickering, but other observers may prefer a stylization that is temporally stable.

To obtain a temporally stable stylization, coherence between keyframes can be easily obtained with the method we proposed for temporal style propagation. We generalize the approach for keyframe coherence by assuming we have a pair of consecutive keyframes u_k and u_{k+r}, and that we know $\tilde{u}_k$ and we search to estimate $\tilde{u}_{k+r}$. Then, the formalism described in Section 6.4.1 can be applied by setting $u_t := u_{k+r}$ and $\varepsilon := r$. Optical flow between keyframes u_k and u_{k+r} is obtained by composition of all successive pairwise optical flows between u_k and u_{k+r}.

6.4.4 Experiments

In this section, we present the results of some experiments conducted with the proposed video style transfer method. Some results presented in this section (Figs 6.14 and 6.15) were obtained with sequences taken from the Sintel dataset [Butler 2012], which includes ground truth optical flow and occlusions. Thus, for these results, we use the optical flow and occlusion maps provided with the dataset.

It can be noted in Figure 6.14, (in particular on the highlighted red and green rectangles) that our method guarantees temporal coherency of style, while a stylization performed frame by frame results in severe texture variation. In Figure 6.15, we show video stylization with two different example styles. Again, it can be seen that style remains coherent through different frames of the stylized sequence.

Finally, we show in Figure 6.16 a result of our method applied to a real sequence, where optical flow is estimated by "DeepFlow" [Weinzaepfel 2013]. For a better appreciation of our results, we recommend the reader to watch the videos in our project website[5].

6.5 Considerations

In this chapter, we have proposed a new style transfer method that, differently to previous patch-based approaches, is able to synthesize textures independently of their scale. Our results suggest that the decomposition of content and style in artistic images can be achieved with a simple yet efficient adaptive image partition.

Moreover, we have shown that a local texture modeling and a global color transfer strategy leads to convincing and structure-preserving example-based stylization. On the other hand, state-of-the-art style transfer methods are likely to destroy structures at the cost of synthesizing style.

The proposed style transfer method is completely unsupervised, does not require the original and example images to be visually similar and has an energy-based approach to penalize texture repetition.

We also have seen in this chapter that the extension of style transfer from images to videos is not straightforward, since a stylization in a frame by frame basis results in flickering. Thus, we proposed a technique that guarantees a temporally coherent stylization by temporal style propagation from keyframes. For that, we relied on a combination of optical flow warping and style resynthesis. Our results suggest that such a technique is well adapted for stylization of videos. However, it should be noted that the quality of video stylization is strongly dependent on the accuracy of optical flow estimation. Hence, our video style transfer would clearly benefit from advances in the field of optical flow estimation.

At the moment, an unoptimized implementation of our style transfer takes approximately 3 minutes to process a 512^2 image and 30 minutes to stylize a video from sintel dataset of 2 seconds length. A possible future work is to accelerate the multi-scale patch search, and to adapt our method for other patch features (such as CNN filter responses), to take benefits from recent advances in feature matching and description.

[5] http://oriel.github.io/video_style_transfer.html

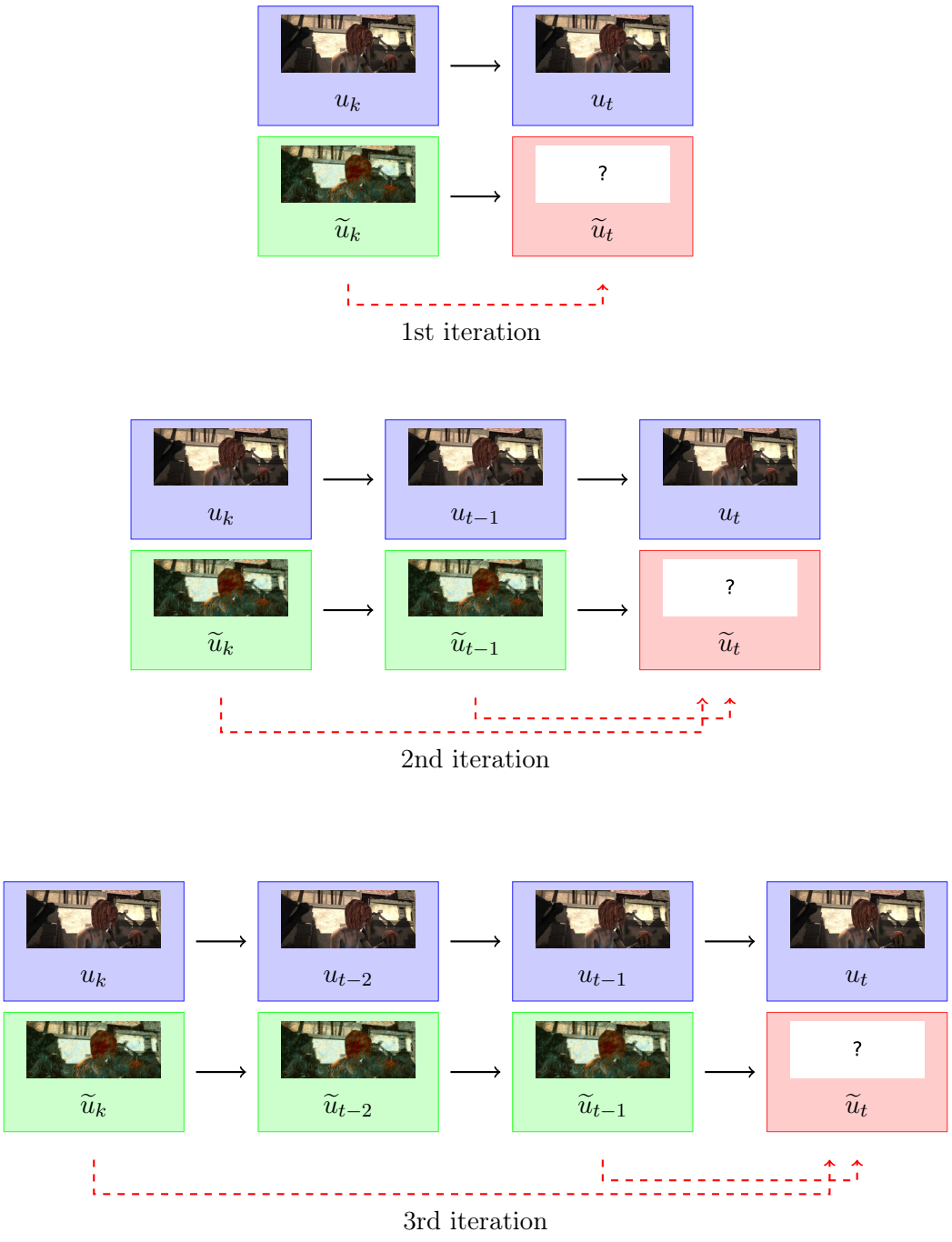


Figure 6.13: **Forward** Temporal Style Propagation. We show on the top row the original frames, and in the bottom row, we show the stylized frames: previously stylized frames are represented in green rectangles, and in red rectangles we represent the stylized frame $\widetilde{u}_t$ that we search to compute through style propagation. Red dashed lines illustrate from which frame texture is being transferred to $\widetilde{u}_t$. It can be noted that in the first iteration, we only have the stylized keyframe $\widetilde{u}_k$ as example. From the second iteration, we use both the stylized keyframe $\widetilde{u}_k$ and the previously stylized frame $\widetilde{u}_{t^\star}$ as example images from where style is transferred.

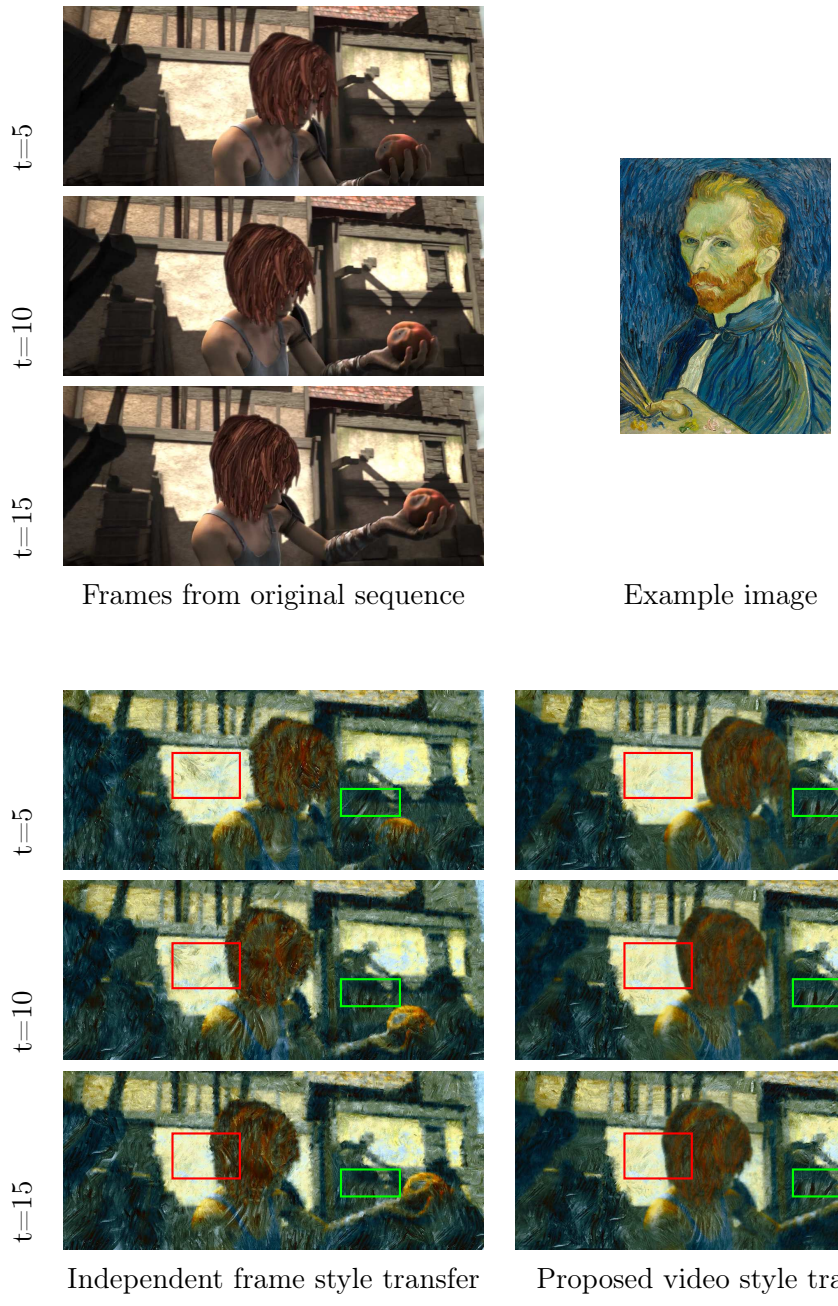


Figure 6.14: Illustration of our temporally coherent video style transfer. It can be noted that the stylized frames in the left column, generated by frame-by-frame stylization, suffer from texture flicker. The brush strokes highlighted by the red and green rectangles change abruptly from a frame to the next. On the right column, we show the stylized frames resulting from our video style transfer method, which has temporally coherent style.

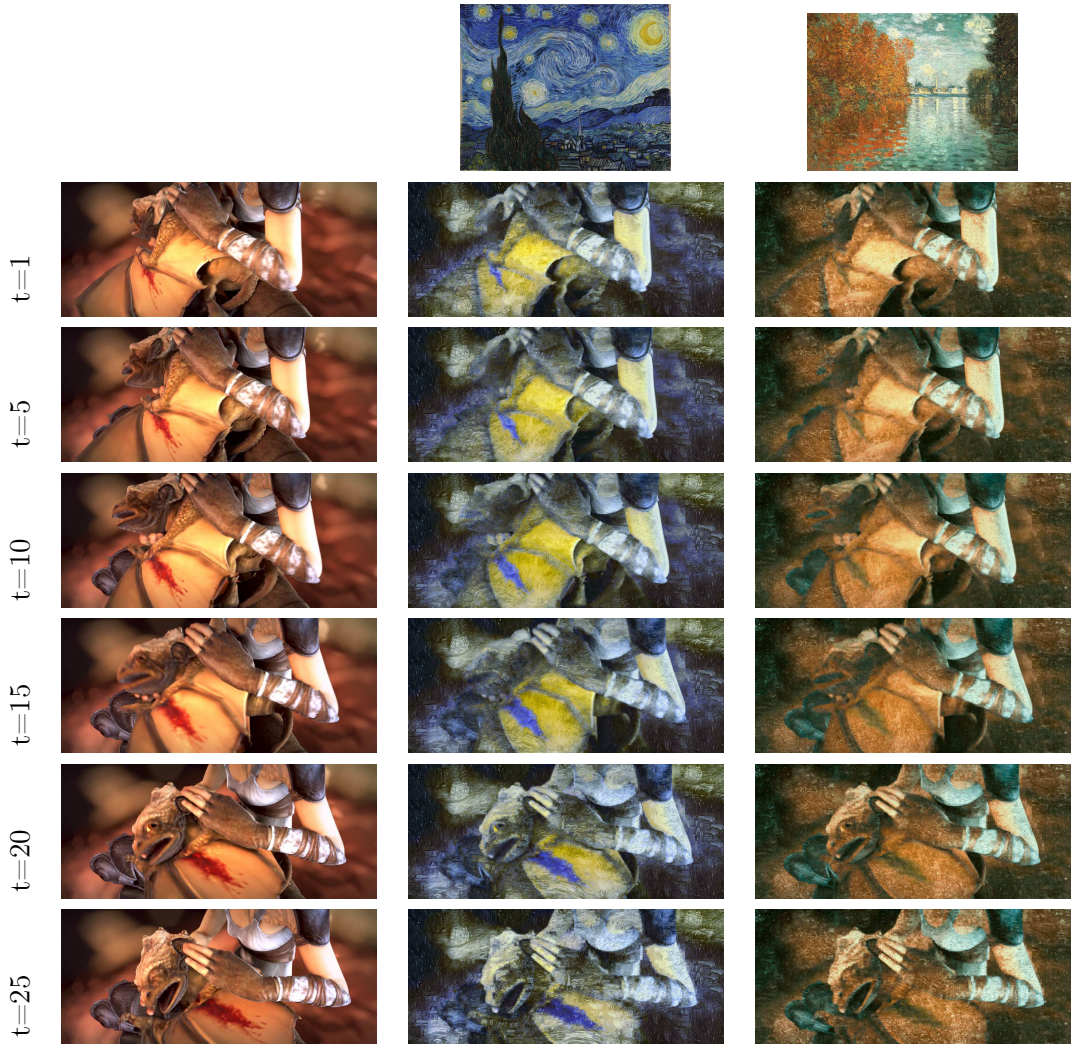


Figure 6.15: Video style transfer results for two different example styles. Top row: example images, Left column: original images. It can be noted that the stylized frames resulting from our video style transfer method have temporally coherent style.

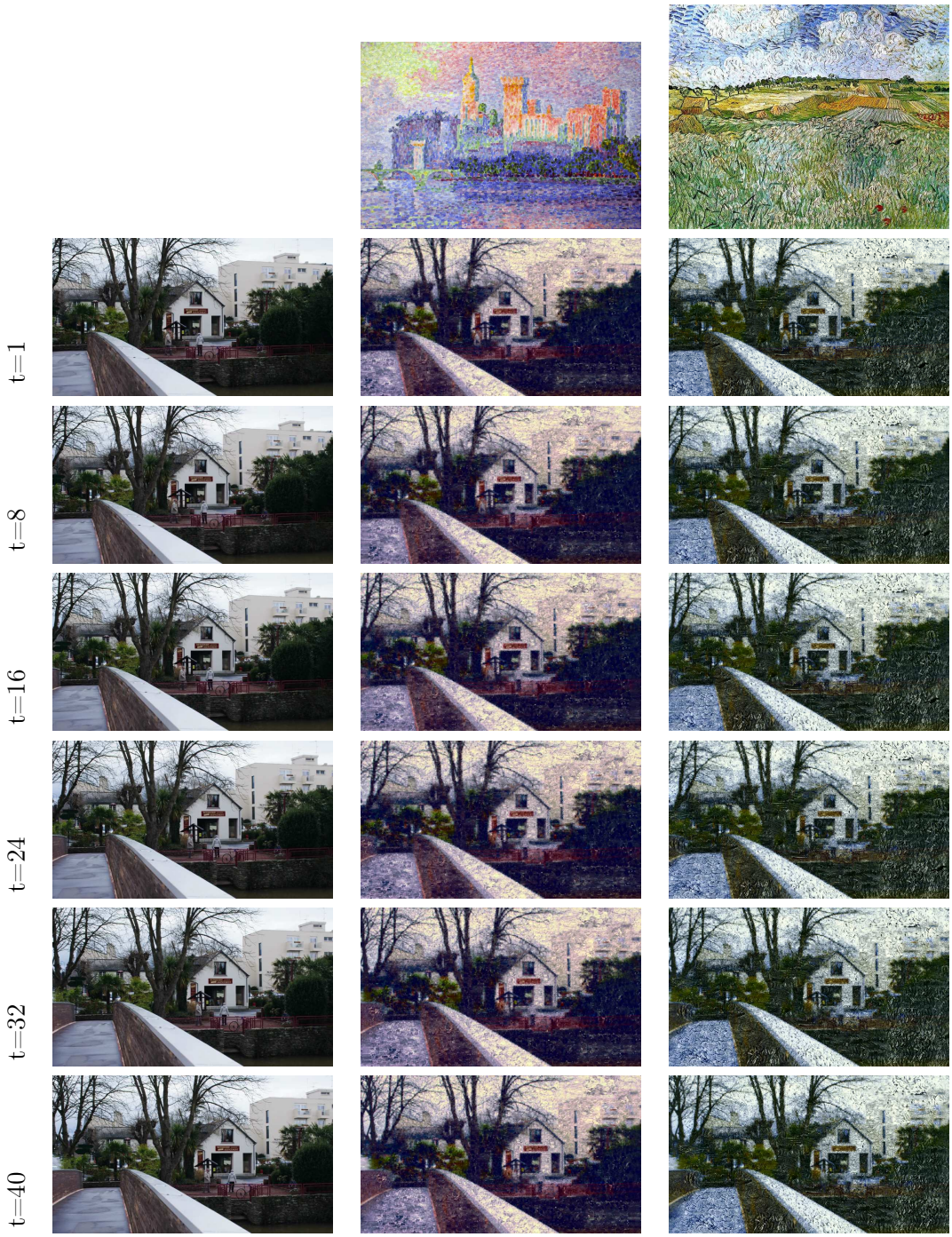


Figure 6.16: Video style transfer results for real sequence. Top row: example images, Left column: original images. Although real sequences offer the challenge of estimating accurate optical flow, our video style transfer produces temporally coherent stylization.

Chapter 7

Conclusion

In this thesis, we have explored different aspects and contributed with new techniques on the general problem of example-based video editing.

In the first part, we have investigated the use of color as an example characteristic for video editing through the related problems of color transfer and video tonal stabilization. In Chapter 4, we have observed that color transfer is a powerful tool to assist color correction, but most previous techniques in the state-of-the-art have the tendency of amplifying image artifacts. Our contribution was to provide a method that reduces the risk of artifact creation by applying a smooth color transformation based on optimal transportation between color palettes.

We have seen in Chapter 5 that an improper application of color transformations to videos may produce tonal instability. Tonal instability artifacts were shown to be created by modern smartphone cameras, due to the temporal instability of automatic white balance and automatic exposure algorithms. We have come up with a new solution to process videos containing tonal instability, by fitting a power law color transformation between frames and keyframes, and taking the motion of the video into account to guide the color correction. Our contribution for tonal stabilization can be seen as an example-based approach, in the sense that keyframes are taken as example images from which color is temporally propagated. In other words, our method encourages that corrected frames are similar to the keyframe in terms of color balance and exposure. Simple yet efficient, the proposed tonal stabilization method has comparable or superior accuracy with respect to the state-of-the-art methods, while being computationally fast to meet the requirements for an embedded smartphone implementation.

In the last part of the thesis, we have explored the possibility of mimicking the style of example images for video editing. Style was approached from the perspective of non-photorealistic rendering, where the aim is to depict images such that they look similar to paintings or drawings. Style was characterized in this thesis as a mixture of color and texture characteristics, so that style transfer could be achieved by a combination of color and texture transfer.

In our discussion, we considered non-parametric patch sampling as a powerful strategy for synthesis of complex textures such as the brush strokes that characterize painting styles. Our main contribution was to show that patch-based texture transfer can largely benefit from the use of an adaptive patch partition (patch sizes adapted to image structures) instead of a regular patch grid (patches having all same size) traditionally used in state-of-the-art patch-based approaches. We have presented texture transfer as a probabilistic labeling problem, where labels are the

coordinates of patches coming from an example image. Finally, we have shown that an extension of style transfer from images to videos is not straightforward, as a frame-by-frame stylization results in severe texture flickering. Similarly to our solution for tonal stabilization in Chapter 5, we relied on keyframes to achieve temporally coherent style transfer. Our main idea for temporally coherent stylization was to rely on optical flow to propagate style from a frame to the next and resynthesize the regions where optical flow is not well defined. Experimental results have shown that this strategy is effective for a convincing stylization of videos that has no visible flickering.

In summary, this manuscript presented new and efficient example-based techniques to address the modification and enhancement of videos. In addition, the presented techniques were mature enough for practical applications, with a relevant impact for multimedia industry.

7.1 Discussion and Perspectives

A number of significant discoveries and perspectives for further research were made during the preparation of this thesis. From the experience that was acquired exploring the problems of color transfer and tonal stabilization, we can make the following observations:

- Our proposed approach for *color transfer* was readily adapted to be used in practice by a film colorist. This adaptation consisted simply in encoding the color transformation computed from our color transfer method in a 3D Look Up Table (LUT). Since most film editing softwares perform color grading based on LUT color processing, the integration of our technique for post-production was straightforward.

- We realized that a first interesting problem associated with color grading is the possibility of combining different complex 3D color transformations into a single one. The problem can be seen as a LUT blending, where we search for a smooth interpolation between different LUTs. A second related problem is how to perform intuitive LUT modification, in such a way that a colorist could edit a 3D color transformation starting from a basic one. To the best of our knowledge, these problems have not been studied in the literature, the closest to that being the work of temporal color transformation smoothing in [Bonneel 2013]. Still, we observe that combining and editing LUTs is both a real necessity for professional colorists and a promising research problem.

- In practice, *color transfer* can be easily extended for videos without color flickering generation, also by encoding the color transformation in a LUT. Based on the fact that color grading professionals prefer to use one LUT per shot of film, color temporal stability is not currently an issue in the context of film editing.

- *Tonal stabilization* for correction of automatic white balance and exposure instability can be efficiently achieved by very simple parametric transformations such as the power law we have shown in Chapter 5. We have observed that in color correction simplicity can sometimes be seen as an advantage for robustness, where parametric transformations with small number of degrees of freedom can reduce the risk of artifact generation and overfitting over color correspondences.

- Still, we believe that one of the main challenges for video *tonal stabilization* is to find a good trade-off between color correction and preservation of some tonal variation, in particular some exposure variation. We have seen in Chapter 5 that a strict color stabilization of a video can be disastrous in practice because of the possible large variations in exposure settings in the same video. The main problem comes from the limitation of camera's dynamic range and the risk is that such a strict tonal stabilization could overexpose or underexpose the whole sequence.

- We suggested in Chapter 5 that stabilizing video exposure variations based on a keyframe could be used to create temporal high dynamic range for videos. Hence, an in-depth investigation of the connection between video tonal stabilization and High Dynamic Range (HDR) image processing can be promising for future work.

As with respect to the *style transfer* problem, we can observe the following:

- Our adaptive patch sampling approach could also be adapted to solve different problems, such as inpainting or super-resolution. Both in inpainting or in super-resolution, large patches could be used to synthesize homogeneous areas in images, while smaller patches could be used to synthesize image details. Moreover, An adaptive patch sampling approach can lead to reduction in computational complexity in comparison with regular sampling of small patches.

- As we have seen in Chapter 6, patch-based approaches are excellent to transfer complex textures from example images. However, we realized that the classical patch matching approach is not able to correctly transfer edge styles, and it may fail to transfer the style from some abstract art. On the other hand, in parallel to our research, [Gatys 2015, Gatys 2016] introduced the style transfer based on convolutional neural nets, which have some intriguing properties. One interesting property we have observed about the "neural style transfer" approach is the capability to synthesize the style of the edges from an example image. This is possibly due to the high level representation of the input and example images taken from convolutional layer responses of a pre-trained neural network. It follows that features (such as edges) that are not similar in intensity seems to be correctly matched, as they are similar in the space of neural feature responses. Therefore, we consider that exploring convolutional

neural networks is an interesting direction to style transfer, in particular for achieving a more abstract stylization that respects less the structures of the original image.

- One possibility to extend our work would be to rely on a neural adaptive patch partition, where patches would be matched according to their feature responses from neural networks. A close idea is the neural patch representation which was recently introduced in [Li 2016], as a midway between CNN and MRF texture modeling.

In conclusion, this thesis has shown that example-based modification of videos is a rich and promising research topic, with many applications and possible directions for future work. We have just contributed with some brush strokes in a vast canvas, and we hope that it can inspire others to continue this work.

Appendix A

Additional Color Transfer Results

A.1 Generic color transfer

Here, we present additional objective and visual comparisons of our method to state-of-the-art "global statistics" color transfer methods (Table A.1, Figures A.1, A.2 and A.3).

Table A.1: Comparison of the SSIM measure [Wang 2004] between input and output images for different color transfer methods, corresponding to Figures A.1, A.2 and A.3. A SSIM value of 1 denotes that no artifacts have been generated after color transfer. Our method creates no artifacts compared to other techniques.

	Fig. A.1	Fig. A.2	Fig. A.3
Reinhard [Reinhard 2001a]	0.5790	0.8927	0.3402
Pouli [Pouli 2011]	0.5881	0.8063	0.4481
Pitié+Rabin [Pitié 2007, Rabin 2011]	0.6138	0.8789	0.5596
Papadakis [Papadakis 2011]	0.5401	0.7198	0.4715
Ferradans [Ferradans 2013]	0.6318	0.7668	0.5586
Our	**0.9598**	**0.9945**	**0.9863**

Figure A.1: Results obtained by state-of-the-art global color transfer techniques compared to our method. Note that state-of-the-art methods [Pouli 2011], [Papadakis 2011], [Pitié 2007, Rabin 2011], [Ferradans 2013] produce washed out colors with reduced contrast. In contrast, both Reinhard's [Reinhard 2001a] and our method give a visually plausible and artifact-free result, but our method arguably better matches the skin tones and the overall yellow/brown example color palette, while Reinhard's result presents pink/red colors that are not visible in the example image.

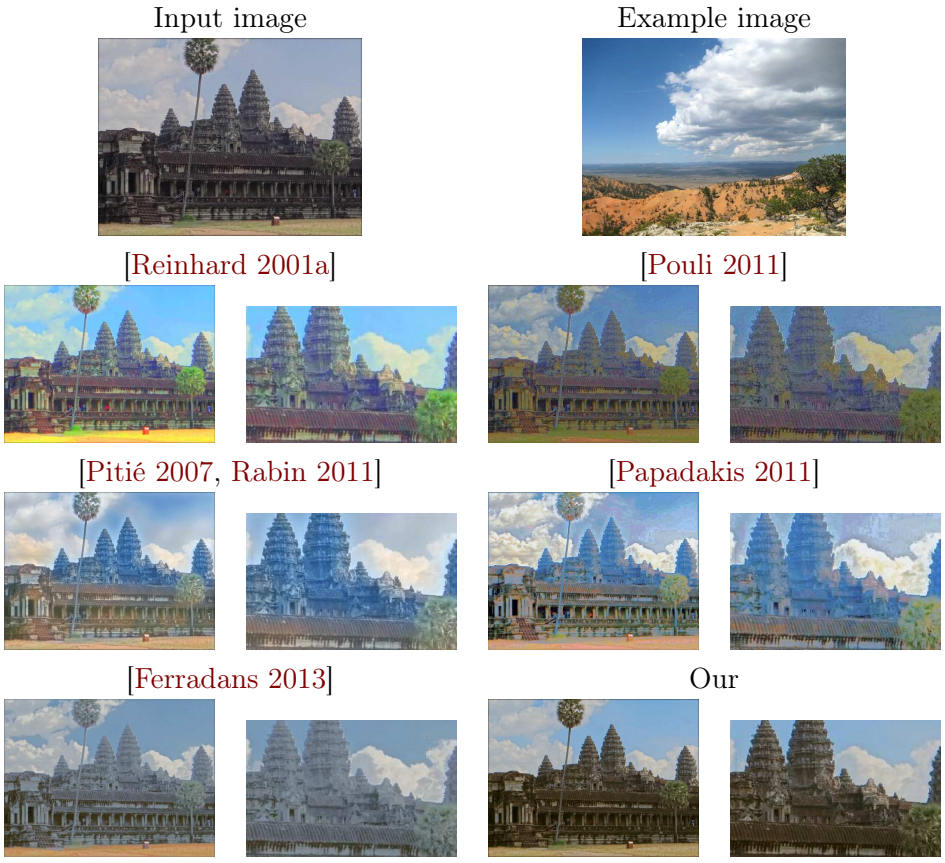


Figure A.2: Results obtained by state-of-the-art global color transfer techniques compared to our method. We also present a zoomed patch for finer visualization of each result. Note that Reinhard's method assign an unnatural yellow color to the ground, while other state-of-the-art methods produce color aberrations in the building. Our method enhances the sky and the building colors, while preserving the fidelity of the input image.

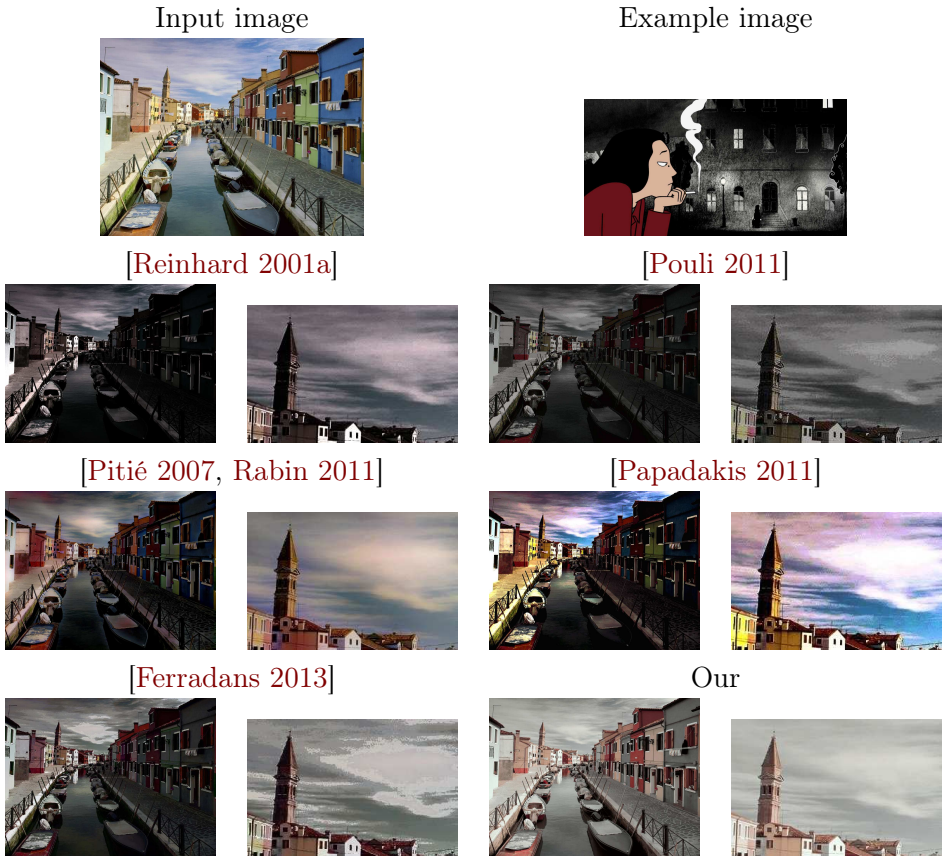


Figure A.3: Results obtained by state-of-the-art global color transfer techniques compared to our method. We also present a zoomed patch for finer visualization of each result. Our result leads to a visually plausible and artifact-free result.

A.2 Comparison to local color transfer

Here, we present additional objective and visual comparisons of our method to the global methods, plus the "correspondence-based" local color transfer method of [HaCohen 2011] (Table A.2, Figures A.4, A.5, A.6 and A.7 where input and example images are from the same scene). Note that the method of [HaCohen 2011] is restricted to image pairs of the same scene, where spatial correspondences are to be found.

Table A.2: Comparison of the SSIM measure [Wang 2004] between input and output images for different color transfer methods, corresponding to Figures A.4, A.5, A.6 and A.7. A SSIM value of 1 denotes that no artifacts have been generated after color transfer. Our method creates no artifacts compared to other techniques.

	Fig. A.4	Fig. A.5	Fig. A.6	Fig. A.7
Reinhard [Reinhard 2001a]	0.7266	0.9126	0.8342	0.9376
Pouli [Pouli 2011]	0.7586	0.8877	0.8602	0.9107
Pitié+Rabin [Pitié 2007, Rabin 2011]	0.7840	0.9468	0.9672	0.9495
Papadakis [Papadakis 2011]	0.6193	0.9064	0.8196	0.9001
Ferradans [Ferradans 2013]	0.6475	0.8754	0.7875	0.8983
HaCohen [HaCohen 2011]	0.9181	0.9362	0.9349	0.9250
Our	**0.9920**	**0.9939**	**0.9947**	**0.9835**

Figure A.4: Results obtained by state-of-the-art global and local color transfer techniques compared to our method. Although [HaCohen 2011] has better recovered the specularity of the dress, our result is less saturated (arguably more natural) on the skin, on the hair and on the background.

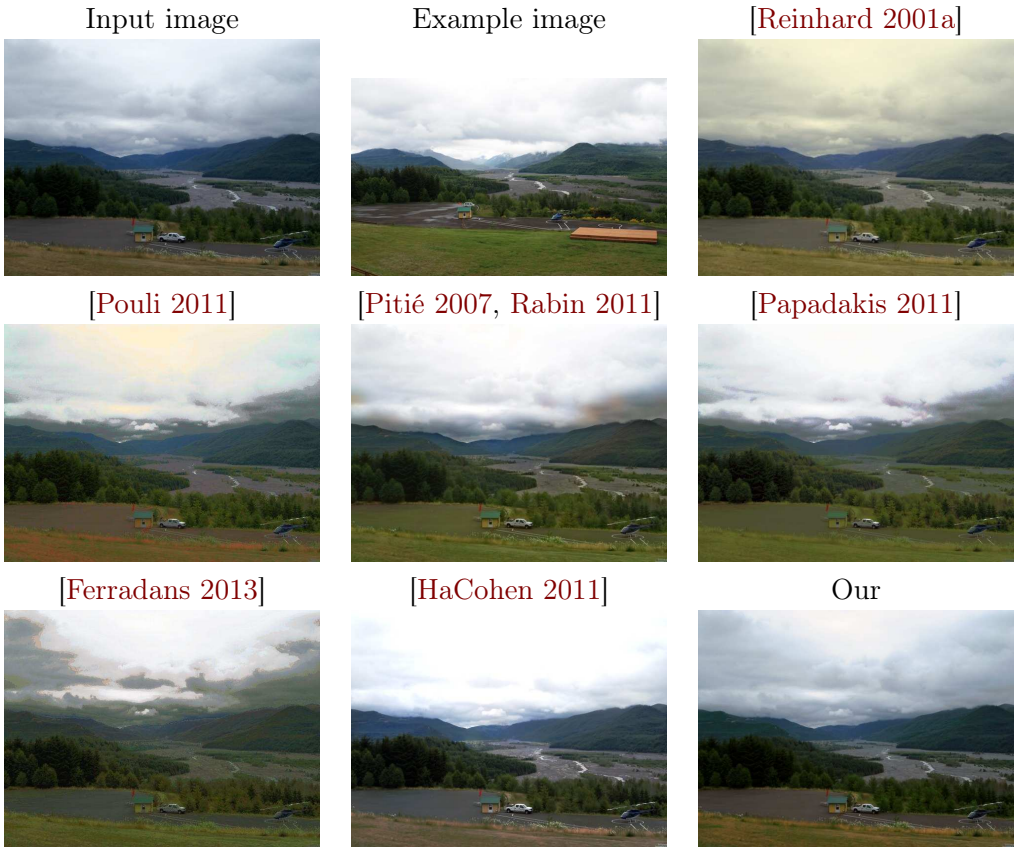


Figure A.5: Results obtained by state-of-the-art global and local color transfer techniques compared to our method. Note that both [HaCohen 2011] and our method give coherent results, but our method better matches the color of the asphalt and preserves the cloud structures in the sky, while the method of [HaCohen 2011] produces an oversaturation effect in the sky.

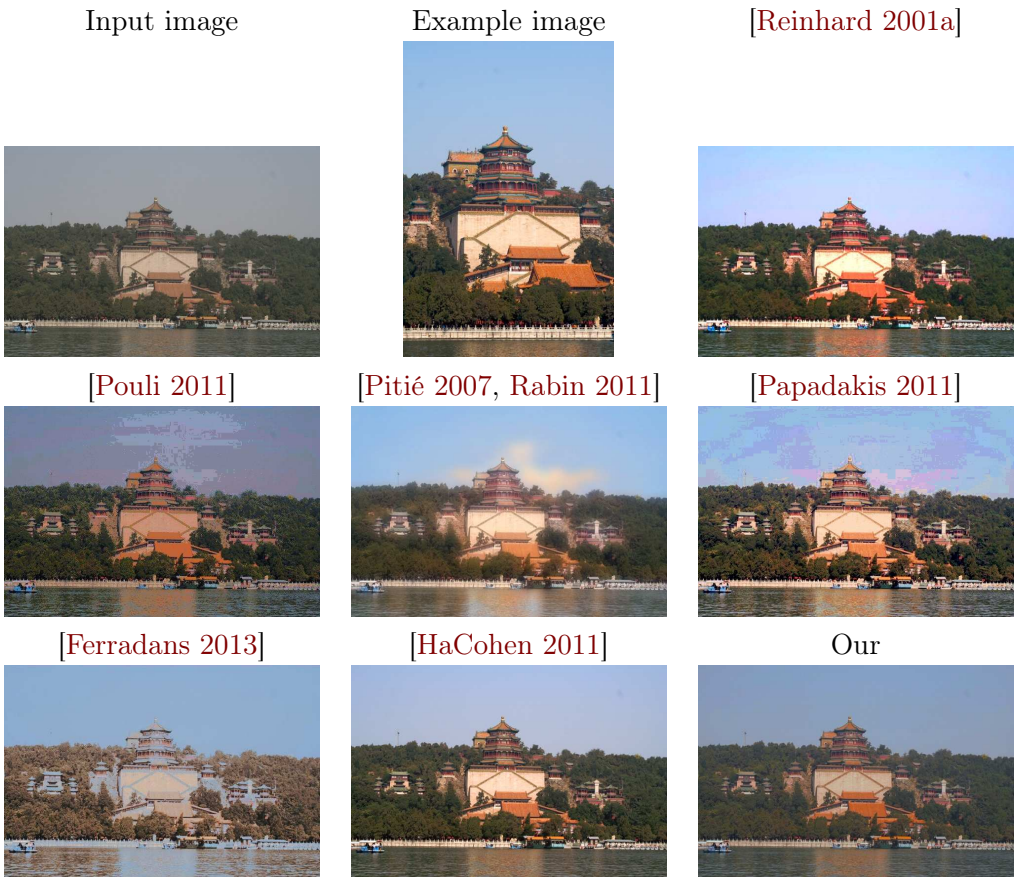


Figure A.6: Results obtained by state-of-the-art global and local color transfer techniques compared to our method. Both [HaCohen 2011] and our method lead to reasonable results, and the objective comparison is difficult. However, [HaCohen 2011] uses spatial correspondences while our method is generic, depending only of color correspondences.

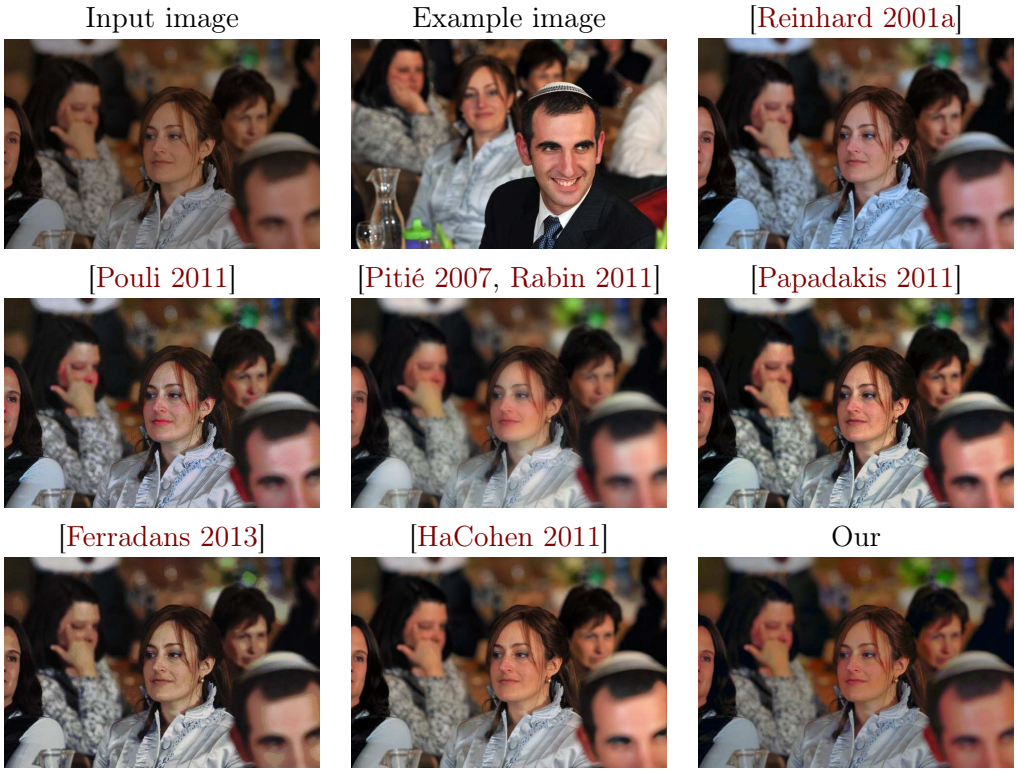


Figure A.7: Results obtained by state-of-the-art global and local color transfer techniques compared to our method. Both [HaCohen 2011] and our method lead to reasonable results, and the objective comparison is difficult. However, [HaCohen 2011] uses spatial correspondences while our method is generic, depending only of color correspondences.

A.3 Constrained color transfer

Here, we present additional results concerning very challenging image pairs, for which the introduction of additional constraints such as saliency improve the result. In addition to the paper, we show the results of other stat-of-the-art methods and present the SSIM measure.

Table A.3: Comparison of the SSIM measure [Wang 2004] between input and output images for different color transfer methods, corresponding to Figures A.8 and A.9. A SSIM value of 1 denotes that no artifacts have been generated after color transfer. Our method creates no artifacts compared to other techniques.

	Fig. A.8	Fig. A.9
Reinhard [Reinhard 2001a]	0.9495	0.9860
Pouli [Pouli 2011]	0.9448	0.8940
Pitié+Rabin [Pitié 2007, Rabin 2011]	0.9245	0.9252
Papadakis [Papadakis 2011]	0.9269	0.8653
Ferradans [Ferradans 2013]	0.9285	0.8736
Our method (unconstrained)	0.9646	**0.9865**
Our method (saliency driven)	**0.9683**	0.9817

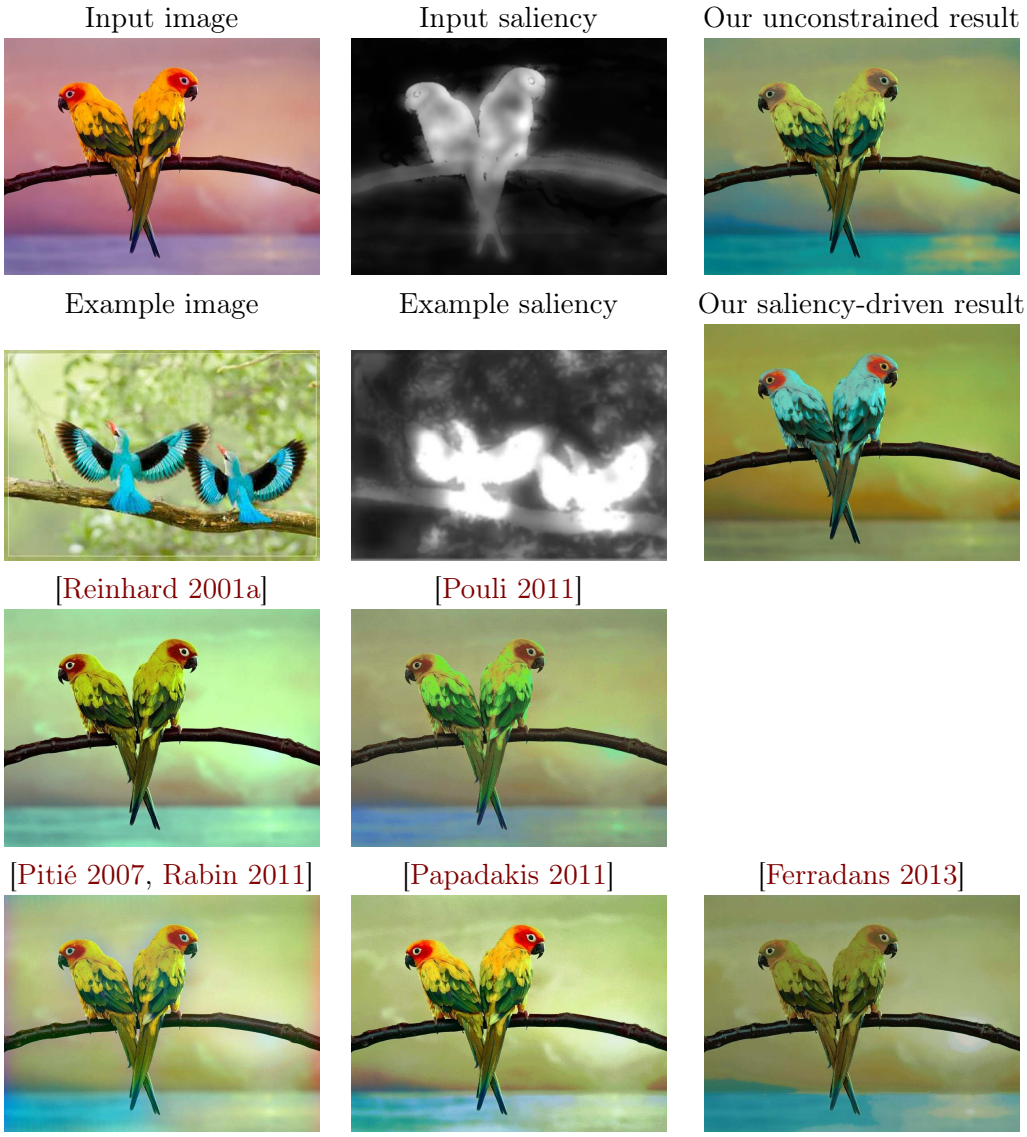


Figure A.8: Illustration of semantically consistent result by applying saliency constraint. It must be noted that this example is very difficult, and all global methods fail to transfer efficiently the birds colors. We show that saliency can be easily integrated as a constraint. In our saliency-driven color transfer, both the birds and the background have been assigned the expected colors of the example.

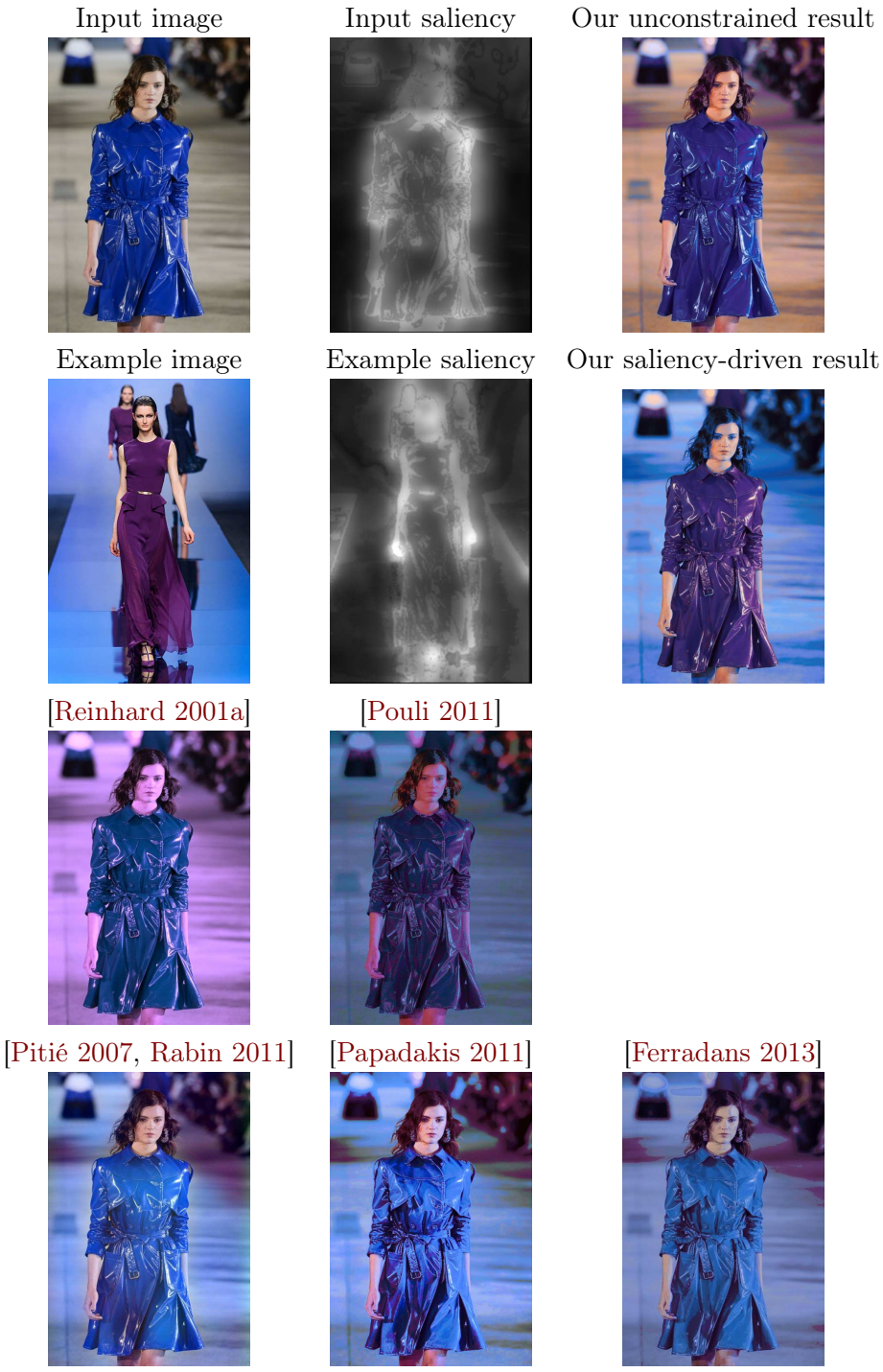


Figure A.9: Illustration of semantically consistent result by applying saliency constraint. As for the previous example, all global methods cannot recover the purple color of the dress. In our saliency-driven color transfer, both the dress and the background have been assigned the expected colors of the example.

A.4 Video color transfer

Note that for the presented video color transfer results, we have proceeded as follows: First, we extracted a specific frame from the source video, this frame was used as input image to compute the optimal transportation color transfer (as described in the paper). Then, the obtained Thin Plate Spline transformation was stored in a $3D$ LUT (Look-up Table). Finally, this smooth color transformation was applied to all frames of the input video.

Figure A.10 illustrates how the video color transfer was computed.

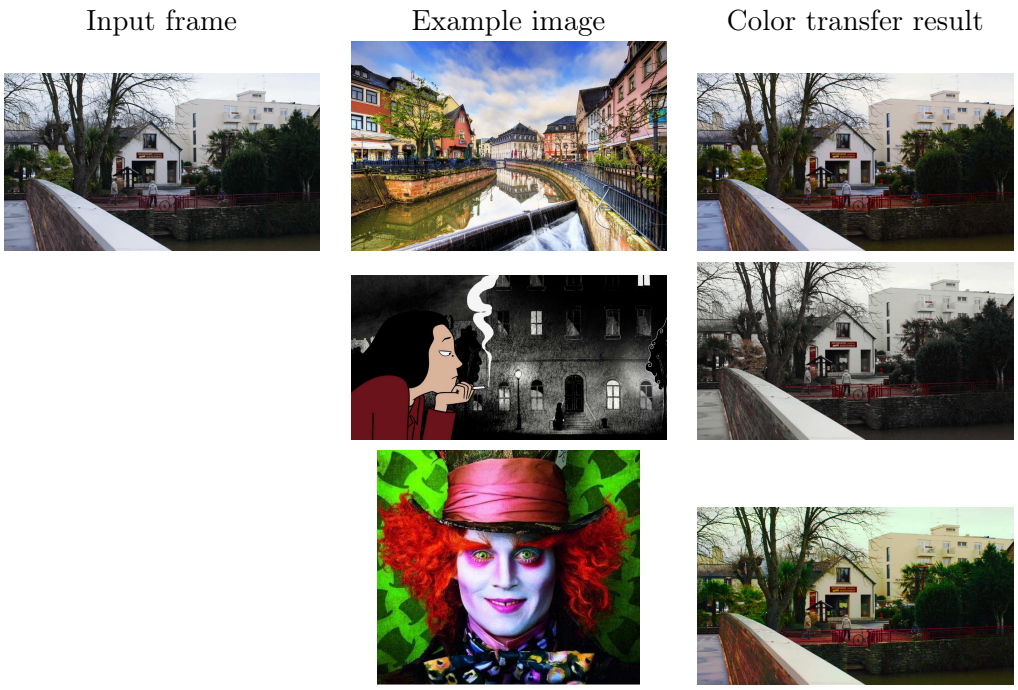


Figure A.10: Results used to compute a color transformation to perform video color transfer. In order to illustrate two different scenarios of video color transfer, the example image *Saarburg* (first row) was chosen to show a demonstration of color enhancement application, while the example images *Persepolis* (second row) and *Alice* (third row) were chosen to demonstrate an artistic color grading application.

Appendix B

Additional Tonal Stabilization Results

We present tonal stabilization results for sequences that were shot aiming to have an easily localized ground truth for the sequences “PhD Office” (Fig. B.1) and “Corridor” (Fig. B.2). In particular, a Macbeth colorchart was arranged in a fixed position with respect to the camera, so that no trajectory tracking is necessary for evaluation. Then, we use the mean of all colors to compute the chromatic differences. Note that the colorchart is not used to guide the estimation of tonal transformations.

Obtaining tonal stabilization is challenging for the “PhD Office” sequence (Fig. B.1). This sequence contains fast motion, which in turn compromises the accuracy of the correspondence set computed by [Farbman 2011]. In fact, we can observe that the method of [Farbman 2011] fails to handle tonal stabilization in this sequence, since the corrected sequence has higher temporal chromatic error than the original sequence. Note for example the color of the floor, which turns from blue to green in the original sequence, while in the stabilization performed by [Farbman 2011], the floor is mapped to an even less consistent color than the original sequence. On the other hand, we can observe that our method is able to efficiently preserve the initial blue color of the floor.

Similarly, the "Corridor" sequence (Fig. B.2) is challenging for having fast motion and also severe noise. We can observe that the method from [Farbman 2011] produces some flickering in the result (peaks in the plot) and overall higher chromatic difference than the original sequence.

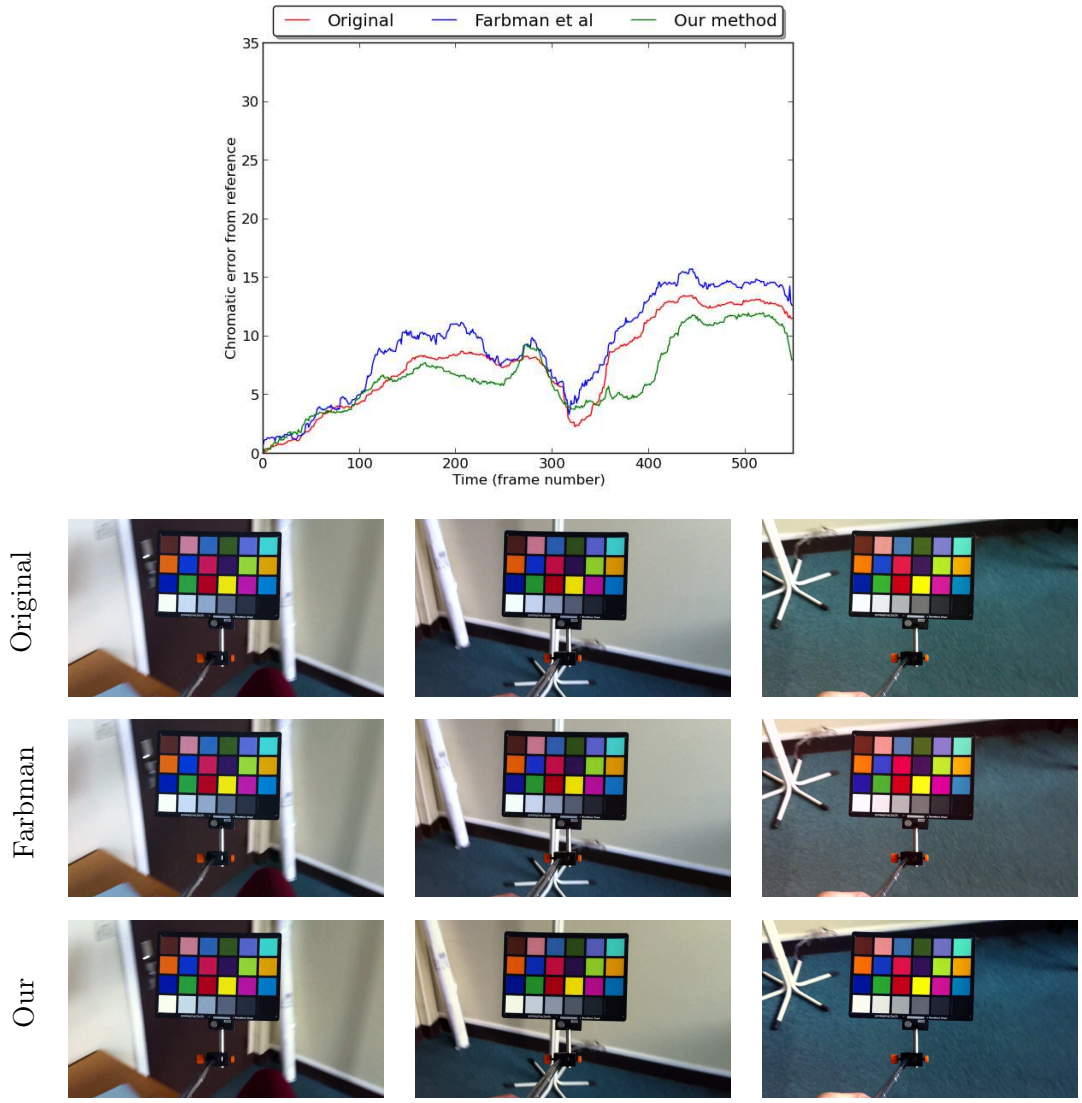


Figure B.1: Objective comparison of our method to Farbman *et al* [Farbman 2011] for sequence "PhD Office". We plot chromatic difference (chromatic intensities in CIELAB) from reference color chart colors. Note that our method reduces the tonal instability (clearly visible in preservation of blue color in the floor), while the method [Farbman 2011] fail to stabilize the sequence, producing higher instability than observed in the original sequence.

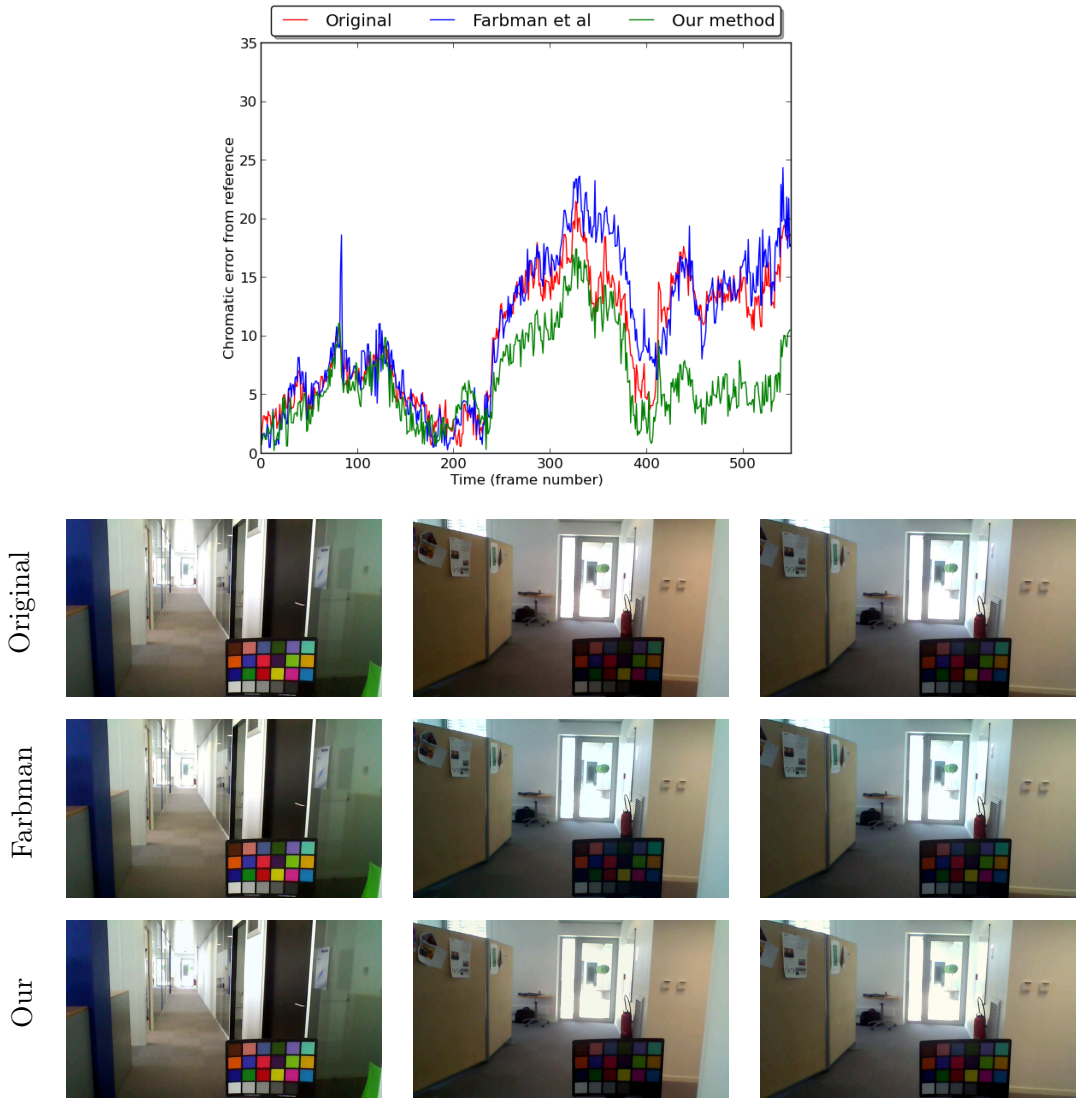


Figure B.2: Objective comparison of our method to Farbman *et al* [Farbman 2011] for sequence "Corridor". We plot chromatic difference (chromatic intensities in CIELAB) from reference color chart colors. Note that this sequence contains motion, noise and extreme changes in exposure. These changes in exposure implies the impossibility to preserve the chromaticities in color chart (color intensity in color chart is nearly lost in underexposured frames). Even though, our method is able to reduce the tonal instability, while [Farbman 2011] reduces very little the instbaility and produces some flicker (high peaks in the graph).

Appendix C

Additional Style Transfer Results

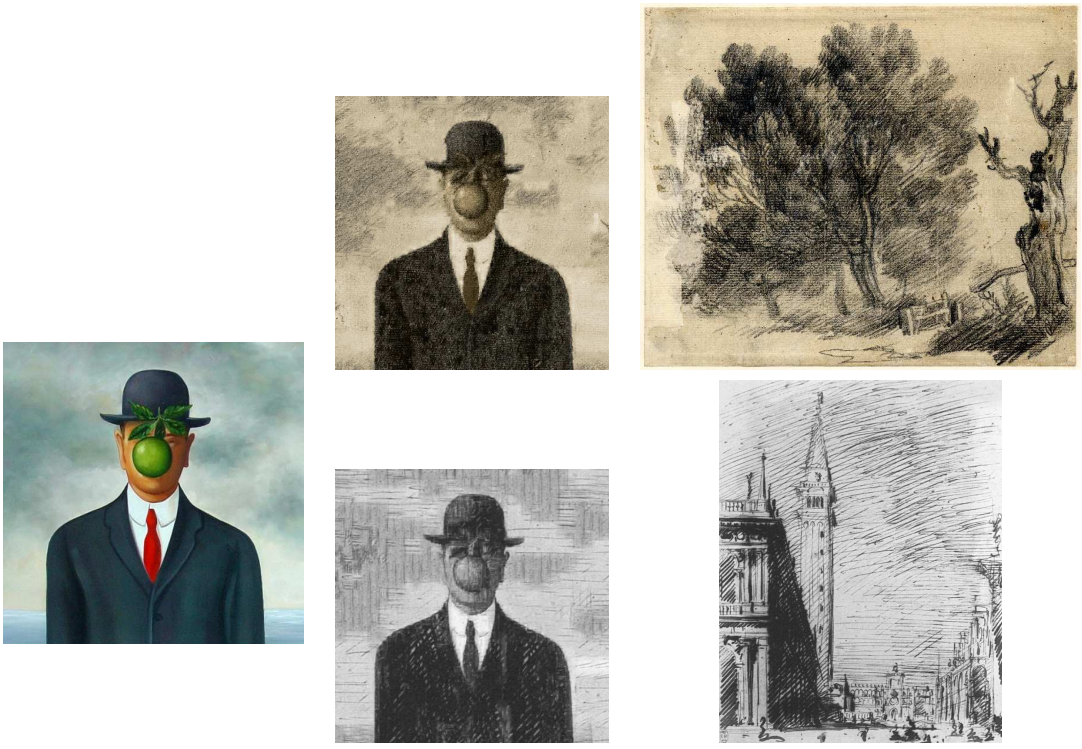


Figure C.1: Illustration of example-based style transfer for sketches. First row: original image from Magritte; Second and third row: result of our algorithm using as example the smaller sketches at the right. In this example texture as well as color are very important to reproduce the style.

Figure C.2: Results of our method with Van Gogh's paintings as examples. Top row: example images, Left column: original images.

Figure C.3: Results of our method with Monet's paintings as examples. Top row: example images, Left column: original images.

Figure C.4: Results of our method with Seurat's paintings as examples. Top row: example images, Left column: original images.

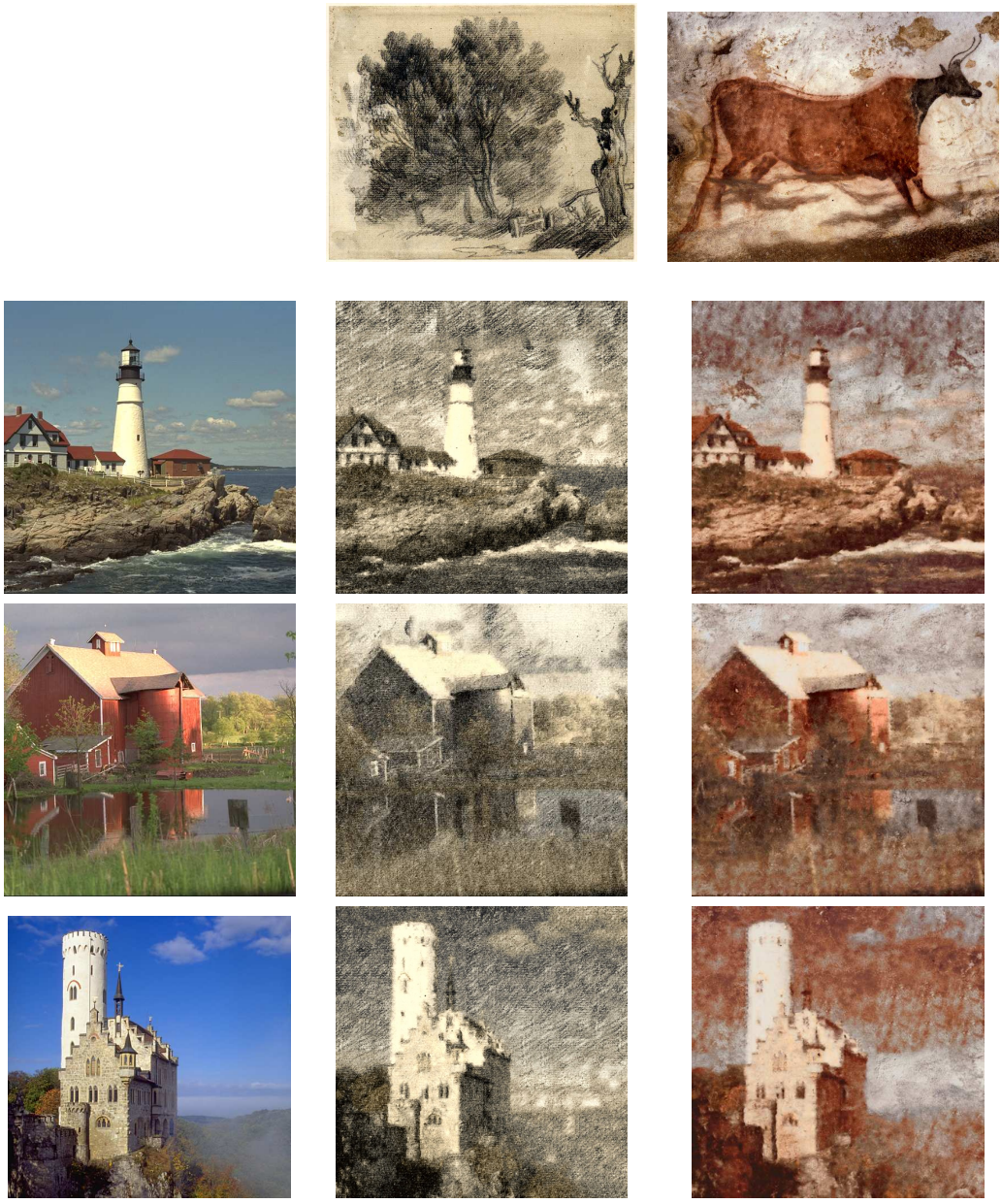
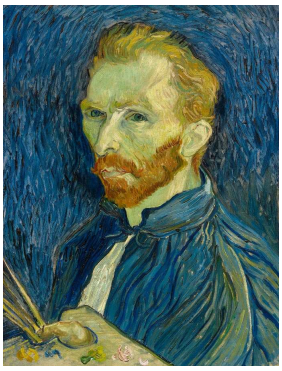


Figure C.5: Results of our method with Gainsborough's sketch and Lascaux cave drawing as examples. Top row: example images, Left column: original images.

Original



Van Gogh's example

Gainsborough's example

Style transfer - Van Gogh

Style transfer

Texture transfer - Van Gogh

Texture transfer

Color transfer - Van Gogh

Color transfer

Figure C.6: Illustration of color, texture and style transfer. Note that only color transfer (fourth row) and only texture transfer (third row) are not sufficient to capture the style of the example images. Our method combines texture and color transfer (second row) to better capture the style of the example image.

Bibliography

[Ashikhmin 2001] Michael Ashikhmin. *Synthesizing Natural Textures.* In Proceedings of the 2001 Symposium on Interactive 3D Graphics, I3D '01, pages 217–226, New York, NY, USA, 2001. (Cited on page 85.)

[Barnes 2010] Connelly Barnes, Eli Shechtman, DanB. Goldman and Adam Finkelstein. *The Generalized PatchMatch Correspondence Algorithm.* In Kostas Daniilidis, Petros Maragos and Nikos Paragios, editors, Computer Vision – ECCV 2010, volume 6313 of *Lecture Notes in Computer Science*, pages 29–43. Springer Berlin Heidelberg, 2010. (Cited on page 86.)

[Barnes 2015] Connelly Barnes, Fang-Lue Zhang, Liming Lou, Xian Wu and Shi-Min Hu. *PatchTable: Efficient Patch Queries for Large Datasets and Applications.* In SIGGRAPH, August 2015. (Cited on pages 86, 94 and 96.)

[Bénard 2013] Pierre Bénard, Forrester Cole, Michael Kass, Igor Mordatch, James Hegarty, Martin Sebastian Senn, Kurt Fleischer, Davide Pesare and Katherine Breeden. *Stylizing Animation by Example.* ACM TOG, vol. 32, no. 4, pages 119:1–119:12, July 2013. (Cited on page 86.)

[Black 1996] Michael J. Black and P. Anandan. *The Robust Estimation of Multiple Motions: Parametric and Piecewise-smooth Flow Fields.* Comput. Vis. Image Underst., vol. 63, no. 1, pages 75–104, January 1996. (Cited on page 54.)

[Blake 2011] Andrew Blake, Pushmeet Kohli and Carsten Rother. Markov random fields for vision and image processing. The MIT Press, 2011. (Cited on page 83.)

[Bonneel 2013] Nicolas Bonneel, Kalyan Sunkavalli, Sylvain Paris and Hanspeter Pfister. *Example-Based Video Color Grading.* ACM Transactions on Graphics (Proceedings of SIGGRAPH 2013), vol. 32, no. 4, 2013. (Cited on pages 27 and 114.)

[Bookstein 1989] F. L. Bookstein. *Principal warps: thin-plate splines and the decomposition of deformations.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 6, pages 567–585, Jun 1989. (Cited on pages 30 and 31.)

[Brainard 1986] D. H. Brainard and B. A. Wandell. *Analysis of the retinex theory of color vision.* JOSA A, vol. 3, no. 10, pages 1651 – 1661, 1986. (Cited on page 12.)

[Buchsbaum 1980] G. Buchsbaum. *A Spatial Processor Model for Object Colour Perception.* Journal of the Franklin Institute, vol. 310, no. 1, pages 1–26, 1980. (Cited on page 14.)

[Butler 2012] D. J. Butler, J. Wulff, G. B. Stanley and M. J. Black. *A naturalistic open source movie for optical flow evaluation.* In A. Fitzgibbon et al. (Eds.), editor, European Conf. on Computer Vision (ECCV), Part IV, LNCS 7577, pages 611–625. Springer-Verlag, October 2012. (Cited on page 106.)

[Chakrabarti 2009] Ayan Chakrabarti, Daniel Scharstein and Todd Zickler. *An Empirical Camera Model for Internet Color Vision.* Procedings of the British Machine Vision Conference 2009, pages 51.1–51.11, 2009. (Cited on page 22.)

[Cheng 2008] Li Cheng, S.V.N. Vishwanathan and Xinhua Zhang. *Consistent image analogies using semi-supervised learning.* In CVPR, 2008. (Cited on page 86.)

[Chiou 2010] Wan-Chien Chiou, Yi-Lei Chen and Chiou-Ting Hsu. *Color transfer for complex content images based on intrinsic component.* In Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on, pages 156–161, 2010. (Cited on page 35.)

[Colom 2013] Miguel Colom and Antoni Buades. *Analysis and Extension of the Percentile Method, Estimating a Noise Curve from a Single Image.* Image Processing On Line, vol. 3, pages 332–359, 2013. (Cited on page 57.)

[Criminisi 2004] Antonio Criminisi, P. Perez and K. Toyama. *Region filling and object removal by exemplar-based image inpainting.* IEEE T-IP, vol. 13, no. 9, pages 1200–1212, Sept 2004. (Cited on page 82.)

[Cusano 2012] Claudio Cusano, Francesca Gasparini and Raimondo Schettini. *Color transfer using semantic image annotation.* pages 82990U–82990U–8, 2012. (Cited on page 27.)

[Decencière 1997] Étienne Decencière. *Restauration automatique de films anciens.* Theses, École Nationale Supérieure des Mines de Paris, December 1997. (Cited on page 44.)

[Delon 2004] Julie Delon. *Midway image equalization.* Journal of Mathematical Imaging and Vision, vol. 21, pages 119–134, 2004. (Cited on page 45.)

[Delon 2006] J. Delon. *Movie and video scale-time equalization application to flicker reduction.* Image Processing, IEEE Transactions on, vol. 15, no. 1, pages 241–248, Jan 2006. (Cited on pages 45 and 57.)

[Delon 2007] J. Delon, A. Desolneux, J.-L. Lisani and A.B. Petro. *A Nonparametric Approach for Histogram Segmentation.* Image Processing, IEEE Transactions on, vol. 16, no. 1, pages 253 –261, jan. 2007. (Cited on page 30.)

[Delon 2010] J. Delon and A. Desolneux. *Stabilization of Flicker-Like Effects in Image Sequences through Local Contrast Correction.* SIAM Journal on Imaging Sciences, vol. 3, no. 4, pages 703–734, 2010. (Cited on page 45.)

[Durand 2002] Frédo Durand. *An Invitation to Discuss Computer Depiction.* In NPAR, pages 111–124, New York, NY, USA, 2002. (Cited on page 81.)

[Efros 1999] Alexei A. Efros and Thomas K. Leung. *Texture Synthesis by Non-Parametric Sampling.* In ICCV, pages 1033–, Washington, DC, USA, 1999. (Cited on pages 84 and 85.)

[Efros 2001] Alexei A. Efros and William T. Freeman. *Image Quilting for Texture Synthesis and Transfer.* In SIGGRAPH, pages 341–346, New York, NY, USA, 2001. (Cited on pages 82, 84 and 85.)

[en Guo 2003] Cheng en Guo, Song Chun Zhu and Ying Nian Wu. *Towards a mathematical theory of primal sketch and sketchability.* In ICCV, 2003. (Cited on page 87.)

[Fairchild 2005] Mark D. Fairchild. Color appearance models, second edition. Wiley-IS and T Series in Imaging Science and Technology, Chichester, UK, 2005. (Cited on page 19.)

[Farbman 2011] Zeev Farbman and Dani Lischinski. *Tonal Stabilization of Video.* ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2011), vol. 30, no. 4, pages 89:1 – 89:9, 2011. (Cited on pages 42, 45, 47, 48, 51, 58, 63, 67, 69, 72, 73, 76, 77, 78, 131, 132 and 133.)

[Faridul 2013] H. S. Faridul, J. Stauder, J. Kervec and A. Trémeau. *Approximate Cross Channel Color Mapping from Sparse Color Correspondences.* In Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on, pages 860–867, Dec 2013. (Cited on page 27.)

[Ferradans 2013] Sira Ferradans, Nicolas Papadakis, Julien Rabin, Gabriel Peyré and Jean-François Aujol. *Regularized Discrete Optimal Transport.* In Arjan Kuijper, Kristian Bredies, Thomas Pock and Horst Bischof, editors, Scale Space and Variational Methods in Computer Vision, volume 7893 of *Lecture Notes in Computer Science*, pages 428–439. Springer Berlin Heidelberg, 2013. (Cited on pages 27, 33, 34, 35, 38, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127 and 128.)

[Finlayson 2000] Graham D. Finlayson and Sabine Süsstrunk. *Spectral Sharpening and the Bradford Transform.* In Proc. CIS2000, pages 236–243, 2000. (Cited on page 19.)

[Finlayson 2005] Graham D. Finlayson and Elisabetta Trezzi. *Shades of Gray and Colour Constancy.* In Color Imaging Conference, pages 37–41. IS&T - The Society for Imaging Science and Technology, 2005. (Cited on page 15.)

[Fišer 2016] Jakub Fišer, Ondřej Jamriška, Michal Lukáč, Eli Shechtman, Paul Asente, Jingwan Lu and Daniel Sýkora. *StyLit: Illumination-Guided*

Example-Based Stylization of 3D Renderings. ACM Transactions on Graphics, vol. 35, no. 4, 2016. (Cited on page 87.)

[Freedman 2010] D. Freedman and P. Kisilev. *Object-to-object color transfer: Optimal flows and SMSP transformations*. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 287–294, 2010. (Cited on page 27.)

[Freeman 2000] WilliamT. Freeman, EgonC. Pasztor and OwenT. Carmichael. *Learning Low-Level Vision*. IJCV, vol. 40, no. 1, pages 25–47, 2000. (Cited on pages 83, 84, 85 and 91.)

[Freeman 2002] W.T. Freeman, T.R. Jones and E.C. Pasztor. *Example-based super-resolution*. IEEE Comput. Graph. Appl., vol. 22, no. 2, pages 56–65, Mar 2002. (Cited on page 82.)

[Freeman 2010] Michael Freeman. *The DSLR Field Guide: The Essential Guide to Getting the Most from Your Camera*. 2010. (Cited on page 16.)

[Frigo 2014] Oriel Frigo, Neus Sabater, Vincent Demoulin and Pierre Hellier. *Optimal Transportation for Example-Guided Color Transfer*. In ACCV, pages 655–670, 2014. (Cited on pages 5, 82, 89 and 94.)

[Frigo 2015a] O. Frigo, N. Sabater, J. Delon and P. Hellier. *Motion driven tonal stabilization*. In Image Processing (ICIP), 2015 IEEE International Conference on, pages 3372–3376, Sept 2015. (Cited on page 6.)

[Frigo 2015b] O. Frigo, N. Sabater, J. Delon and P. Hellier. *Stabilisation tonale de videos*. In Colloque GRETSI, Sept 2015. (Cited on page 6.)

[Frigo 2016] O. Frigo, N. Sabater, J. Delon and P. Hellier. *Split and Match: Example-based Adaptive Patch Sampling for Unsupervised Style Transfer*. In Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on, 2016. (Cited on page 7.)

[Gatys 2015] Leon A. Gatys, Alexander S. Ecker and Matthias Bethge. *A Neural Algorithm of Artistic Style*. CoRR, vol. abs/1508.06576, 2015. (Cited on pages 86, 95, 96 and 115.)

[Gatys 2016] Leon A. Gatys, Alexander S. Ecker and Matthias Bethge. *Image Style Transfer Using Convolutional Neural Networks*. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016. (Cited on page 115.)

[Geman 1984] Stuart Geman and Donald Geman. *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images*. IEEE T-PAMI, vol. 6, no. 6, pages 721–741, November 1984. (Cited on page 83.)

[Gijsenij 2011] A. Gijsenij, T. Gevers and J. Van De Weijer. *Computational color constancy: Survey and experiments.* Image Processing, IEEE Transactions on, vol. 20, no. 9, pages 2475 – 2489, 2011. (Cited on pages 12 and 14.)

[Gijsenij 2012] A. Gijsenij, T. Gevers and J. van de Weijer. *Improving color constancy by photometric edge weighting.* IEEE Trans Pattern Anal Mach Intell, vol. 34, no. 5, pages 918 – 929, 2012. (Cited on page 15.)

[Gonzalez 2007] Rafael C Gonzalez and Richard E Woods. Digital Image Processing (3rd Edition). Prentice Hall, 3 édition, 2007. (Cited on page 11.)

[Grossberg 2002] M Grossberg and S Nayar. *What Can Be Known about the Radiometric Response from Images?* In Computer Vision - ECCV 2002, pages 189–205. 2002. (Cited on page 22.)

[Grossberg 2003] Michael D. Grossberg and Shree K. Nayar. *Determining the camera response from images: What is knowable?* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, pages 1455–1467, 2003. (Cited on page 23.)

[HaCohen 2011] Yoav HaCohen, Eli Shechtman, Dan B. Goldman and Dani Lischinski. *Non-rigid dense correspondence with applications for image enhancement.* ACM Trans. Graph., vol. 30, no. 4, pages 70:1–70:10, July 2011. (Cited on pages 27, 35, 36, 43, 121, 122, 123, 124 and 125.)

[HaCohen 2013] Yoav HaCohen, Eli Shechtman, Dan B. Goldman and Dani Lischinski. *Optimizing Color Consistency in Photo Collections.* ACM Trans. Graph., vol. 32, no. 4, pages 38:1–38:10, July 2013. (Cited on page 43.)

[Hertzmann 2001] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless and David H. Salesin. *Image Analogies.* In SIGGRAPH, pages 327–340, New York, NY, USA, 2001. (Cited on pages 84, 85, 86, 94, 101 and 103.)

[Hirai 2010a] K. Hirai, N. Tsumura, T. Nakaguchi, Y. Miyake and S. Tominaga. *Spatio-velocity contrast sensitivity functions and video quality assessment.* In Intelligent Signal Processing and Communication Systems (ISPACS), 2010 International Symposium on, pages 1–4, Dec 2010. (Cited on page 60.)

[Hirai 2010b] Keita Hirai, Toshiaki Mikami, Norimichi Tsumura and Toshiya Nakaguchi. *Measurement and Modeling of Chromatic Spatio-Velocity Contrast Sensitivity Function and its Application to Video Quality Evaluation.* Color and Imaging Conference, vol. 2010, no. 1, pages 86–91, 2010. (Cited on page 60.)

[Hordley 2006] S. D. Hordley. *Scene illuminant estimation: Past, present, and future.* Color Research & Application, vol. 31, no. 4, pages 303 – 314, 2006. (Cited on page 12.)

[Horowitz 1974] S.L. Horowitz and T. Pavlidis. *Picture Segmentation by a Directed Split and Merge Procedure.* In ICPR, pages 424–433, 1974. (Cited on page 89.)

[Huo 2006] Jun-yan Huo, Yi-lin Chang, Jing Wang and Xiao-xia Wei. *Robust automatic white balance algorithm using gray color points in images.* IEEE Trans. on Consum. Electron., vol. 52, no. 2, pages 541–546, September 2006. (Cited on pages 15 and 28.)

[Hwang 2014] Youngbae Hwang, Joon-Young Lee, In So Kweon and Seon Joo Kim. *Color Transfer Using Probabilistic Moving Least Squares.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014. (Cited on pages 27 and 35.)

[Kim 2012] Seon Joo Kim, Hai Ting Lin, Zheng Lu, Sabine Süsstrunk, Stephen Lin and Michael S. Brown. *A new in-camera imaging model for color computer vision and its application.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, pages 2289–2302, 2012. (Cited on pages 22, 23, 48, 49, 50, 51, 54 and 75.)

[Kyprianidis 2013] J.E. Kyprianidis, J. Collomosse, Tinghuai Wang and T. Isenberg. *State of the Art: A Taxonomy of Artistic Stylization Techniques for Images and Video.* IEEE TVCG, vol. 19, no. 5, pages 866–885, May 2013. (Cited on page 81.)

[Land 1971] E. H. Land, J. J. McCann*et al. Lightness and retinex theory.* Journal of the Optical society of America, vol. 61, no. 1, pages 1 – 11, 1971. (Cited on page 12.)

[Land 1977] Edwin H Land. *The Retinex Theory of Color Vision The Retinex Theory of Color Vision.* vol. 237, no. 6, 1977. (Cited on page 12.)

[Le Meur 2006] O. Le Meur, P. Le Callet, D. Barba and D. Thoreau. *A coherent computational approach to model bottom-up visual attention.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 28, no. 5, pages 802–817, 2006. (Cited on page 32.)

[Levenberg 1944] K. Levenberg. *A method for the solution of certain non-linear problems in least squares.* Quart. J. Appl. Maths., vol. II, no. 2, pages 164–168, 1944. (Cited on page 53.)

[Li 2016] Chuan Li and Michael Wand. *Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis.* In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016. (Cited on page 116.)

[Liang 2001] Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo and Heung-Yeung Shum. *Real-time Texture Synthesis by Patch-based Sampling*. ACM Trans. Graph., vol. 20, no. 3, pages 127–150, July 2001. (Cited on page 85.)

[Lin 2011] Haiting Lin, Seon Joo Kim, Sabine Susstrunk and Michael S. Brown. *Revisiting radiometric calibration for color computer vision*. In Proceedings of the IEEE International Conference on Computer Vision, pages 129–136, 2011. (Cited on pages 22 and 23.)

[Lindner 2012] Albrecht Lindner, Appu Shaji, Nicolas Bonnier and Sabine Süsstrunk. *Joint Statistical Analysis of Images and Keywords with Applications in Semantic Image Enhancement*. In Proceedings of the 20th ACM International Conference on Multimedia, MM '12, pages 489–498, New York, NY, USA, 2012. ACM. (Cited on page 27.)

[Liu 2008] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic and William T. Freeman. *SIFT Flow: Dense Correspondence across Different Scenes*. In Proceedings of the 10th European Conference on Computer Vision: Part III, ECCV '08, pages 28–42, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on page 55.)

[Luo 1998] M. R. Luo and R. W. G. Hunt. *The structure of the CIE 1997 Colour Appearance Model (CIECAM97s)*. Color Research and Application, vol. 23, no. 3, pages 138–146, 1998. (Cited on page 19.)

[Mann 1995] S Mann and R W Picard. *On Being 'undigital' With Digital Cameras: Extending Dynamic Range By Combining Differently Exposed Pictures*. Proceedings of IS&T, pages 442–448, 1995. (Cited on page 23.)

[Marquardt 1963] Donald W. Marquardt. *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*. SIAM Journal on Applied Mathematics, vol. 11, no. 2, pages 431–441, 1963. (Cited on page 53.)

[Marr 1982] David Marr. Vision: A computational investigation into the human representation and processing of visual information. Henry Holt and Co., Inc., New York, NY, USA, 1982. (Cited on page 87.)

[Mazin 2015] Baptiste Mazin, Julie Delon and Yann Gousseau. *Estimation of Illuminants From Projections on the Planckian Locus*. IEEE Trans. Image Processing, vol. 24, no. 6, pages 1944–1955, 2015. (Cited on page 12.)

[McCann 2005] John J McCann. *Do Humans Discount the Illuminant?* vol. 5666, pages 9–16, 2005. (Cited on page 13.)

[Mitsunaga 1999] T. Mitsunaga and S.K. Nayar. *Radiometric self calibration*. Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), vol. 1, 1999. (Cited on pages 22 and 23.)

[Moroney 2002] Nathan Moroney, Mark D. Fairchild, Robert W. G. Hunt, Changjun Li, M. Ronnier Luo and Todd Newman. *The CIECAM02 Color Appearance Model*. In Color Imaging Conference, pages 23–27, 2002. (Cited on pages 20 and 29.)

[Murray 2011] Naila Murray, Sandra Skaff, Luca Marchesotti and Florent Perronnin. *Towards automatic concept transfer*. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering, NPAR '11, pages 167–176, New York, NY, USA, 2011. ACM. (Cited on page 27.)

[Naranjo 2000] V. Naranjo and A. Albiol. *Flicker reduction in old films*. In Image Processing, 2000. Proceedings. 2000 International Conference on, volume 2, pages 657–659 vol.2, Sept 2000. (Cited on page 44.)

[Neumann 2005] Laszlo Neumann and Attila Neumann. *Color Style Transfer Techniques using Hue, Lightness and Saturation Histogram Matching*. In B. Gooch W. Purgathofer L. Neumann M. Sbert, editor, Computational Aesthetics in Graphics, Visualization and Imaging 2005, pages 111–122, 5 2005. (Cited on page 26.)

[Odobez 1995] J. M. Odobez and P. Bouthemy. *Robust multiresolution estimation of parametric motion models*. Jal of Vis. Comm. and Image Representation, 1995. (Cited on pages 54, 55 and 56.)

[Okura 2015] Fumio Okura, Kenneth Vanhoey, Adrien Bousseau, Alexei A Efros and George Drettakis. *Unifying Color and Texture Transfer for Predictive Appearance Manipulation*. In Computer Graphics Forum, volume 34, pages 53–63, 2015. (Cited on page 86.)

[Papadakis 2011] N. Papadakis, E. Provenzi and V. Caselles. *A Variational Model for Histogram Transfer of Color Images*. Image Processing, IEEE Transactions on, vol. 20, no. 6, pages 1682–1695, 2011. (Cited on pages 26, 33, 34, 35, 38, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127 and 128.)

[Pearl 1988] Judea Pearl. Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. (Cited on page 93.)

[Pitie 2004] François Pitie, Rozenn Dahyot, Francis Kelly and Anil Kokaram. *A New Robust Technique for Stabilizing Brightness Fluctuations in Image Sequences*. In Dorin Comaniciu, Rudolf Mester, Kenichi Kanatani and David Suter, editors, Statistical Methods in Video Processing, volume 3247 of *Lecture Notes in Computer Science*, pages 153–164. Springer Berlin Heidelberg, 2004. (Cited on page 45.)

[Pitié 2007] François Pitié, Anil C. Kokaram and Rozenn Dahyot. *Automated colour grading using colour distribution transfer*. Comput. Vis. Image Underst.,

vol. 107, no. 1-2, pages 123–137, July 2007. (Cited on pages 26, 33, 34, 38, 71, 74, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127 and 128.)

[Pouli 2011] Tania Pouli and Erik Reinhard. *Progressive color transfer for images of arbitrary dynamic range*. Computers and Graphics, vol. 35, no. 1, pages 67 – 80, 2011. Extended Papers from Non-Photorealistic Animation and Rendering (NPAR) 2010. (Cited on pages 26, 33, 34, 35, 38, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127 and 128.)

[Provenzi 2007] Edoardo Provenzi, Massimo Fierro, Alessandro Rizzi, L. De Carli, Davide Gadia and Daniele Marini. *Random Spray Retinex: A New Retinex Implementation to Investigate the Local Properties of the Model*. IEEE Trans. Image Processing, vol. 16, no. 1, pages 162–171, 2007. (Cited on page 12.)

[Rabin 2010] J. Rabin, J. Delon and Y. Gousseau. *Regularization of transportation maps for color and contrast transfer*. In Image Processing (ICIP), 2010 17th IEEE International Conference on, pages 1933–1936, 2010. (Cited on page 51.)

[Rabin 2011] Julien Rabin, Julie Delon and Yann Gousseau. *Removing artefacts from color and contrast modifications*. IEEE Transactions on Image Processing, vol. 20, no. 11, pages 3073–3085, 2011. (Cited on pages 26, 33, 34, 35, 38, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127 and 128.)

[Reinhard 2001a] E. Reinhard, M. Adhikhmin, B. Gooch and P. Shirley. *Color transfer between images*. Computer Graphics and Applications, IEEE, vol. 21, no. 5, pages 34 –41, sep/oct 2001. (Cited on pages 26, 27, 28, 33, 34, 35, 82, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127 and 128.)

[Reinhard 2001b] Erik Reinhard. *Statistical Approaches to Image and Scene Manipulation*, 2001. (Cited on page 20.)

[Rosales 2003] Romer Rosales, Kannan Achan and Brendan J. Frey. *Unsupervised Image Translation*. In ICCV, pages 472–478, 2003. (Cited on page 86.)

[Rubner 2000] Yossi Rubner, Carlo Tomasi and Leonidas J. Guibas. *The Earth Mover's Distance as a Metric for Image Retrieval*. International Journal of Computer Vision, vol. 40, no. 2, pages 99–121, 2000. (Cited on pages 30 and 31.)

[Ruder 2016] Manuel Ruder, Alexey Dosovitskiy and Thomas Brox. *Artistic style transfer for videos*. CoRR, vol. abs/1604.08610, 2016. (Cited on page 87.)

[Ruderman 1998] Daniel L. Ruderman, Thomas W. Cronin and Chuan-Chin Chiao. *Statistics of cone responses to natural images: implications for visual coding*. J. Opt. Soc. Am. A, vol. 15, no. 8, pages 2036–2045, Aug 1998. (Cited on page 20.)

[Russell 2011] Bryan C. Russell, Josef Sivic, Jean Ponce and Helene Dessales. *Automatic alignment of paintings and photographs depicting a 3D scene.* In IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011, pages 545–552, 2011. (Cited on page 88.)

[Shannon 1948] C. E. Shannon. *A Mathematical Theory of Communication.* Bell System Technical Journal, vol. 27, no. 3, pages 379–423, 1948. (Cited on page 84.)

[Shi 1993] Jianbo Shi and Carlo Tomasi. *Good Features to Track.* Technical report, Ithaca, NY, USA, 1993. (Cited on page 54.)

[Shih 2013] Yichang Shih, Sylvain Paris, Frédo Durand and William T Freeman. *Data-driven hallucination of different times of day from a single outdoor photo.* ACM TOG, vol. 32, no. 6, page 200, 2013. (Cited on page 83.)

[Shih 2014] YiChang Shih, Sylvain Paris, Connelly Barnes, William T Freeman and Frédo Durand. *Style transfer for headshot portraits.* ACM TOG, vol. 33, no. 4, page 148, 2014. (Cited on page 83.)

[Sundaram 2010] N. Sundaram, T. Brox and K. Keutzer. *Dense point trajectories by GPU-accelerated large displacement optical flow.* In European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science. Springer, Sept. 2010. (Cited on page 102.)

[Sustrunk 2005] Sabine Sustrunk and Graham D. Finlayson. *Evaluating Chromatic Adaptation Transform Performance.* In Proc. IS&T/SID 13th Color Imaging Conference, pages 75–78, 2005. (Cited on page 13.)

[Tai 2005] Yuwing Tai, Jiaya Jia and Chi keung Tang. *Local Color Transfer via Probabilistic Segmentation by Expectation-Maximization.* In Proc. Computer Vision and Pattern Recognition, pages 747–754, 2005. (Cited on page 26.)

[van Roosmalen 1999] P.M.B. van Roosmalen, R.L. Lagendijk and J. Biemond. *Correction of intensity flicker in old film sequences.* IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 7, pages 1013–1019, 1999. (Cited on pages 44 and 59.)

[Vazquez-Corral 2014] J. Vazquez-Corral and M. Bertalmio. *Color Stabilization Along Time and Across Shots of the Same Scene, for One or Several Cameras of Unknown Specifications.* Image Processing, IEEE Transactions on, vol. 23, no. 10, pages 4564–4575, Oct 2014. (Cited on pages 43, 53, 54, 63, 68, 69, 71, 74, 75 and 76.)

[Viola 2004] Paul Viola and Michael J Jones. *Robust real-time face detection.* International journal of computer vision, vol. 57, no. 2, pages 137–154, 2004. (Cited on page 32.)

[Wandell 1995] B. A. Wandell. Foundations of vision. Sinauer Associates, Inc., 1995. (Cited on pages 12, 17 and 18.)

[Wang 2004] Zhou Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli. *Image quality assessment: from error visibility to structural similarity.* IEEE Transactions on Image Processing, vol. 13, no. 4, pages 600–612, April 2004. (Cited on pages 35, 38, 117, 121 and 126.)

[Wang 2009] Xiaogang Wang and Xiaoou Tang. *Face Photo-Sketch Synthesis and Recognition.* IEEE T-PAMI, vol. 31, no. 11, pages 1955–1967, 2009. (Cited on page 85.)

[Wang 2014] Y. Wang, D. Tao, X. Li, M. Song, J. Bu and P. Tan. *Video Tonal Stabilization via Color States Smoothing.* IEEE transactions on image processing, vol. 23, no. 11, pages 4838–4849, 2014. (Cited on pages 42, 46, 69, 72, 73, 76, 77 and 78.)

[Weijer 2007] J. Van De Weijer, T. Gevers and A. Gijsenij. *Edge-based color constancy.* Image Processing, IEEE Transactions on, vol. 16, no. 9, pages 2207 – 2214, 2007. (Cited on page 15.)

[Weinzaepfel 2013] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui and Cordelia Schmid. *DeepFlow: Large displacement optical flow with deep matching.* In IEEE Intenational Conference on Computer Vision (ICCV), Sydney, Australia, December 2013. (Cited on pages 102 and 107.)

[Weiss 1997] Yair Weiss. *Belief Propagation and Revision in Networks with Loops.* Technical report, Cambridge, MA, USA, 1997. (Cited on pages 84, 85 and 93.)

[Weiss 2007] Yair Weiss and William T. Freeman. *What makes a good model of natural images.* In in: CVPR 2007: Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, pages 1–8, 2007. (Cited on page 83.)

[Wu 2013] Fuzhang Wu, Weiming Dong, Yan Kong, Xing Mei, Jean-Claude Paul and Xiaopeng Zhang. *Content-Based Colour Transfer.* Computer Graphics Forum, pages no–no, 2013. (Cited on page 27.)

[Wyszecki 2000] Günther Wyszecki and W S Stiles. Color Science: Concepts and Methods, Quantitative Data and Formulae (Wiley Series in Pure and Applied Optics). Wiley-Interscience, 2 édition, 2000. (Cited on pages 13 and 16.)

[Xiong 2012] Ying Xiong, Kate Saenko, Trevor Darrell and Todd Zickler. *From pixels to physics: Probabilistic color de-rendering.* In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 358–365, 2012. (Cited on pages 23 and 50.)

[Zhang 2013] Wei Zhang, Chen Cao, Shifeng Chen, Jianzhuang Liu and Xiaoou Tang. *Style Transfer Via Image Component Analysis.* IEEE Transactions on Multimedia, vol. 15, no. 7, pages 1594–1601, 2013. (Cited on pages 84 and 86.)

Example-based Video Editing

Abstract: The objective of this thesis is to provide new techniques for example-based video editing. We address three related problems: color transfer, tonal stabilization and color transfer. The first problem consists in transferring colors from an example image. We provide an efficient solution that maps color distributions by optimal transportation and performs a smooth color transformation that creates no artifacts.

The second problem is related to tonal fluctuation in videos. Due to the automatic settings of consumer cameras, the colors of objects in image sequences might change over time. We present a fast and computationally light method to stabilize video tonal appearance using a minimally-viable color correction model.

Finally, we discuss the problem of transferring the style of an example image to a source image. The complex notion of image style is considered as a local texture transfer, eventually coupled with a global color transfer. For the local texture transfer, we propose a new patch-based method using an adaptive partition that captures the style of the example image and preserves the structure of the source image. Results on various images show that the proposed technique is competitive with the most recent style transfer algorithms. **Keywords:** Video Editing, Color Image Processing, Color Transfer, White balance, Texture Synthesis