Statistical Signal Processing Techniques for Visual Post-Production

A dissertation submitted to the University of Dublin
for the Degree of Doctor of Philosophy

François Pitié
Trinity College Dublin, May 2006

Signal Processing and Media Applications
Department of Electronic and Electrical Engineering
Trinity College Dublin

Abstract

Digital media post-production is an industry standard step in media creation. Now that issues of speed and physical storage have largely been rendered less problematic, the emphasis has shifted toward increasing levels of automation. This thesis makes several contributions in the domain of visual post-production in an attempt to bring advanced statistical digital signal processing techniques to bear on several key problem.

The problem of transfer of statistics is first addressed. It transpires that this fundamental process underlies several key problems in film grading. This can be understood by considering an image as of features samples. Then texture or colour aspects can be described by the statistics of texture or colour samples. The first and major problem is that of colour grading. In that process the colour of each film frame has to be adjusted in order to match lighting and *atmosphere* throughout the shot. Consider a scene shot at midday and the same scene shot later in the afternoon, a typical procedure in movie production. The film shot during the midday shoot will have a totally different look and feel than the afternoon shoot. In addition, brightness can change from frame to frame due to film degradation, editing of different physical film material and even fluctuating lighting (*e.g.* fluorescent). In all the cases the solution is to design an automated process that alters the distribution of colour and brightness of each frame in such a way that it balances across all shots. The thesis proposes a new mechanism for doing this based on the iterative matching of projections of the colour distribution on different set of colour axes. In addition, algorithms for robustness to grain artefacts artefacts caused by stretched histogram matching, and occlusion problems are also proposed.

The second part of the thesis deals with tracking of objects and contours. Both methods proposed are founded on Bayesian grounds but adopt different optimisation strategies. Contour tracking is achieved by Bayesian Filtering exploiting a new prior that extracts local direction information from the image. Object tracking is achieved by a Viterbi algorithm that incorporates candidates for object position available in each frame. Both these tracking tools have implications for post-production. Contour tracking for rotoscoping is well established, but up till now incorporating edge information has been difficult. MCMC Bayesian filters have had great success for object tracking in recent times, but this thesis shows that given low level content knowledge it is possible to exploit deterministic strategies instead of MCMC and yield the same or better performance.

The thesis ends with work that proposes a new method for simplifying images implicitly. The idea is to create representations that are transmitted or compressed easily without exploiting heavy content information. A bit like in cartoonisation, images are re-expressed in a way which is not realistic but which conveys the same content while throwing away confusing details. Thus a textured object could be replaced by a simplified object in which the texture is removed for instance. The tools discussed previously could all be used in

tandem to create such an effect. This section of the work considers instead a purely implicit way to find similar patches in images and replace them by a unique patch representation. The work draws from the success of recent texture synthesis results that exploit non-parametric image models. The result is an image which has a better viewability and can be better compressed.

Declaration

I hereby declare that this thesis has not been submitted as an exercise for a degree at this or any other University and that it is entirely my own work.

I agree that the Library may lend or copy this thesis upon request.

Signed,

François Pitié

May 5, 2006.

Acknowledgments

If you are reading this page, you are probably looking for your name, and you most likely deserve it, so thank you very much! Else I believe you are suffering from compulsive insanity.

I would like to thank all former and present members of the Sigmedia group who have helped me over the years, and especially my PhD companions: the exhilarating Hugh Denman—Anil bless his thesis one day—for his huge and generous help, and Dr. Francis Kelly because he's a man from Carlow and for all the good time in and outside the lab.

I wish to express my deep gratitude to my co-supervisor Dr. Rozenn Dahyot for the time and precious help she gave me. And of course, most of all, I would like to thank my supervisor Dr. Anil Kokaram for his patience and intelligence in deciphering my obscure drafts, the good atmosphere he instigates in the lab, his contagious motivation and, above all, his constant efforts and help over the last 3 years. Thanks Anil.

And of course, I wish to thank my parents, my brothers and all my friends for their love and support and the good time I had in Dublin.

Contents

1 Introduction **8**

1.1 Transfer of Statistics in Image Processing . . . 9

1.2 Probabilistic Tracking . . . 9

1.3 Image Simplification . . . 10

1.4 Thesis Outline . . . 11

I Distribution Transfer and its Applications **13**

2 Distribution Transfer **14**

2.1 The Linear Case. . . . 16

2.2 The General Non-Linear Case . . . 19

2.2.1 The 1D case . . . 19

2.2.2 Extending the 1D Case to Higher Dimensions . . . 19

2.2.3 Description of the Manipulation . . . 20

2.3 Theoretical Convergence Study . . . 20

2.4 Experimental Convergence Study . . . 25

2.4.1 Kullback-Leibler Divergence . . . 25

2.4.2 Experimental Datasets: Sampling PDFs . . . 26

2.4.3 Results and Choice of Axis Sequence . . . 27

2.5 Conclusion . . . 31

3 Image re-Colouring **35**

3.1 Related Works . . . 37

3.2 Reducing Grain Noise Artefacts . . . 39

3.3 Results . . . 42

3.4 Conclusion . . . 43

4 Flicker Removal **47**

4.1 Related works . . . 48

4.2 Spatial and Temporal Integrations . . . 53

4.3 Estimation of the Mapping . . . 56
4.3.1 Flicker Compensation . . . 57
4.4 Practical Issues . . . 57
4.5 Results . . . 59
4.6 Conclusion . . . 62

5 Robust One-Dimensional PDF Transfer **63**
5.1 PDF Transfer Using the Distributions Separately . . . 64
5.2 PDF Transfer Using the Joint Distribution. . . . 69
5.3 Conclusion . . . 74

II Tracking Applications **78**

6 Contour Following using Particle Filters **79**
6.1 Contour Tracing . . . 79
6.2 Probabilistic Contour Tracking Framework . . . 81
6.2.1 Standard Approach using Particle Filters . . . 81
6.2.2 Exact Importance Sampling . . . 82
6.3 The Prior on the Contours . . . 83
6.4 Likelihood . . . 84
6.5 Results and Remarks . . . 86

7 Off-line Multiple Object Tracking **90**
7.1 Introduction . . . 90
7.2 Overview of the Methodology . . . 92
7.3 Application to a Simple Case Study . . . 93
7.4 Application to Multiple Objects Tracking . . . 93
7.4.1 Player Candidate Positions . . . 93
7.4.2 Set of Rules . . . 94
7.5 Conclusion . . . 96

III Image Simplification **99**

8 Towards Image Simplification **100**
8.1 Implicit Statistical Simplification . . . 102
8.2 MeanShift Filtering . . . 103
8.3 Smoothness Prior . . . 105
8.4 MeanShift with Implicit Prior . . . 107
8.5 Results . . . 111
8.6 Conclusion . . . 111

9 Closing Remarks **119**

A Appendix **121**

A.1 Sampling PDFs 121

A.2 Directions 123

A.3 Simple Matlab Code for 1D Signal 124

Chapter 1

Introduction

Digital media technology is currently in an evolutionary stage. What is observed is that technology domains like Computer Graphics, Computer Vision and Computer Vision are now converging. The convergence is patent in the research area where scientific papers can often be equivalently published in any of the conferences of these fields. The convergence is also reflected in the industry where media applications are reciprocally borrowing core technologies. One emblematic example of this would be the recent integration of Industrial Light and Magic®, the special visual effects branch of Lucasfilm Ltd., and the gaming division LucasArts®.

There is thus a need to push this convergence further and bring recent advances in image processing to bear on key problems for media applications and especially for post-production. Digital media post-production is an industry standard step in media creation, which includes all stages of production between the actual recording and the complete film or video. Now that issues of speed and physical storage have largely been rendered less problematic, the emphasis has shifted toward increasing levels of automation. Companies are already using image processing tools for simple automations. Only a few companies are however routinely using the most recent image processing advances. Examples of such companies are 2d3®with their feature-tracking engine Boujou, or The Foundry®with their set of plug-ins Furnace.

A number of key applications still need to be improved. Post-production tools like grading, deflicker, deshake, dirt removal or noise reducer are commodity tools that everybody needs but recognises that they have a broken one. This thesis proposes thus to make several contributions in the domain of visual post-production where the use of statistical digital signal processing techniques could help with some of these key problems.

1.1 Transfer of Statistics in Image Processing

The first application considered in this work concerns a major problem in the post-production industry, which is to change the 'look and feel' of the multitude of shots in such a way that they match the global atmosphere of the movie. Every aspect of the images has to be carefully controlled: colour palettes, textures, form of the grain and so on. This activity of *film grading* is currently fixed by experienced artists who manipulate the frames by tuning parameters and painting what is possible to paint. Sometimes the aspect of the film varies from frame to frame, like for instance in old footages where exposure time and film stock ageing can be inconsistent. Then the brightness and colour aspect varies across frames and produces some annoying *flicker* artefact. Similar fluctuations also arise in modern footages if the sequence is a composition of different camera sources. In these cases, a similar restoration process needs to be performed to adjust the aspect of the frames throughout the shot.

One novel aspect of this thesis is to demonstrate that a statistical approach can help automating this painstaking process. Consider the colour aspect of an image: each picture can be represented by the set of its colour samples. The colour property or 'feel' of a picture corresponds to the statistics of the colour samples. Then the grading the colour aspect of a frame can be done by transferring to the current frame the colour statistics of an example image that possesses the desired colour aspect.

The first contribution of this thesis is to propose a method that transfers the complete statistics of the samples and not simply the mean and the variance as it is usually done. The method, described in chapter 2, is to find a one-to-one mapping that transforms the original colour samples in a new set that exhibits the exact same propability density function (pdf) that the sample set of an example picture. With this method, it is easy to give the correct feel to an image sequence by simply providing a picture that has the wanted feel and then apply the corresponding mapping on each of the images. The first application of the method is presented in chapter 3 and is a direct use of the method to the problem of recolouring of images by example. The second application, presented in chapter 4, is part of the larger process of the digital movie restoration and concerns the stabilisation of colour level fluctuations across frames of a single shot or *flicker* removal. The last chapter of this part studies ways of estimating the transfer of statistics in the presence of occlusions or missing data like blotches and dirt.

1.2 Probabilistic Tracking

What transpires when working with images is that accessing higher content information can help performances of image and video processing algorithms. In the general sense, it would be advantageous to handle *objects* rather than *pixels* in the image. Unfortunately, object based segmentation is a decidedly non-trivial task and there is no reliable automated

mechanism. In the post-production industry however, objects are routinely manipulated. The objects are extracted using a combination of filming in a contrived environment (e.g. green-screen matting) or using manual delination of objects in each image frame. What is emerging is that for reliable segmentation, good enough to fool audiences, a combination of manual and automated tools are best. Two key operations are contour following to delineate an object in an image, and object tracking. Object tracking obviously has many applications outside of post-production.

The second main contribution of this thesis is to propose two probabilistic tracking algorithms that ease the burden of manual manipulation. Both methods proposed are founded on Bayesian grounds but adopt different optimisation strategies. The first method is used for semi-automated contour delineation—or rotoscoping. One recent technique called JetStream [Pér01] is a considerable advance on manual or semi-automatic tracing. The method, based on the use of Bayesian Filtering suffers however from a lack of direction information in the image. The method proposed in chapter 6 exploits a new prior that extracts local direction information and so reworks the principle of density propagation for contour following.

The second method is a deterministic method used to track multiple objects in videos. A major difficulty in tracking object is the dimension of the space of possible solutions. Popular stochastic methods like Particle Filters offer an attractive generic solution to this problem. These methods are simple to implement and can be easily scaled to any model complexity. It transpires however from a user point of view that deterministic methods are more suitable because reproduceable and predictable. The objective of this chapter is to show that it is often possible to simplify the object tracking problem in such way that it becomes tractable to provide a small set of deterministic position candidates and then use deterministic methods like Viterbi to track object among the reduced set of candidates.

1.3 Image Simplification

With both high-level information and low-level manipulations, it is now possible to develop media applications that would be content aware. This part explores the idea that objects of an images could be re-expressed in a different visual form that is not necessarily realistic but which conveys the same content. The aim is to develop representations that would have better viewability and that would be then transmitted or compressed easily.

Rendering non-photographic pictures has raised some interest in the computer graphic community, especially to design filters [Mig03] that simulate an artistic style. This concept is here pushed further by designing a non-photographic manipulation that focuses on the content of the images, whilst *simplifying* their representation.

The tools discussed previously could all be used in tandem to create a such an effect. The work presented in chapter 8 considers instead a purely implicit method to simplify signals. The work draws from the success of recent texture synthesis results that exploit

non-parametric image models. The method consists in finding similar patches in the image and then replacing them by only one image patch that represents them all. This can be illustrated by considering an image sequence. An object can appear on several frames and presents slightly different appearances on these frames. Then replacing these object instances by a unique representation would still preserve the content while simplifying the video.

1.4 Thesis Outline

This thesis is thus divided into tree distinct parts. Part one covers chapters 2-5, part two covers chapters 6 & 7 and part tree the chapter 8. The part one treats the problem of transferring statistics from one image to another. Chapter 2 proposes a new method for doing so by transferring the actual pdf of image feature samples from one image to another. The method is applied to example-based image recolouring in chapter 3, and then extended to videos in chapter 4 for the restoration application of flicker removal. Impatient (and reasonable) readers can first focus on the description of the method in chapter 2 up to section 2.3, and then read the less mathematical application chapters 3 & 4. Chapter 5 considers ways of robustly estimating the transfer in the presence of outliers (*e.g.* content difference). The second part, which covers chapter 6 & 7, presents two applications for probabilistic tracking. Finally chapter 8 concludes this thesis by presenting an implicit method for image simplification.

Chapter 2 proposes a novel method to estimate a continuous mapping that transfers one distribution to another. The distributions are possibly N-dimensional. The method proposed is iterative, and its convergence is studied in the second part of the chapter.

Chapter 3 applies the method proposed in chapter 2 to the difficult problem of example-based image recolouring. The distribution transfer technique is used in conjunction with a post-processing algorithm that re-grains and protects the picture content. The results demonstrates the effectiveness of the method.

Chapter 4 applies the method proposed in chapter 2 to the problem of restoring image sequences degraded by brightness and colour fluctuations. To treat this problem, referred to as flicker removal, this chapter considers the possible variations of the mapping, temporally and spatially.

Chapter 5 considers the estimation of the mapping in situations where occlusions and missing data (outliers) pollute the original data samples (inliers). The chapter proposes two methods to separate outliers from inliers and then estimate the mapping.

Part II

Chapter 6 considers the problem of probabilistic tracking applied to the semi-automated delineation of object contours. This chapter considers the incorporation of a better directional information in the JetStream algorithm [Pér01] and so reworks the principle of density propagation for contour following.

Chapter 7 considers shortcomings of MCMC approaches like Particle Filter methods as proposed in chapter 6 and points out that in many situations, it is possible to use a similar Bayesian framework but without having to resort to stochastic optimisation methods. The chapter illustrates this idea by presenting off-line multi-tracking applications.

Part III

Chapter 8 proposes an new implicit way of simplifying images by exploiting redundancy within images. The method aims at establishing statistics of pixel values according to their neighbourhood pixel values. Then by using Mean-Shift algorithm on the statistics pdfs results in an filtering that implicitly replaces similar objects by one instance of this object.

Contributions of this Thesis. This thesis offers new contributions which can be summarised by the following list.

- Transferring pdfs in N dimensions (chapter 2)
- Finding an optimised sequence of rotation matrices for the N-dimensional pdf transfer (chapter 2)
- Transferring pdf in 1D in the presence of outliers (chapter 5)
- Generating randomly pdfs (chapter 2 and appendix A)
- Removing grain structures that appear after over-stretched mapping (chapter 3)
- Stabilising brightness fluctuations in videos or Flicker Removal (chapter 4)
- Using GPU to speed up flicker removal (chapter 4)
- Using edge direction information to track contours in pictures using particle filter (chapter 6)
- Computing edge direction information (chapter 6)
- Using simple colour segmentation and deterministic algorithms (Viterbi) to track players in videos (chapter 7)
- Establishing neighbourhood statistics to integrate an implicit prior in Mean-Shift (chapter 8)

Part I

Distribution Transfer and its Applications

Chapter 2

Distribution Transfer

THE principle of example-based rendering is probably the simplest and most effective approach to rendering realistic images. The idea is to transform an original image in such a way that the resulting picture has the same 'look and feel' as an example picture. One efficient method of realising this effect is to extract the image statistics of the example, and transfer them on the original image.

The notion of image statistics covers a wide range of properties. For instance, in the case of texture, the relevant image statistics are those pertaining to texture samples, and it is these statistics that are transferred to the original image. In the case of colour transfer, the image is expressed in terms of colour samples.

The recent breakthrough in texture synthesis and texture transfer [Efr99, Efr01, Her01] is one of the most significant example of this idea. The texture transfer technique is based on a non-parametric approach, which consists simply in replacing the blocks in the original picture by the most similar blocks in a seed example texture. The resulting image consequently displays the same texture statistics as the target texture. As shown in figure 2.1, this technique gives impressive results, and suggests that achieving realistic rendering results relies on the transfer of real data statistics.

The original and example images can be represented as the two sets of feature samples $\{\mathbf{u}_i\}_{i\leq M}$ and $\{\mathbf{v}_i\}_{i\leq M'}$ respectively. The feature samples are possibly N-dimensional and depend on the application. The problem is to find a mapping t that transforms the original set $\{\mathbf{u}_i\}_{i\leq M}$ into a new set $\{t(\mathbf{u}_i)\}_{i\leq M}$ such that the statistics of the new data samples are equal to the statistics of the target set.

Parts of this chapter are based on the work published in [Pit05c]

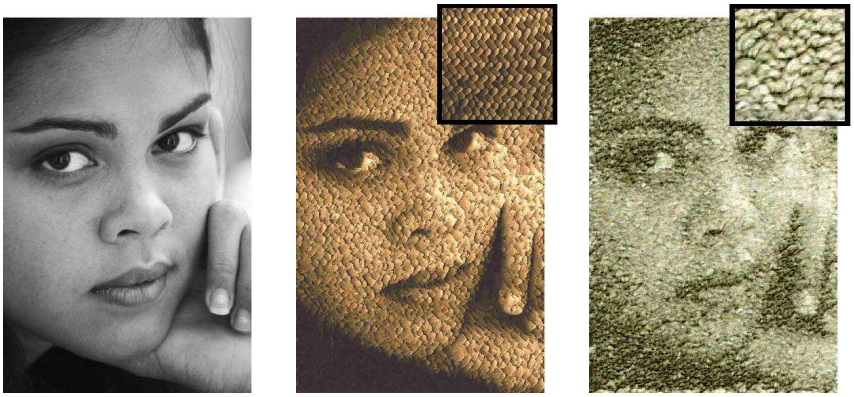


Figure 2.1: Example of Texture Transfer (from [Her01]).

Figure 2.2: Example of Colour Transfer. (see chapter 3)

There is a wide range of possible approaches to the transfer of statistics. For example a linear mapping suffices to match the means and the variances of the distributions. But if it is sought to transfer the actual probability density function (pdf) of the samples, the mapping becomes more complex. This latter problem will be referred to as the Distribution Transfer problem and is the subject of this chapter.

Distribution Transfer Problem. *Consider a set of M data samples $\mathbf{u}_i$. Denote by f the continuous pdf of these samples. The problem is to find an infinitely differentiable bijective mapping $t : \mathbf{u} \to t(\mathbf{u})$ that transforms the original set $\{\mathbf{u}_i\}_{i \leq M}$ into a new set $\{t(\mathbf{u}_i)\}_{i \leq M}$ such that the pdf f' of the transformed data samples is equal to a target continuous pdf g.*

There are a few constraints on the transformation. Firstly, the differentiable constraint is simply the mathematical expression that the transformation has to be smooth, in the sense that two feature samples that are similar in the original picture should still be loosely similar after transformation. Combined with the bijective constraint this means that the mapping has to be monotonic. This intuitive constraint has however some consequences of importance. For instance, under the monotonic constraint, it is impossible to allow swapping clusters of samples. Hence the operation in figure 2.3 is not differentiable, but simply at best piecewise differentiable. Also the folding-like transformation described on the left of figure 2.4 is not allowable. This is because where the mapping folds (red arrows), multiple points have the same target, and this violates the bijective constraint. The allowable space of monotonic mappings encompasses transformations equivalent to deforming a rubber sheet.

Bearing these issues in mind, after a brief study of the linear case, this section proposes a solution to the distribution transfer. Although the concept of distribution transfer can provide a framework for thinking about texture manipulation, it will not necessarily yield in an efficient algorithm. However, in the case of colour transfer, it is clearly an attractive solution, as shown in the next chapter.

2.1 The Linear Case.

Consider at first the case of a linear transformation as follows:

$$t(\mathbf{u}) = A\mathbf{u} + b \tag{2.1}$$

The parameters of the transformations A and b, can be derived by matching the first and second moments of both the original pdf f and the target pdf g. Denote as μ_f and μ_g the means of f and g, and Σ_f and Σ_g as the covariance matrices:

$$\Sigma_f = (\mathbf{u} - \mu_f)(\mathbf{u} - \mu_f)^{\mathrm{T}} \tag{2.2}$$

$$\Sigma_g = (\mathbf{v} - \mu_g)(\mathbf{v} - \mu_g)^{\mathrm{T}} \tag{2.3}$$

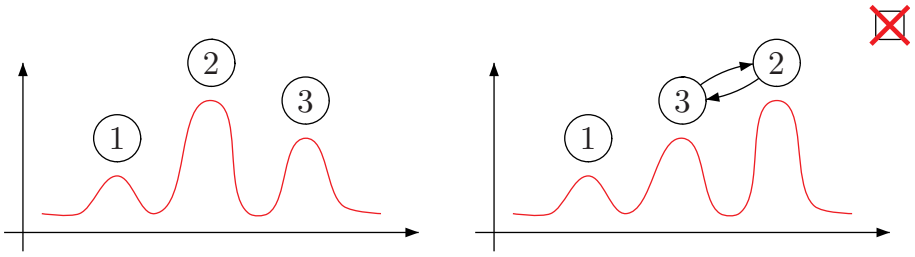


Figure 2.3: Monotonous limitations: clusters cannot be swapped.

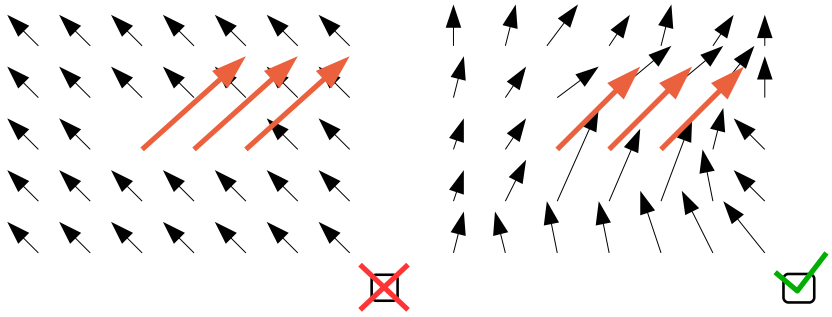


Figure 2.4: The Monotonous constraint does not allow folding mappings like the one on the left, because two points would have then the same mapping, which makes the inversion impossible. Monotonic mappings are equivalent to deforming a rubber sheet, where every point maps has a unique mapping.

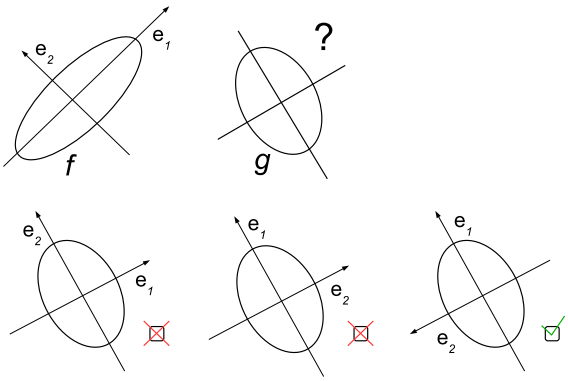


Figure 2.5: Matching 2 Normal distributions. The assignment of axis numbers and directions should be consistent for both distributions.

The linear transformation is thus as follows:

$$t(\mathbf{u}) = (\Sigma_g)^{\frac{1}{2}} (\Sigma_f)^{-\frac{1}{2}} (\mathbf{u} - \mu_f) + \mu_g \tag{2.4}$$

where $\Sigma_g^{\frac{1}{2}}$ is given by the Cholesky decomposition of Σ_g. Experience has shown however that the Cholesky decomposition offers insufficient control over the transformation. The preferred approach for this application is instead the singular value decomposition (SVD):

$$(\Sigma_f) = (\mathbf{E}_f^{\mathrm{T}} \mathbf{D}_f \mathbf{E}_f) = \left(\mathbf{D}_f^{\frac{1}{2}} \mathbf{E}_f\right)^2 \tag{2.5}$$

$$(\Sigma_g) = (\mathbf{E}_g^{\mathrm{T}} \mathbf{D}_g \mathbf{E}_g) = \left(\mathbf{D}_g^{\frac{1}{2}} \mathbf{E}_g\right)^2 \tag{2.6}$$

where $\mathbf{E}_f = [\mathbf{e}_f^1, \ldots, \mathbf{e}_f^N]$ and $\mathbf{E}_g = [\mathbf{e}_g^1, \ldots, \mathbf{e}_g^N]$ are the NxN orthogonal matrices containing the eigenvectors of the covariance matrices. The diagonal matrices $\mathbf{D}_f$ and $\mathbf{D}_g$ contain the eigenvalues corresponding to the eigenvectors in $\mathbf{E}_f$ and $\mathbf{E}_g$. The final transformation is thus as follows:

$$t(\mathbf{u}) = \mathbf{E}_g^{\mathrm{T}} \mathbf{D}_g^{-\frac{1}{2}} \mathbf{D}_f^{\frac{1}{2}} \mathbf{E}_f (\mathbf{u} - \mu_f) + \mu_g \tag{2.7}$$

The additional control offered by the SVD method derives from the possibilities in ordering the eigenvectors and choosing their direction (see figure 2.5). For example, changing the sign of an eigenvector results in swapping the data samples from one side of the eigenvector axis to the other side, which is not wanted. The idea is to preserve the *content* information by ordering the eigenvectors with respect to the magnitude of the corresponding eigenvalues and making sure that they do not point in opposite directions, *i.e.*

$$\forall\ i \leq N\ ,\ \mathbf{e}_f^{i\mathrm{T}} \mathbf{e}_g^i > 0 \tag{2.8}$$

2.2 The General Non-Linear Case

2.2.1 The 1D case

Consider now the general distribution transfer problem where it is sought to transfer the actual target pdf g. If the data samples are of dimension 1, the distribution transfer problem has a very simple solution. The differentiable mapping yields the following constraint which simply corresponds to a change of variables:

$$f(u)\mathrm{d}u = g(v)\mathrm{d}v \quad \text{with} \quad t(u) = v \tag{2.9}$$

Integrating both sides of the equality yields

$$\int^{u} f(u)\mathrm{d}u = \int^{t(u)} g(v)\mathrm{d}v \tag{2.10}$$

Using cumulative pdf notations F and G for f and g then yields the expression for the mapping t,

$$\forall u \in \mathbb{R}\ , t(u) = G^{-1}\left(F(u)\right) \tag{2.11}$$

where $G^{-1}(\alpha) = \inf\{u | G(u) \geq \alpha\}$. The mapping can then easily be solved by using discrete look-up tables.

2.2.2 Extending the 1D Case to Higher Dimensions

Extending the 1-dimensional case to higher dimensions is not trivial. The idea proposed here is to break down the problem into a succession of 1-Dimensional distribution transfer problems. Consider the use of the N-dimensional Radon Transform. It is widely acknowledged that via the Radon Transform, any N-dimensional function can be uniquely described as a series of projections onto 1-dimensional axes [Weia]. In this case, the function considered is a N-dimensional pdf, hence the Radon Transform projections result in a series of 1-dimensional marginal pdfs. Intuitively then, operations on the N-dimensional pdf should be possible through manipulations of the 1-dimensional marginals.

Consider that after some sequence of such manipulations, all 1-dimensional marginals match the corresponding marginals of the target distribution. It then follows that, by nature of the Radon Transform, the transformed f, corresponding to the transformed 1-dimensional marginals, now matches g.

There are now several questions to answer. How to manipulate the 1-dimensional pdfs? What guarantee is there of eventual convergence? How many axes are needed? Is there an optimal sequence of axes? The following paragraphs provide the answers to these questions.

2.2.3 Description of the Manipulation

The operation applied to the individual projection axes on the marginal distributions is naturally similar to that used in 1-dimension. Consider a particular axis denoted by its vector direction $\mathbf{e} \in \mathbb{R}^N$. The projection of both pdfs f and g onto the axis $\mathbf{e}$ results in two 1-dimensional marginal pdfs $f_\mathbf{e}$ and $g_\mathbf{e}$. Using the 1-dimensional pdf transfer mapping of the equation (2.11) yields a 1-dimensional mapping $t_\mathbf{e}$ along this axis:

$$\forall u \in \mathbb{R}\,,\; t_\mathbf{e}(u) = G_\mathbf{e}^{-1}\left(F_\mathbf{e}\left(u\right)\right) \tag{2.12}$$

For a N-dimensional sample $\mathbf{u} = [u_1, \cdots, u_N]^\mathrm{T}$, the projection of the sample on the axis is given by the scalar product $\mathbf{e}^\mathrm{T}\mathbf{u} = \sum_i e_i u_i$, and the corresponding displacement along the axis is

$$\mathbf{u} \rightarrow \mathbf{u} + \left(t_\mathbf{e}(\mathbf{e}^\mathrm{T}\mathbf{u}) - \mathbf{e}^\mathrm{T}\mathbf{u}\right)\mathbf{e} \tag{2.13}$$

After transformation, the projection $f'_\mathbf{e}$ of the new distribution f' is now identical to $g_\mathbf{e}$. The manipulation is explained in figure 2.6.

Considering that the operation can be done independently on orthogonal axes, the proposed manipulation consists in: choosing an orthogonal basis $\mathbf{R} = (\mathbf{e}_1, \cdots, \mathbf{e}_N)$ and then applying the following mapping τ:

$$\tau(\mathbf{u}) = \mathbf{u} + \mathbf{R}\begin{pmatrix} t_1(\mathbf{e}_1^\mathrm{T}\mathbf{u}) - \mathbf{e}_1^\mathrm{T}\mathbf{u} \\ \vdots \\ t_N(\mathbf{e}_N^\mathrm{T}\mathbf{u}) - \mathbf{e}_N^\mathrm{T}\mathbf{u} \end{pmatrix} \tag{2.14}$$

where t_i is the 1-dimensional pdf transfer mapping for the axis $\mathbf{e}_i$.

The idea is that iterating this manipulation over different axes will result in a sequence of distributions $f^{(k)}$ that hopefully converges to the target distribution g. The overall algorithm is described on a separate page and will be referred to as the **Iterative Distribution Transfer (IDT)** algorithm.

Although the theoretical study presented in section 2.3 does not provide yet a proof of convergence in all cases, the experimental results presented on figures 2.7 and 2.12 clearly show that the method can be practically used as it is. Impatient readers can thus skip the rest of the chapter and resume their reading at chapter 3 which presents a direct application of the method to colour grading. Note however that the method can be substantially speeded up by using an optimised rotation sequence as proposed in section 2.4.3.

2.3 Theoretical Convergence Study

It is now important to investigate the convergence properties of this algorithm. The following theorem establishes the convergence of the algorithm when the target pdf is the standard normal distribution (denoted by $\mathcal{N}(\mathbf{0}, \mathbf{id}_N)$). Thus by combining the mappings

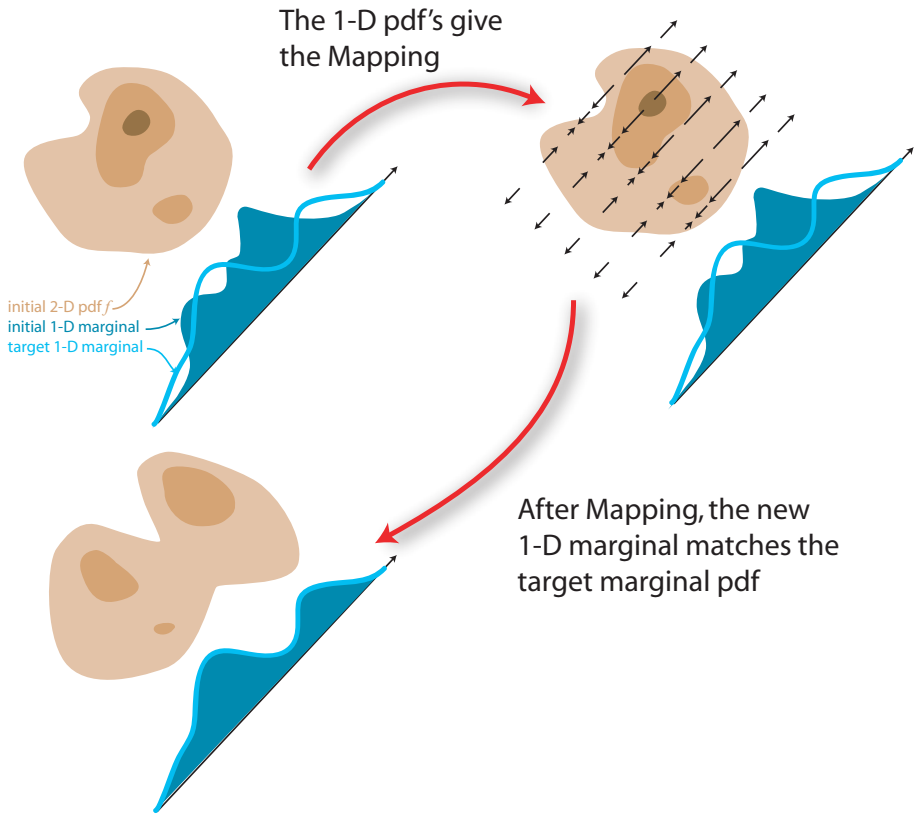


Figure 2.6: Illustration of the data manipulation, based on the 1-dimensional pdf transfer on one axis.

Algorithm 1 IDT method

1: **Initialisation** of the data set source $\mathbf{u}$
$k \leftarrow 0$, $\mathbf{u}^{(0)} \leftarrow \mathbf{u}$
2: **repeat**
3: for every rotated axis i, get the projections f_i and g_i
4: find the 1D transformation t_i that matches the marginals f_i into g_i
5: remap the samples $\mathbf{u}$ according to the 1D transformations:

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{R} \begin{pmatrix} t_1(\mathbf{e}_1^{\mathrm{T}}\mathbf{u}^{(k)}) - \mathbf{e}_1^{\mathrm{T}}\mathbf{u}^{(k)} \\ \vdots \\ t_N(\mathbf{e}_N^{\mathrm{T}}\mathbf{u}^{(k)}) - \mathbf{e}_N^{\mathrm{T}}\mathbf{u}^{(k)} \end{pmatrix}$$

6: $k \leftarrow k + 1$
7: **until** convergence on all marginals for every possible rotation
8: The final one-to-one mapping T is given by: $\forall\ j\ , \mathbf{u}_j \mapsto t(\mathbf{u}_j) = \mathbf{u}_j^{(\infty)}$

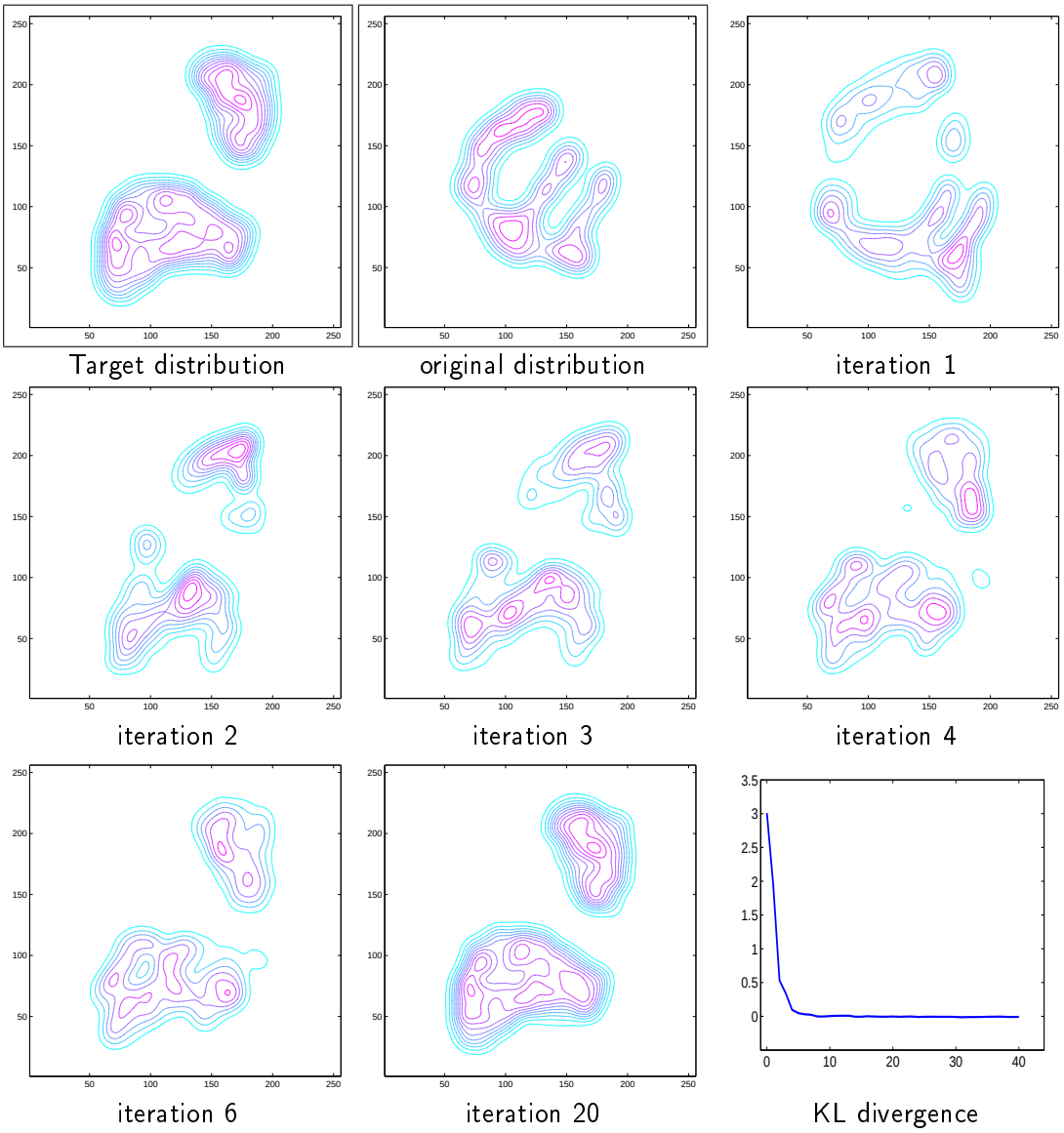


Figure 2.7: Example of a two dimensional pdf transfer using IDT. The decrease of the Kullback-Leibler divergence illustrates the convergence of the method.

at each iteration, it is possible to find for any distribution f a differentiable bijective mapping $t_f = \tau_k \circ \cdots \circ \tau_1$ that transforms f into the standard distribution $\mathcal{N}(\mathbf{0}, \mathbf{id}_N)$. Consider then the two differentiable mappings t_f and t_g that transform f and g into $\mathcal{N}(\mathbf{0}, \mathbf{id}_N)$. Using the standard distribution $\mathcal{N}(\mathbf{0}, \mathbf{id}_N)$ as a pivot results in the transformation $t_g^{-1} \circ t_f$, which is a differentiable bijective mapping that transforms f into g, and is therefore a solution of the distribution transfer problem:

$$\begin{aligned} t_f : &\quad f \to \mathcal{N}(\mathbf{0}, \mathbf{id}_N) \\ t_g : &\quad g \to \mathcal{N}(\mathbf{0}, \mathbf{id}_N) \\ \forall \mathbf{u} \in \mathbb{R}^N &, \; t(\mathbf{u}) = t_g^{-1}\left(t_f(\mathbf{u})\right) \end{aligned} \tag{2.15}$$

The results of the convergence study in the next several paragraphs strongly suggest that the algorithm converges to the solution even if the target distribution is not the normal distribution. Figure 2.7 shows an example illustrating the convergence of the process for 2-dimensional pdfs. Thus the method can also be used directly to find a mapping between f and g, without having to resort to the standard distribution as a pivot. The results of the following chapters are based on this assumption and use the algorithm directly.

Theorem 1 (Isotropic Case). *Let f be an N-dimensional pdf and denote by g the pdf of the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{id}_N)$. Consider the differentiable transformation τ_k that matches as in equation (2.14) the marginals of f to the projections of g for a random set of axes. Then the sequence defined by $f^{(0)} = f$ and $f^{(k+1)} = \tau_k(f^{(k)})$ converges to the target distribution: $f^{(\infty)} = g$.*

Proof. Denote the original pdf f and the target standard normal pdf g. For a particular set of axes, denote $f_1, \cdots, f_N$ the marginals of f and $g_1, \cdots, g_N$ the marginals of g. The standard distribution is isotropic and for all axes, it can be written as the product of its marginals: $g = g_1 \cdots g_N$. The key of the proof is to show that the Kullback-Leibler divergence decreases for any set of axes.

The Kullback-Leibler divergence, or relative entropy, is a quantity which measures the difference between two probability distributions. It is computed as follows:

$$D_{KL}(f \| g) = \int_{\mathbf{u}} f(\mathbf{u}) \ln\left(\frac{f(\mathbf{u})}{g(\mathbf{u})}\right) \, \mathrm{d}\mathbf{u} \tag{2.16}$$

As with many measures over distributions, the KL divergence is not a proper distance. The KL divergence does not satisfy the triangular inequality and is not symmetric. However

the KL divergence is always non-negative and $D_{KL}(p,q)$ is zero iff $p=q$.

$$\begin{aligned} D_{KL}(f\|g) &= \int_{\mathbf{u}} f(\mathbf{u}) \ln\left(\frac{f(\mathbf{u})}{g(\mathbf{u})}\right) \mathrm{d}\mathbf{u} \\ &= \int_{\mathbf{u}} f(\mathbf{u}) \ln\left(\frac{f(\mathbf{u})}{f_1(u_1)\cdots f_N(u_N)}\right) \mathrm{d}\mathbf{u} + \int_{\mathbf{u}} f(\mathbf{u}) \ln\left(\frac{f_1(u_1)\cdots f_N(u_N)}{g(\mathbf{u})}\right) \mathrm{d}\mathbf{u} \\ &= D_{KL}(f\|f_1\cdots f_N) + \int_{\mathbf{u}} f(\mathbf{u}) \ln\left(\frac{f_1(u_1)\cdots f_N(u_N)}{g_1(u_1)\cdots g_N(u_N)}\right) \mathrm{d}\mathbf{u} \end{aligned} \tag{2.17}$$

Then by marginalising,

$$\begin{aligned} \int_{\mathbf{u}} f(\mathbf{u}) \ln\left(\frac{f_1(u_i)}{g_i(u_i)}\right) \mathrm{d}\mathbf{u} &= \int_{u_1}\cdots\int_{u_N} f(u_1,\cdots,u_N) \ln\left(\frac{f_1(u_i)}{g_i(u_i)}\right) \mathrm{d}u_1\cdots\mathrm{d}u_N \\ &= \int_{u_i} f(u_i) \ln\left(\frac{f_1(u_i)}{g_i(u_i)}\right) \mathrm{d}u_i \\ &= D_{KL}(f_i\|g_i) \end{aligned} \tag{2.18}$$

Eventually it follows that

$$D_{KL}(f\|g) = D_{KL}(f\|f_1\cdots f_N) + \sum_{i=1}^{N} D_{KL}(f_i\|g_i) \tag{2.19}$$

Consider now that the mapping transforms f into f' and $f_1\cdots f_N$ into $f'_1\cdots f'_N$. It holds for f' that:

$$D_{KL}(f'\|g) = D_{KL}(f'\|f'_1\cdots f'_N) + \sum_{i=1}^{N} D_{KL}(f'_i\|g_i) \tag{2.20}$$

The transformation is such as for each axis i, $f'_i = g_i$, thus $\sum_{i=1}^{N} D_{KL}(f'_i\|g_i) = 0$. Also the KL divergence is left invariant by bijective transformation. This implies that $D_{KL}(f'\|f'_1\cdots f'_N) = D_{KL}(f\|f_1\cdots f_N)$. Thus the KL difference decreases at each iteration by:

$$D_{KL}(f\|g) - D_{KL}(f'\|g) = \sum_{i=1}^{N} D_{KL}(f_i\|g_i) \geq 0 \tag{2.21}$$

Since $D_{KL}(f\|g)$ is non-negative, it must have a limit. This implies that the KL divergence on the marginals $D_{KL}(f_i\|g_i)$ convergences to 0. Then, if a sufficient number of different axes is considered, all marginals of f converge to the marginals of g, and the Radon transform of $f^{(k)}$ converges to the Radon transform of g. The Radon transform admits a unique inverse, and this shows also that the pdf $f^{(k)}$ converges to g. □

2.4 Experimental Convergence Study

This section studies the convergence of the algorithm when the standard distribution is not used as a pivot. The study also considers the problem of finding sequence of axes that maximises the convergence speed of the algorithm.

To explore the convergence of the algorithm, a measure is needed to quantify how well the transformed distribution $f^{(k)}$ matches the target pdf g. Following the remarks of the previous theorem, the Kullback Leibler *divergence* is used as such a measure. One simple experiment that could be used to assess the impact of axis sequences on convergence is simply to choose two particular datasets, use one as a target pdf and the other as a source pdf. Then for various axis sequences, the KL measure could be used to assess convergence as each iteration of the algorithm proceeds.

In order to provide more evidence for convergence though, it is sensible to use instead an ensemble of datasets which would cover the space of valid pdfs: in a sense then, a Monte Carlo method for assessment would be more useful. Therefore the problem is to generate at random, valid pdfs which then could be used to generate datasets for the application of the Distribution Transfer algorithm. Having done this, an ensemble average convergence or *expected* convergence measure could be obtained for a particular axis sequence using the mean KL divergence. This is as follows:

$$\mathbb{E}_{f,g}\left(D(f\|g)\right) \approx \frac{1}{N_s} \sum_{\substack{i \leq N_s \\ f_i, g_i \in \mathcal{P}}} D(f_i\|g_i) \tag{2.22}$$

where $\mathcal{P}$ denotes the space of the pdfs, thus each element $p \in \mathcal{P}$ is a pdf. Note that the estimation of the KL divergence expectation is dependent on the way the pdfs are chosen: *i.e.* they should be well distributed over the space for the expectation not to be biased. Assuming no prior on the pdfs, they should be generated uniformly over the ensemble of all possible pdfs. This turns out to be an interesting problem, and is discussed in the following sections after some attention to the numerical estimation of the KL divergence.

With this measure of improvement, it is now possible to study the behaviour of different axis sequences. The last paragraph of this section presents an heuristic way of finding a sequence, which shows near optimal improvements.

2.4.1 Kullback-Leibler Divergence

The experiments presented here, employ the datasets generated from the pdfs. The Distribution Transfer technique is then applied to those datasets, and not to the underlying pdfs. Hence the KL divergence must employ the kernel density approximation [Sil86] for

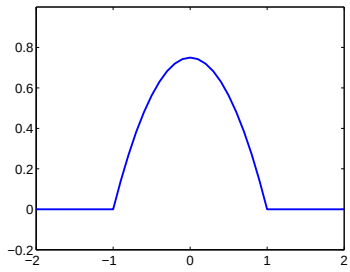


Figure 2.8: Epanechnikov Kernel. This is the function $K(\mathbf{u}) = (3/4)(1 - \|\mathbf{u}\|^2)$ for $-1 < \|\mathbf{u}\| < 1$ and zero for $\|\mathbf{u}\|$ outside that range.

the underlying pdfs as follows:

$$D(f \parallel g) = \frac{1}{M} \sum_{i=1}^{M} \ln \left[\frac{\sum_j K\left(\frac{\mathbf{u}_i - \mathbf{u}_j}{h_i}\right)}{\sum_j K\left(\frac{\mathbf{u}_i - \mathbf{v}_j}{h_i}\right)} \right] \tag{2.23}$$

where K is the Epanechnikov kernel (see figure 2.8). To account for the sparseness of samples, it is crucial to use a variable bandwidth approach. Typically, for a sample $\mathbf{u}_i$, the bandwidth h_i increases if with the sparseness of the samples around $\mathbf{u}_i$. A clear outline of the bandwidth selection is available in [Com01], and that is used here. A major aspect of the experiment is that the KL divergence has to be non-negative. This is indeed observed in numerical results (see figure 2.7), but not for any choice of bandwidth values. To counterbalance this sensitivity, the pdf is over-smoothed by taking larger values of kernel bandwidths. The resulting KL divergence measure is under-evaluated since both estimated pdfs are more uniform. Another consequence is that the convergence speed measured on the figure 2.12 is actually slower than the true one.

2.4.2 Experimental Datasets: Sampling PDFs

As discussed above, measuring the KL divergence on a single example is of course insufficient to infer any useful information. The estimation of the average KL measure requires to evaluate the KL divergence for a large number of datasets (say at least 100), and it is thus necessary to find a way of generating these pdfs. In this study, there is no prior made on the distributions. This means that the pdfs of the datasets have to be generated in a uniform 'random' way. Once a pdf has been generated, the set of N-dimensional data samples is obtained by sampling directly from the newly generated pdf.

Generating pdfs becomes quickly intractable for high dimensions because of the large number of bins involved. For instance in dimension 3, a pdf with a resolution of 256 bins for each coordinate, contains $256^3 > 16$ million bins. A solution is to approximate the pdf by a sum of kernel functions. This imposes some spatial smoothness and risks to under-

estimate the KL divergence. But since the KL estimation already over-smooths the pdf estimation, this does not present a major issue. The kernel approximation is as follows:

$$q(\mathbf{u}) = \frac{1}{k}\sum_{i=1}^{k} \frac{q_i}{h_i^N}\ K\left(\frac{\mathbf{u}-\mu_\mathbf{i}}{h_i}\right) \tag{2.24}$$

where N is the dimension of the pdf space, μ_i are the centres of the kernels, h_i the associated bandwidth, and $(q_i)_{i\leq k}$ a set weights such that $\sum_{i=1}^{k} q_i = 1$ and $q_i \geq 0$. Since no prior information is available on the pdf, it is standard to consider a flat prior for the centres and the standard deviations:

$$p(\mu_i) \propto 1 \tag{2.25}$$

$$p(h_i) \propto \frac{1}{h_i} \tag{2.26}$$

The centres μ_i are thus generated by sampling from a from a simple uniform spatial distribution on the region of interest, or data range, Ω: $\mu_i \sim \mathcal{U}(\Omega)$. Since the prior $p(h_i)$ on the bandwidths is not a proper prior (it does not integrate to 1), the range of values for the bandwidth is restricted from 1% to 100% of the data range. For instance, with a data range of $[0; 255]$, the diametre is 256, and the bandwidth can go from 2.56 to 256. As seen in figure 1, the lower bound of the bandwith values controls the smoothness of the generated pdf.

It still remains to generate the weights $(q_i)_{i\leq k}$. Since $(q_i)_{i\leq k}$ is non-negative and sums up to 1, this is the pdf of a random variable having k possible states, and will be referred to as a k-state pdf. It is shown in the appendix A.1 a method for generating a k-state pdf presents. The method is to generate k exponential random variables $(z_1, z_2, \cdots, z_k)$. Then the distribution of the vectors

$$\frac{1}{z_1 + \ldots + z_k}\begin{pmatrix} z_1 \\ \vdots \\ z_k \end{pmatrix} \tag{2.27}$$

is uniform over the k-state pdf space (see appendix A.1 for a proof). Now that all the parameters μ_i, h_i, q_i are known, it is possible to generate valid pdfs uniformly over the pdf space. The overall method is summarised by Algorithm 2, and examples of 1-dimensional and 2-dimensional pdfs generated by the method are displayed on figure 2.10.

2.4.3 Results and Choice of Axis Sequence

With the help of the pdf generator, it is now possible to study the convergence of the distribution transfer method by measuring the mean KL divergence over the iterations. In the following experiments, the plots are obtained by averaging the KL measure for

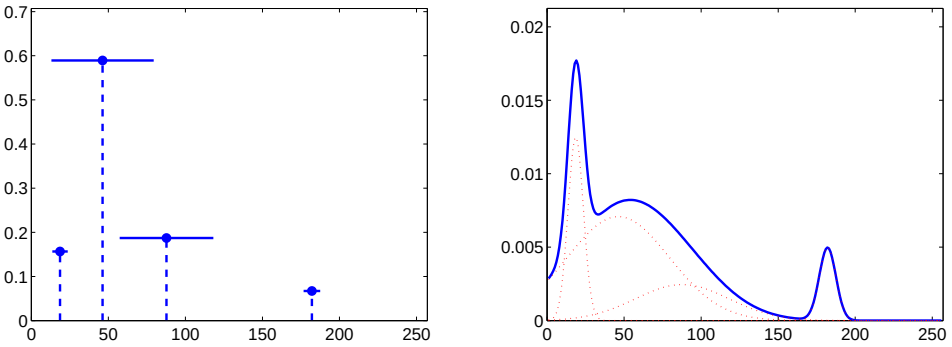


Figure 2.9: Generating a pdf. The pdf here is modelled by a sum of 4 kernel functions. On the left, the positions of the vertical bars correspond to the positions of centroids, the heights of the bars to the weight and the width of the overhead horizontal line to the size of the bandwidth (these values are generated by algorithm 2). The resulting pdf using gaussian kernels is presented on the right.

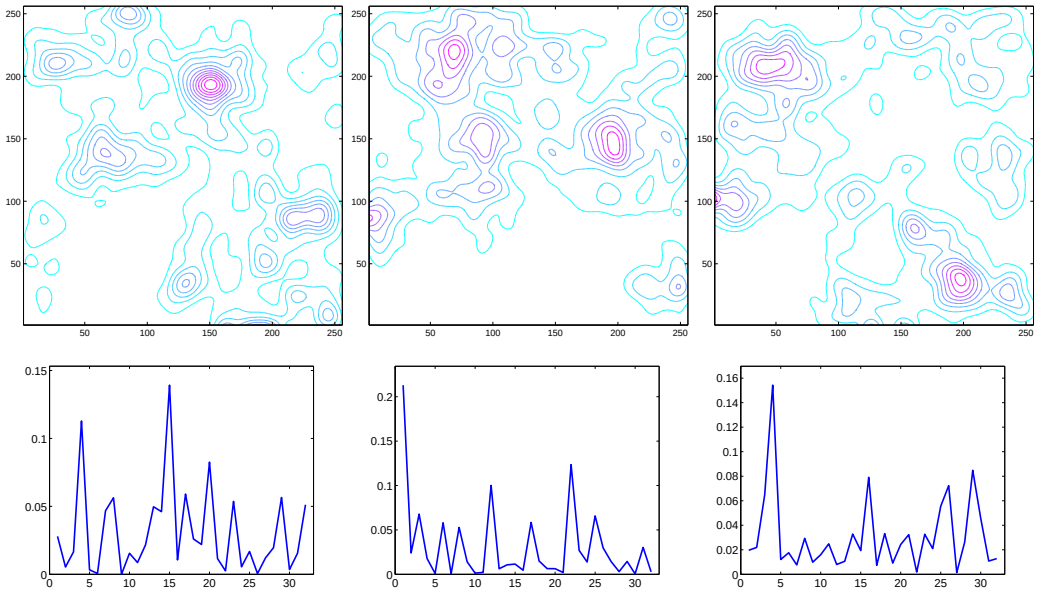


Figure 2.10: Examples of two dimensional and one dimensional pdfs generated by algorithm 2 .

Algorithm 2 Sampling pdfs randomly

The pdf is modelled by a sum of kernel functions

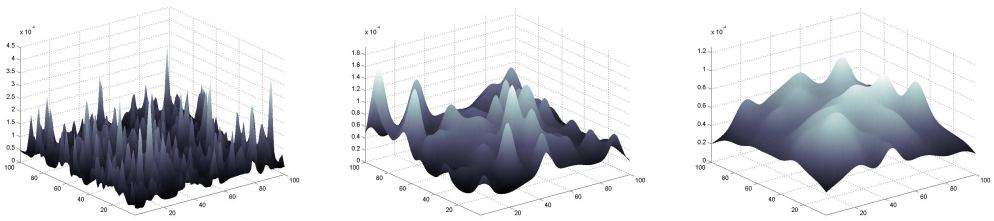
$$q(\mathbf{u}) = \frac{1}{k}\sum_{i=1}^{k} \frac{q_i}{h_i^N}\, K\left(\frac{\mathbf{u}-\mu_\mathbf{i}}{h_i}\right) \tag{2.28}$$

The parameters are sampled as follows:

$$p(\mu_i) \propto 1 \quad \mu_i \in \Omega \tag{2.29}$$

$$p(h_i) \propto 1/h_i \tag{2.30}$$

$$q_i = \frac{e_i}{e_1 + \ldots + e_k} \text{ with } p(e_i) \propto \exp(-e_i) \tag{2.31}$$



min bandwidth = 1% min bandwidth = 3% min bandwidth = 5%

Figure 2.11: The pdf smoothness can be controlled by the minimum bandwidth size allowed. For example on the left, the minimum bandwidth is set to 1% of the data range.

100 different pdfs. Figure 2.12 shows the evolution of the KL divergence for a random selection of axis set sequences, with both random initial and target distributions. This result strongly suggest that the convergence occurs when the target distribution is not the standard distribution but any distribution. This in an important result since it gives ground to use the IDT algorithm, without having to use the standard distribution as a pivot.

Taking a random selection of orthogonal bases seems to be sufficient to obtain convergence, however it is shown hereafter, that a better choice of axis sequence can still substantially improve the convergence speed. This can be understood by considering that random axes are probably correlated. Figure 2.13 shows that for small dimensions, two random vectors are more likely to be aligned than orthogonal. This result can be extended to the distance between orthogonal bases: two random orthogonal basis are more likely to be similar in dimension 2 than in dimension 10. Figure 2.14 confirms this intuition. The figure displays the average KL divergence after two iterations of the algorithm for 2D pdfs. At iteration 1, a fixed set of axes is chosen, thus the plot shows the evolution of the KL divergence depending on the choice of axes at iteration 2. The graph clearly shows that the KL improvement depends on the correlation between axes.

Intuitively then, an interesting heuristic would be to consider a sequence of rotations that maximises the distances between the current axis set at iteration k and the previous axis sets. Define the distance between two axes by

$$d(\mathbf{e_1}, \mathbf{e_2}) = \min(\|\mathbf{e_1} - \mathbf{e_2}\|_2, \|\mathbf{e_1} + \mathbf{e_2}\|_2) \tag{2.32}$$

with $\mathbf{e_1}$ and $\mathbf{e_2}$ the supporting axis vectors. To find axes that are far apart, one solution is to maximise the distances $d(\mathbf{e_1}, \mathbf{e_2})$. This turns out to a numerically unstable solution. A better formulation is to consider instead the minimisation of the potential $1/(1+d(\mathbf{e_1}, \mathbf{e_2}))$ and then to express that distances should be far appart with the following recursion:

$$\left[\mathbf{e}_{k+1}^1, \cdots, \mathbf{e}_{k+1}^N\right] = \arg \min_{[\mathbf{e}^1, \cdots, \mathbf{e}^N]} \left\{ \sum_{l=1}^{k} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{1}{1 + d(\mathbf{e}_l^j, \mathbf{e}^i)} \right\} \tag{2.33}$$

with the constraint that the bases are orthogonal. This minimisation problem can be numerically simulated under MATLAB using standard minimisation algorithms. The constraint of normalisation $\|e\| = 1$ can be avoided by taking hyperspherical coordinates [Wik]. The orthogonality of the base can be obtained using Gram-Schmidt orthogonalisation. The resulting first bases for dimension 2 and 3 are given in appendix tables A.1 and A.2. Note that since the algorithm is iterative, it is not crucial to require high accuracy in the estimation of the bases. The code used to find the bases is available on [Pit06].

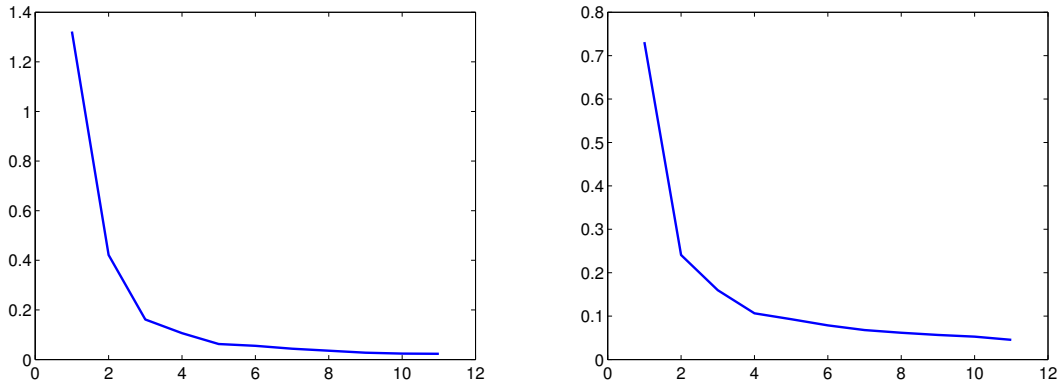
The improvements on the convergence speed are shown on figures 2.15 and 2.16. One can expect in average a speed improvement of 1.5 in 3D to 2.15 in 2D. The sequence which

is sought, is optimal on average over the couple of pdfs. Note that other strategies could be explored, like finding the optimal sequence of axis for a particular couple of distributions.

2.5 Conclusion

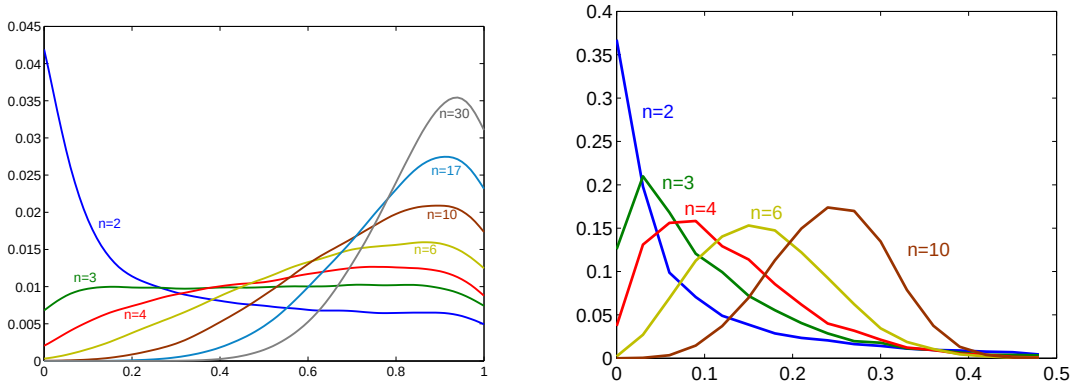
Transferring statistics from one image to another is a non-trivial task if the samples are multidimensional. The method which is proposed is attractive since it fast and the results strongly suggests that the it converges for any continuous distribution. Note as well that the implementation of the method is straightforward. The following chapters explores two applications of this technique that work on the colour aspect of images.

It has been discovered towards the end of this PhD. that the problem exposed in this chapter shares similitude with the problem of Mass Transfer. More details about this problem can be found in [Eva99]. The original problem, first proposed by Monge in the 1780's, is to minimise the amount of work needed to move a pile of soil to an excavation. Using modern notations, the pile and excavation correspond to two pdfs f and g, and the problem is equivalent to find a bijective mapping, that transforms f into g. The problem turns out to be extremely difficult and is still a subject of research two hundred years after. One reason is the complexity of the non-linear terms in the cost function and the constraint. The problem presented in this chapter differs from the Monge problem since it is not sought to minimise the transportation displacement. Note however that the problem resolved in this thesis offers satisfying results for low computational cost. Future research will examine the method developed in this thesis in light of the Monge problem.



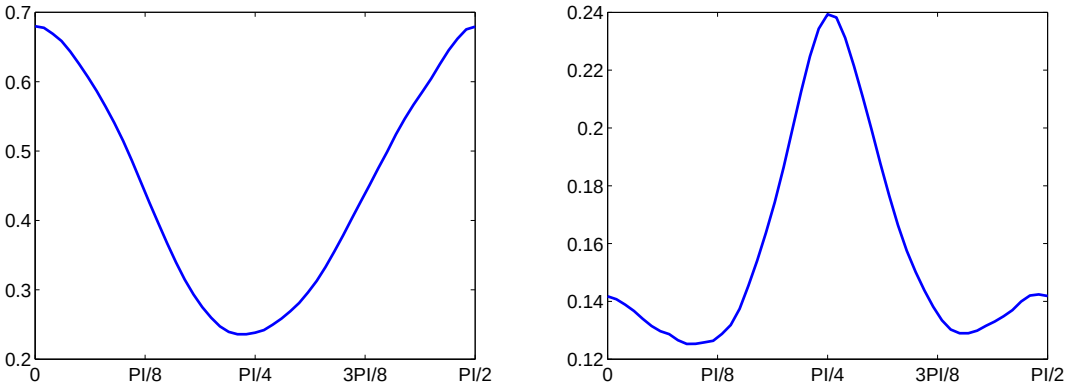
average KL in 2D average KL, in 3D

Figure 2.12: **Averaged evolution of the Kullback Leibler divergence for 100 simulations with both random initial distribution and target distribution. Rotations are taken randomly. It transpires from the results that convergence occurs for any distribution.**



(a)- distance between vectors (a)- distance between matrices

Figure 2.13: **In (a): experimental measure of the average distance between two unit vectors in function of the space dimension. The distance between two vectors u and v is given by $d(u,v) = \min(\|u-v\|_2, \|u+v\|_2)$. The average distance clearly increases with the dimensionality, which in other words shows that random vectors are likely to be nearly orthogonal in high dimensions. The figure (b) extends the results to the distance between matrices of space transformation. The distance between two matrices $A = [a_1, \cdots, a_n]$ and $B = [b_1, \cdots, b_n]$ is given by $d(A,B) = \min_{i,j} d(a_i, b_j)$. As in (a), the average distance clearly increases with the dimensionality. These two results imply that taking random directions in high dimension spaces does not compromise efficiency since the corresponding directions between 2 iterations of the algorithm are naturally far apart. On the contrary, a random selection strategy is surely non-optimal for low dimension.**



(a) - $k = 2$ (b) - $k = 3$

Figure 2.14: Average KL divergence to target distribution in function of the rotation sequence. The distributions are 2D and the rotation matrix can be represented by a single angle α between 0 and $\pi/2$. On the left, after one iteration at $\alpha = 0$. The best rotation is for $\alpha = \pi/4$, which corresponds to the less correlated axes. On the right, after $\alpha = 0, \pi/4$. The best new rotation occurs around $\alpha = 3\pi/8$ and $\pi/8$.

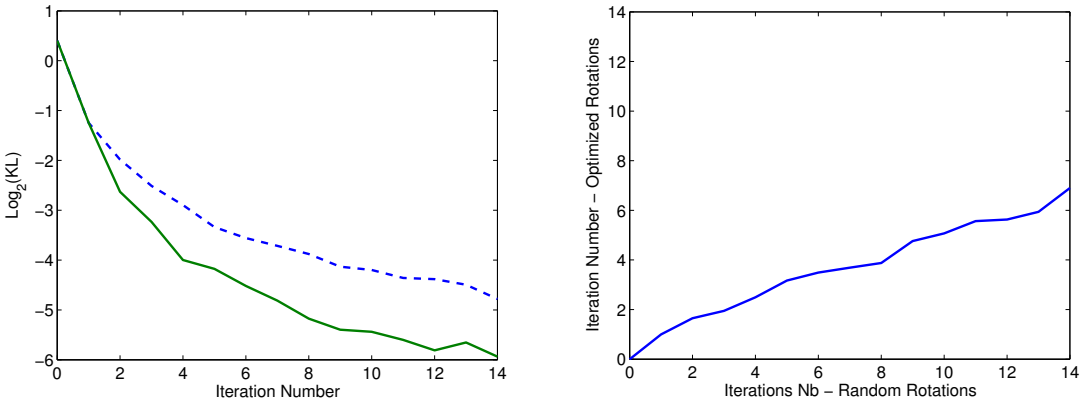


Figure 2.15: Optimised Rotation Sequence Speed Up for $N = 2$. On the left, the averaged evolution of the log Kullback Leibler divergence for 100 simulation for both a random sequence of rotations and an optimised sequence of rotations. On the right, the correspondence between iteration numbers for both strategies. It requires in average 53% less iterations for $N = 2$ when using an optimized rotation sequence.

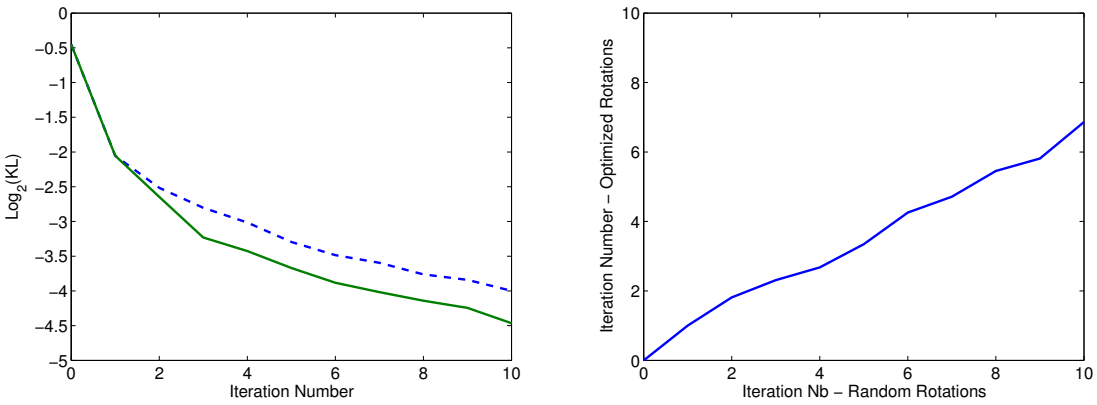


Figure 2.16: Optimised Rotation Sequence Speed Up for $N = 3$. On the left, the averaged evolution of the log Kullback Leibler divergence for 100 simulation for both a random sequence of rotations and an optimised sequence of rotations. On the right, the correspondence between iteration numbers for both strategies. It requires in average 33% less iterations for $N = 3$ when using an optimized rotation sequence.

Chapter 3

Image re-Colouring

THERE are a wide range of applications for the notion of exact transfer of distribution for multidimensional datasets. This chapter considers the difficult problem of example-based re-colouring [Rei01]. The idea of example-based re-colouring is illustrated by the picture below. The 'mountain' picture is required to be transformed so that its colours match the palette of the 'plain' image, regardless of the content of the pictures.

Original Colour Palette re-Coloured Image

Consider the two pictures as two sets of three dimensional colour pixels. A way of treating the re-colouring problem would be to find a one-to-one colour mapping that is applied for every pixel in the original image. For example in the diagram above, every white pixel is re-coloured in blue. Then the new picture is identical in every aspects to the original picture, except that the picture now exhibits the same colour pdf, or palette, as the target picture.

Estimating the mapping can be difficult, but if it is supposed to be *continuous*, a solution is had through the distribution transfer techniques for three dimensional datasets

parts of this chapter are based on the work published in [Pit05c, Pit05d]

as discussed in previous chapter. Thus this chapter examines the application of the distribution transfer techniques in the context of image re-colouring. Note that the *continuous* assumption only transcribes the intuitive idea that two colours that are perceptually similar should be also mapped to similar colours. This also implies that colours cannot be swapped. For instance, it is impossible to map a white to black and a black to white in the same picture. Figure 3.1 illustrates this contrast limitation. To realise a successful transfer, the sky of the picture should be segmented and treated separately.

Re-colouring picture with another has many applications in Computer Vision and in the Post-Production field. In digital restoration [Pap00] the idea is to recolour paintings that have been faded by smoke, dust etc. The process can also be used for colour image equalisation for scientific data visualisation [Pic03] or simply useful for non-realistic rendering.

A major problem in the post production industry is matching the colour between different shots possibly taken at different times in the day. This process is part of the large activity of *film grading* in which the film material is digitally manipulated to have consistent grain and colour. The term *colour grading* will be used specifically to refer to the matching of colour. Colour grading is important because shots taken at different times under natural light can have a substantially different 'feel' due to even slight changes in lighting. Currently these are fixed by experienced artists who manually match the colour between frames by tuning parameters. This is delicate task since the change in lighting conditions induces a very complex change of illumination. The method presented in this chapter however succeeds in automating this painstaking process even when the lighting conditions have dramatically changed as shown in figure 3.6.

The one-to-one colour mapping to the original picture makes the transformed picture have the same 'feel' of the picture example but it might also produce some grain artefacts on parts of the picture. This can be understood by considering a mapping from a low dynamic range to a high dynamic range. The resulting mapping is stretched and thus enhances the noise level of the picture, which makes the transformed picture appear grainy. The second step in colour grading is therefore to reduce this artefact. The method proposed is to use a variational approach to preserve the gradient of the original while preserving also the colour transfer characteristic. Preserving the gradient of the original picture especially protects the flat areas and more generally results in preserving the nature of the film grain/noise as in the original image.

There has been no prior art targeted specifically to the colour transfer problem in the film and digital video industry. However, there are a few articles which disclose ideas that could be used. These are discussed next.

3.1 Related Works

Transfer of Colour Statistics. One popular method proposed by Reinhard [Rei01] matches the mean and variance of the target image to the source image. The transfer of statistics is performed separately on each channel. Since the RGB colour space is highly correlated, the transfer is done in another colourspace $l\alpha\beta$. This colourspace has been proposed in an effort to account for for human-perception of colour [Rud98]. But the method is limited to linear transformations. In fact, the motion picture industry employs routinely non-linear colour grading techniques. Hence, in a practical situation, some example-based recolouring scenarios actually require non-linear colour mapping. Figure 3.2 shows exactly this problem and the method fails to transfer any useful statistics.

The problem of finding a non-linear colour mapping is addressed in particular in [Pic03] for colour equalisation (c.f. grayscale histogram equalisation). That work proposes to deform tessellation meshes in the colour space to fit to the 3D histogram of a uniform distribution. This method can be seen as being related to warping theory which is explicitly used in [Luc01] where the transfer of the 2D chromatic space is performed directly by using a 2D-bi-quadratic warping. Without having to invoke image warping, a natural extension of the 1D case is to treat the mapping via linear programming and the popular Earth-Mover distance [Mor03]. The major disadvantage of the method is that 1) the mapping is not continuous and 2) pixels of same colours may be mapped to pixels of different colours, which require random selection. Furthermore the computational cost becomes intractable if a very fine clustering of the colour space is desired.

Dealing with Content Variations. One important aspect of the colour transfer problem is the change of content between the two pictures. Consider a pair of images which are of landscapes but in one picture the sky covers a larger area than the other. When transferring the colour from one picture to the other therefore, the sky colour may be applied also to parts of the scenery on the ground in the other. This is because all colour transfer algorithms are sensitive to variations in the areas of the image occupied by the same colour, they risk overstretching the colour mappings and thus producing unbelievable renderings. To deal with this issue a simple solution (presented in [Rei01]) is to manually select swatches in both pictures and thus associate colour clusters corresponding to the same content. This is tantamount to performing manual image segmentation, and is simply impractical for a large variety of images, and certainly for sequences of images.

One automated solution is to invoke the spatial information of the images to constrain the colour mapping [Wel02, Ji04, Jia04]. In an extreme situation, colour from a coloured image may be required to be transferred to a grayscale image. Hence similarities between spatial neighbourhoods of the two pictures is then the only way to create a colour transfer operation automatically. This is a computationally demanding solution. Another automated solution is to restrict the variability on the colour mapping. For example in [Cha04], the pixels of both images are classified in a restricted set of basic colour

Original Colour Palette Resulting Transfer

Figure 3.1: Issues of the Continuous Assumption for Colour Transfer. The Campanile appears clearer than the sky in the original picture, and darker than the sky in the target picture. This contrast inversion can not be handled correctly by continuous transfer. Even if the result image has the same colour statistics than the target image, the transfer is clearly not the one wanted. Note that the white of the Campanile comes from the white of the target sky.

Original Colour Palette Reinhard Results

Figure 3.2: Example of Colour transfer using Reinhard [Rei01] Colour Transfer. The transfer fails to re-synthesise the colour scheme of the target image. To be successful the method would require human interaction.

categories, derived from psycho-physiological studies (red, blue, pink ...). The colour transfer ensures for instance that blue-ish pixels remain blue-ish pixels. This gives a more natural transformation. The disadvantage is that it limits the range of possible colour transfers.

Novelty. Using the distribution transfer method for the example-based recolouring problem can fix many of the shortcomings of previous efforts in this area. Firstly, it is computationally attractive as it uses just 1D pdf matching in an iterative scheme. Secondly, the method is completely non-parametric and is very effective at matching arbitrary colour distributions. Thirdly, the proposed method for reducing grain artefact results in high quality picture.

3.2 Reducing Grain Noise Artefacts

The colour mapping to the original picture transfers correctly the target colour palette to the original picture but it might also produce some grain artefacts as shown in figure 3.2 and 3.5. When the content differs, or when the dynamic range of both pictures are too different, the resulting mapping function can be stretched on some parts (see figure 3.2-e), and thus enhances the noise level (see figure 3.2-c). This can be understood by taking the simple example of a linear transformation t of the original picture I: $t(I) = a\ I + b$. The overall variance of the resulting picture is changed to $\text{var}\{t(I)\} = a^2\ \text{var}\{I\}$. This means that a greater stretching ($a > 1$) produces a greater noise.

The solution proposed here to reduce the grain is to run a post-processing algorithm that forces the level of noise to remain the same. The idea is to adjust the gradient field of the picture result so that it matches the gradient field of the original picture. If the gradient fields of both pictures are similar, the level of noise will be the same. Matching the gradient of a picture has been addressed in different computer vision applications like high dynamic range compression [Fat02]; the value of this idea has been thoroughly demonstrated by Pérez *et al.* in [Pér03]. The manipulation of gradient can be efficiently solved using a variational approach. The problem here is slightly different, since re-colouring also implies changing the contrast levels. Thus the new gradient field should only loosely match the original gradient field.

Denote $I(x, y)$ the 3-dimensional original colour picture. To simplify the coordinates are omitted in the expressions and I, J, ψ,ϕ, etc. actually refer to $I(x, y)$, $J(x, y)$, $\psi(x, y)$ and $\phi(x, y)$. Let $t : I \rightarrow t(I)$ be the colour transformation. The problem is to find a modified image J of the mapped picture $t(I)$ that minimises on the whole picture range Ω

$$\min_J \iint_\Omega \phi \cdot ||\nabla J - \nabla I||^2 + \psi \cdot ||J - t(I)||^2 \mathrm{d}x\mathrm{d}y \tag{3.1}$$

with Neumann boundaries condition $\nabla J|_{\partial\Omega} = \nabla I|_{\partial\Omega}$ so that the gradient of J matches with the gradient of I at the picture border $\partial\Omega$. The term $||\nabla J - \nabla I||^2$ forces the image

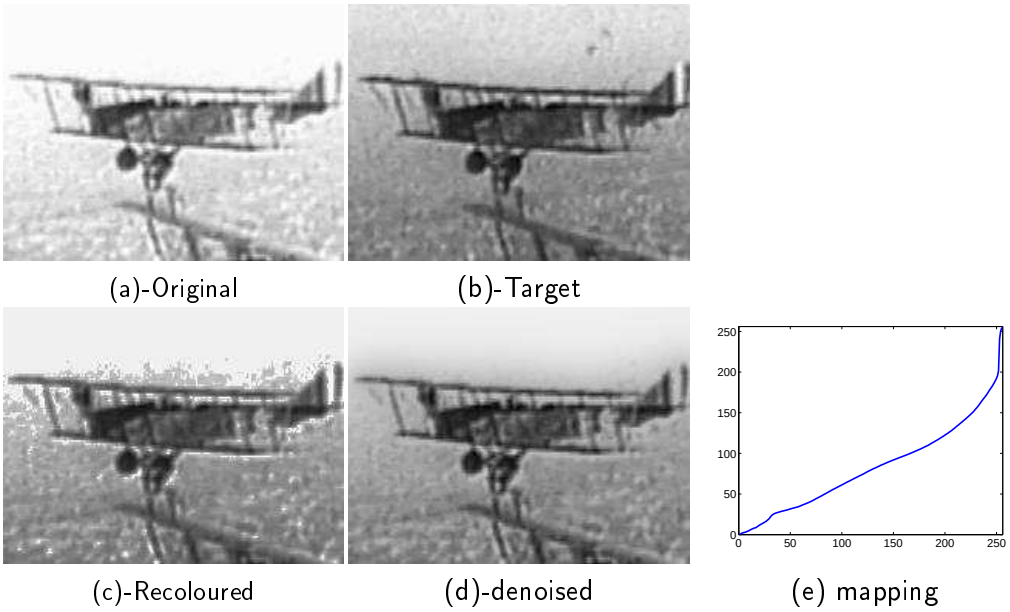


Figure 3.3: Result of grain reducing. The two consecutive archive frames (a) and (b) suffer from extreme brightness variation (this artefact is known as flicker [Pit04a]). The corresponding mapping transformation (e) is overstretched, which results in an increased level of noise on the mapped original frame (c). The proposed grain artefact reducer is able to reproduce the noise level of the original picture. The top of the original picture is saturated and cannot be retrieved but the algorithm succeeds in preserving the soft gradient.

gradient to be preserved. The term $||J - t(I)||^2$ ensures that the colours remain close to the target picture and thus protects the contrast changes. Without $||J-t(I)||^2$, a solution of equation (3.1) will be actually the original picture I.

The weight fields $\phi(x, y)$ and $\psi(x, y)$ affect the importance of both terms. Many choices are possible for ϕ and ψ, and the following study could be easily be changed, depending on the specifications of the problem.

The weight field ϕ has been here chosen to emphasise that only flat areas have to remain flat but that gradient can change at object borders:

$$\phi(x, y) = \frac{30}{1 + 10\ ||\nabla I||} \tag{3.2}$$

The weight field ψ accounts for the possible stretching of the transformation t. Where ∇t is big, the grain becomes more visible:

$$\psi(x, y) = \begin{cases} 2/\left(1 + ||(\nabla t)(I)||\right) & \text{if } ||\nabla I|| > 5 \\ ||\nabla I||/5 & \text{if } ||\nabla I|| \leq 5 \end{cases} \tag{3.3}$$

where $(\nabla t)(I)$ is the gradient of t for the colour I and thus refers to the colour stretching. The case $||\nabla I|| \leq 5$ is necessary to re-enforce that flat areas remains flat. While the gradient of t is easy to estimate for grayscale pictures, it might be more difficult to obtain for colour mappings. The field can then be changed into:

$$\psi(x, y) = \begin{cases} 1 & \text{if } ||\nabla I|| > 5 \\ ||\nabla I||/5 & \text{if } ||\nabla I|| \leq 5 \end{cases} \tag{3.4}$$

Numerical Solution. The minimisation problem in equation (3.1) can be solved using the variational principle which states that the integral must satisfy the Euler-Lagrange equation:

$$\frac{\partial F}{\partial J} - \frac{d}{dx}\frac{\partial F}{\partial J_x} - \frac{d}{dy}\frac{\partial F}{\partial J_y} = 0 \tag{3.5}$$

where

$$F(J, \nabla J) = \phi \cdot ||\nabla J - \nabla I||^2 + \psi \cdot ||J - t(I)||^2 \tag{3.6}$$

from which the following can be derived:

$$\phi \cdot J - \operatorname{div}\left(\psi \cdot \nabla J\right) = \phi \cdot t(I) - \operatorname{div}\left(\psi \cdot \nabla I\right) \tag{3.7}$$

This is an elliptic partial differential equation. The expression $\operatorname{div}\left(\psi \cdot \nabla I\right)$ at pixel $\mathbf{x} = (x, y)$ can be approximated using standard finite differences [Wei98] by:

$$\operatorname{div}\left(\psi \cdot \nabla I\right)(\mathbf{x}) \approx \sum_{\mathbf{x}_n \in \mathcal{N}_\mathbf{x}} \frac{\psi_{\mathbf{x}_n} + \psi_\mathbf{x}}{2} \left(I_{\mathbf{x}_n} - I_\mathbf{x}\right) \tag{3.8}$$

where $\mathcal{N}_{\mathbf{x}}$ corresponds to the four neighbouring pixels of $\mathbf{x}$. Using this in equation (3.7) yields a linear system as follows:

$$\begin{aligned} a_1(x,y)J(x,y-1) + a_2(x,y)J(x,y+1) + a_3(x,y)J(x-1,y) + a_4(x,y)J(x+1,y) \\ +a_5(x,y)J(x,y) = a_6(x,y) \end{aligned} \tag{3.9}$$

with

$$\begin{aligned} a_1(x,y) &= -\frac{\psi(x,y-1)+\psi(x,y)}{2} \\ a_2(x,y) &= -\frac{\psi(x,y+1)+\psi(x,y)}{2} \\ a_3(x,y) &= -\frac{\psi(x-1,y)+\psi(x,y)}{2} \\ a_4(x,y) &= -\frac{\psi(x+1,y)+\psi(x,y)}{2} \\ a_5(x,y) &= \frac{1}{2}\Big(4\psi(x,y)+\psi(x,y-1)+\psi(x,y+1)+\psi(x-1,y)+\psi(x+1,y)\Big)+\phi(x,y) \\ a_6(x,y) &= \frac{1}{2}\Big(\psi(x,y)+\psi(x,y-1))(I(x,y-1)-I(x,y)) \\ &\quad +(\psi(x,y)+\psi(x,y+1)(I(x,y+1)-I(x,y)) \\ &\quad +(\psi(x,y)+\psi(x-1,y)(I(x-1,y)-I(x,y)) \\ &\quad +(\psi(x,y)+\psi(x+1,y)(I(x+1,y)-I(x,y))\Big) \\ &\quad +\phi(x,y)I(x,y) \end{aligned}$$

The system can be solved by standard iterative methods like SOR, Gauss-Seidel with multigrid approach. Implementations of these numerical solvers are widely available and one can refer for instance to the Numerical Recipes [Pre92]. The main step of these methods is to solve iteratively for $J(x,y)$. Note that $J(x,y)$ and $a_i(x,y)$ are of dimension 3, but that each colour component can be treated independently. For instance, the iteration for the red component field is of the form

$$\begin{aligned} J_R^{(k+1)}(x,y) = \frac{1}{a_5^R(x,y)}\Big(a_5^R(x,y) - a_1^R(x,y)J_R^{(k)}(x,y-1) - a_2^R(x,y)J_R^{(k)}(x,y+1) \\ - a_3^R(x,y)J_R^{(k)}(x-1,y) - a_4^R(x,y)J_R^{(k)}(x+1,y)\Big) \end{aligned} \tag{3.10}$$

where $J_R^{(k)}(x,y)$ is the result in the red component at the k^{th} iteration.

3.3 Results

The remaining pages show some results from the proposed Colour Grading technique. In Figure 3.4 for instance, the original mountain picture in (a) is recoloured using the target colour scheme for the sea-side image in (b). The result of the transfer appears in the third

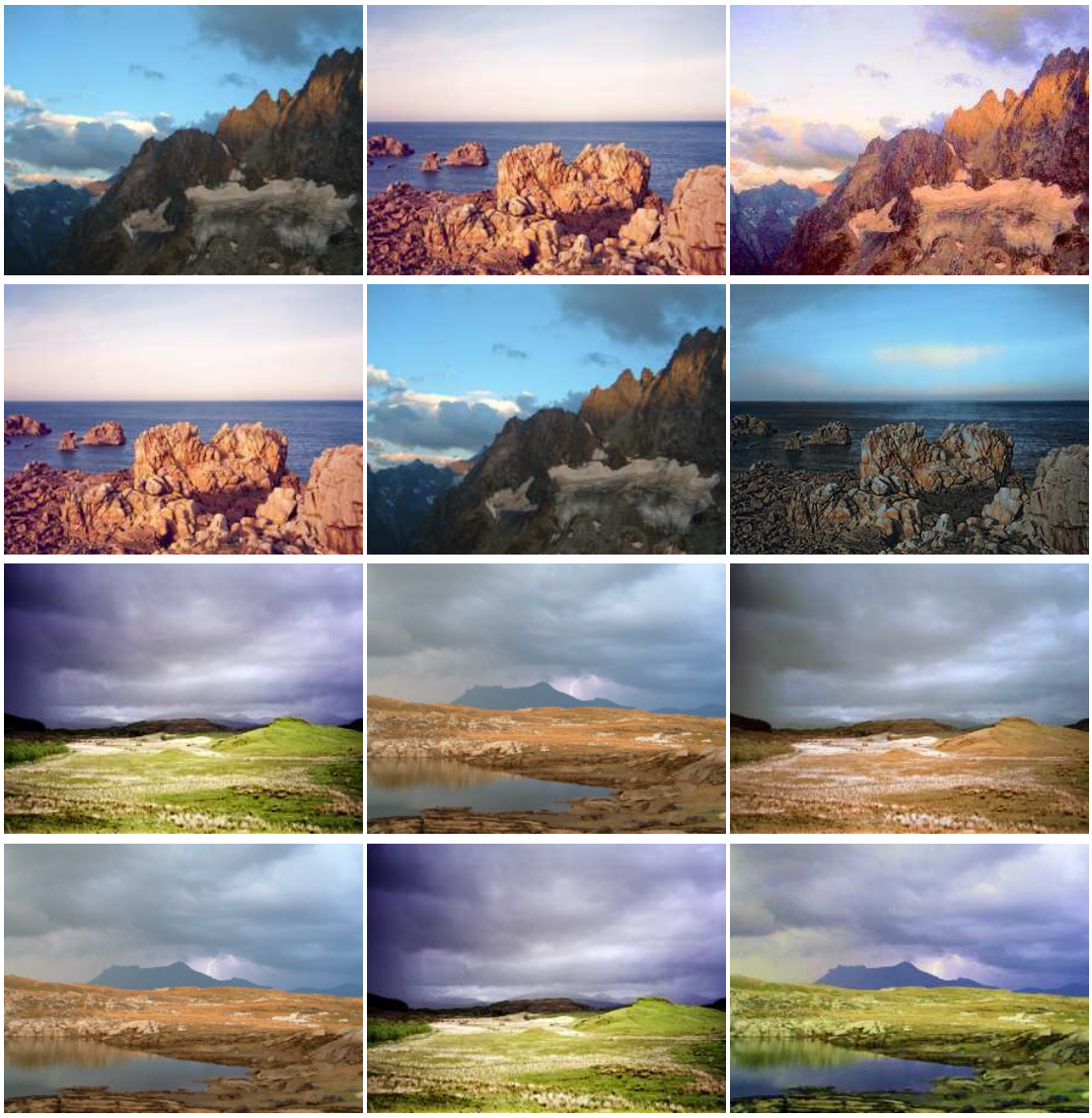
column.

Examples of Colour Grading for matching lighting conditions are presented in 3.6. On the first row, the colour properties of the sunset are used to synthesise the 'evening' scene depicted at sunset. On the second row, the colour grading allows correction of the change of lighting conditions induced by clouds. Even when using the grain artefact reducer, an unavoidable limitation of colour grading is the clipping of the colour data: saturated areas cannot be retrieved (for instance the sky on the golf image cannot be recovered). A general rule is to match pictures from higher to lower range dynamics.

The figure 3.7 displays examples of colour restoration of faded movies. The idea is similar to colour grading as the idea is to recreate different atmospheres. The target pictures used for recreating the atmosphere are on the second row.

3.4 Conclusion

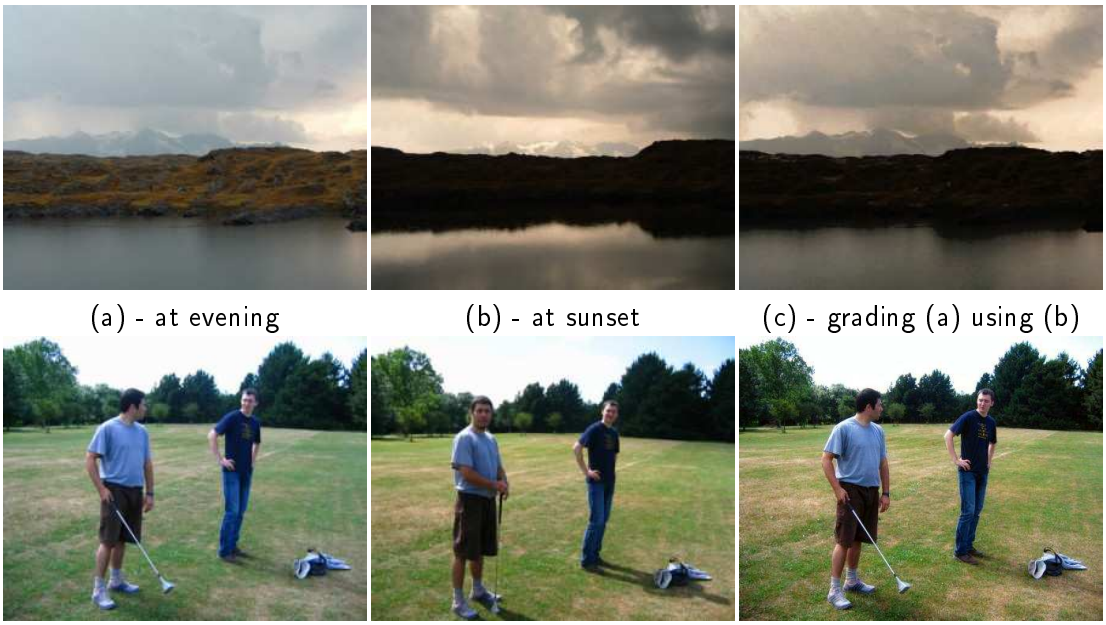
This chapter has proposed an original technique for colour grading. The technique is based on an exact transfer of colour pdf of the target picture. The possible grain artefacts are removed in a second step. The overall technique is simple, easily to implement and works for a large variety of scenarios, even when the example picture is very different from the processed images.



(a) - original image (b) - target palette (c) - (a) recoloured with (b)

Figure 3.4: Examples of Image re-Colouring: the original image (a) is re-coloured using the palette of the example image (b). The result is displayed on (c).

Figure 3.5: Artefact grain reducing for colour picture. See how the details of the picture are preserved, while the spurious graininess in the sky is washed out.



(a) - at evening (b) - at sunset (c) - grading (a) using (b)

(a) - with clouds (b) - without clouds (c) - grading (a) using (b)

Figure 3.6: Examples of Colour Grading for matching lighting conditions. On the first row, the colour properties of the sunset are used to synthesise the 'evening' scene depicted at sunset. On the second row, the colour grading allows to correct the change of lighting conditions induced by clouds.

Original Frame 70's atmosphere pub atmosphere

Original Frame Original 70's atmosphere Original pub atmosphere

Figure 3.7: Example of Colour Grading for Image and Video Restoration. It is possible to recreate different atmospheres. Here an old faded film is transformed to match the colour scheme of a movie from the 70's and a pub ambiance.

Chapter 4

Stabilisation of Brightness Fluctuations in Image Sequences

WHILE in re-Colouring images the target colour palette is known, there exists a number of scenarios where the target palette is a priori unknown. Consider an image sequence, where colour and brightness levels fluctuate for each frame of the sequence. The problem is to align these levels in order to remove the fluctuations. The underlying ground-truth target palette is unknown and the problem comes down in first place to estimate the target palette. Most frequently fluctuations also present some spatial correlation, which need also to be accounted for. This chapter proposes a solution to this problem by extending the ideas developed in the previous chapters to include both temporal and spatial considerations.

Instances of Brightness Fluctuations. Random fluctuation in the observed brightness of recorded image sequences, also called *flicker*, occur in a variety of situations. The most commonly consumer observed instance of flicker is in archived film and video (see figure 4.1 and 4.2, see also examples of videos in [Pit02]). It is caused by the degradation of the medium (ageing of the film stock), varying exposure times, or curious effects of poor standards conversion. Varying exposure time is common to hand-cranked footages, but happens too with mechanised cameras in early films or most recently with personal 8mm. Remarkably flicker often affects modern film and video media if the lighting conditions are poor, as in submarines surveillance, or when the transfer from film to video (telecine) is not properly done.

The flicker artefact is still an actual issue, even with the use of digital cameras. Consider

parts of this chapter have been published in [Kok03, Pit03, Pit04a]

for instance the popular Inbetweening special effect used in "The Matrix" (1999). This special effect is based on the interpolation of images coming from multiple cameras shouting with different angle the same scene. Unfortunately, the brightness levels of the pictures are sometimes misaligned. This is due to different camera behaviour or lighting conditions due to camera orientation, even if video cameras have been previously calibrated using radiometric calibration routines. Figure 4.3 shows this radiometric calibration issue for an outdoor footage.

Consequences. The presence of flicker is often praised by film enthusiasts as an essential aesthetic element of the film experience. Unfortunately people with medical conditions or simply high sensitivity to flickering can suffer eye strain and even headaches while watching such movies. This results of course in much discomfort and pain.

From an image processing point of view, the presence of flicker has also a detrimental effect on many applications since it breaks the common assumption that object brightness is constant over the frames (this is the so-called "brightness constancy"). For instance, motion estimation [Lai99, Jin01] or any other image matching algorithms fail in presence of flicker as they try to match brightness levels of two pictures.

Brightness fluctuations have also dramatic consequences on the video compressibility. The flicker reduces the redundancy between frames and hence increases the bandwidth of transmitted sequences. This is particularly a problem for Digital Television or broadcasting of video content over the Internet.

Organisation of the Chapter. This chapter proposes therefore to stabilise these fluctuations in image sequences. Dealing with the problem of flicker requires some attempt to model or measure the fluctuation between frames (the estimation process), and then to remove this fluctuation in some way (the correction process). The core of the model and its estimation process rely on the re-colouring technique exposed in the previous chapter. After an exposition of related works in the area, the following sections examine how to integrate spatial and temporal considerations to distribution transfer techniques, in order to establish a generic flicker removal framework. In addition, this chapter proposes a novel implementation based on general purpose PC graphics hardware. The results show that it is possible to cope with a wide range of flicker effect. The section 4.5 presents results showing the effect of flicker on MPEG4 compression of different kinds of sequences including multi-view camera sequences, and the improvement in bandwidth usage with the proposed flicker reduction method.

4.1 Related works

The following paragraphs present several approaches that have been proposed in the literature—most of them independently—for the removal of flicker in videos. The approaches can be categorised by two criteria, the linearity of the intensity mapping (linear

Figure 4.1: Example of flicker manifestation on two consecutive frames of *Rory O'More*, 1911. Note in particular the black diagonal on the right frame.

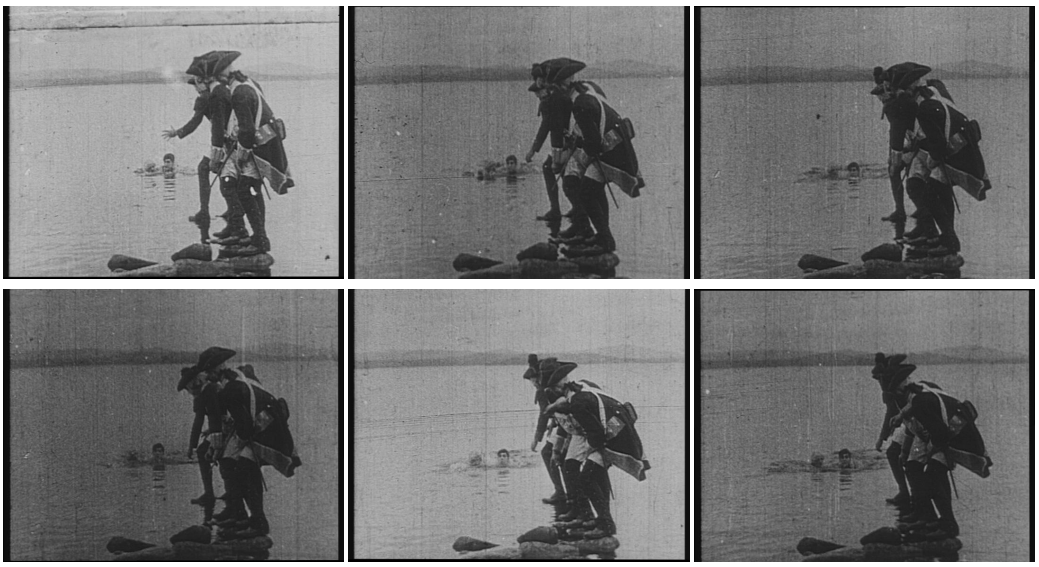


Figure 4.2: Example of flicker manifestation on image sequence of *Rory O'More*.

or non-linear), and the spatial considerations (with modelling of spatial variations or without). The restoration scheme of the methods is based on the recovery of the unknown original intensity image (ground truth) I_n^o given the observed intensity images I_n at time n. The estimation $\hat{\mathrm{I}}_n^o$ of the non-degraded image gives the restored frame I_n^R.

Linear model. The linear model proposed by Decencière [Dec97] link the intensities of the observed image I_n to the original flicker free image I_n^o. It involves a gain a_n and an offset b_n to create a degradation model as follows:

$$\mathrm{I}_n(x,y) = a_n \; \mathrm{I}_n^o(x,y) + b_n \tag{4.1}$$

where $\mathrm{I_n}(x,y)$ is the intensity of picture n at pixel (x,y). While processing the video, the image $\mathrm{I}_n^o(x,y)$ is not yet available at time n and is replaced in equation (4.1) by the previous estimated one $\hat{\mathrm{I}}_{n-1}^o$. The maximum and minimum of the intensity histograms of the I_n and $\hat{\mathrm{I}}_{n-1}^o$ are matched to estimate the gain and the offset, assuming that those extrema are constant in the flicker free sequence.

One major weakness of this formulation is that the estimation of the current flicker free image relies on the availability of a flicker free instance of the previous image. A reference frame has therefore to be chosen by an operator to start the restoring process. Furthermore, the success of the restoration is then sensitive to any possible estimation-restoration error occurring any time from the start.

Linear model with spatial dependencies. As illustrated in figure 4.1, the flicker is not the same across an entire frame. Roosmalen [Roo99a, Roo99b] introduced thus spatial dependency (x,y) as follows:

$$\mathrm{I}_n^o(x,y) = a_n(x,y) \; \mathrm{I}_n(x,y) + b_n(x,y) \tag{4.2}$$

The original image $\mathrm{I}_n^o(x,y)$ is replaced in the equation by the previous restored frame. The spatial variation is introduced on a block basis, thus a_n, b_n are piecewise constant. Parameters can be estimated by least squares estimation [Yan00], and each estimate is associated with a confidence measure [Roo99a].

The problem with direct parameter estimation for flicker is that not all of each image pair can be matched. The parts that cannot be matched are due to occlusions/uncovering because of motion or simply due to missing data (blotches/dirt/dropout) in the case of degraded film and video material. To cope with this problem, Roosmalen [Roo99a] and Yang [Yan00] suggest to detect occluding areas based on spotting large intensity differences that cannot be explained by flicker alone. Parameter estimation is then performed only on the blocks in which there are no outliers detected. Estimates for the "missing blocks" are then generated by some suitable interpolation algorithm. Unfortunately, this method for detecting outliers fails in the presence of heavy flicker degradation.

Because the restored images are generated by *locking* the brightness to previous frames, errors can accumulate. To avoid this problem, Roosmalen [Roo99b] proposes to construct the restored intensity image as a mixture of $\hat{\mathrm{I}}_n^o$ and the observed image:

$$\mathrm{I}_n^R(x,y) = k\ \hat{\mathrm{I}}_n^o(x,y) + (1-k)\ \mathrm{I}_n(x,y) \tag{4.3}$$

where k is the forgetting factor set experimentally at 0.85.

Linear Model with Robust Regression. Ohuchi [Ohu00] and also Lai [Lai99] consider the same linear model as in equation (4.2) but the spatial variation of the gain and the offset is expressed directly using second order polynomials [Lai99, Ohu00]. The parameters of those polynomials are then estimated using robust M-estimation [Lai99, Ohu00, Hub81, Pre92] involving a Reweighted Least Squares algorithm. Robustness of the estimator is needed to deal with outliers that frequently occur in old videos.

However, as noticed by Kokaram et al.[Kok03], due to the correlation between the frames, the regression (robust and non-robust) introduces a bias in the estimates that can damage seriously the restoration process in case of heavily degraded sequences. Therefore, the authors [Kok03] have introduced a slightly modified linear model that allows reduction of this bias. Another important improvement proposed in [Kok03] is to change the polynomial basis to a cosine basis to express the gain and the offset. Since the success of global motion estimation is linked to flicker correction (and vice versa), some have proposed to couple both estimations [Lai99, Roo99b, Kok03]. Thus in the technique proposed in this chapter, images will be registered prior to any flicker estimation.

Non-linear model. The brightness distortion can be sometimes non-linear. To account for this non-linearity, Ridcharson and Suter [Ric95] propose to individually stretch each frame by applying a grayscale equalisation, and then linearly remap the intensities levels onto a constant smaller dynamic range. Instead of linearly remap the intensities on a fixed dynamic range, Naranjo et al. [Nar00] propose to use directly the simple 1-dimensional pdf transfer scheme as in equation (2.11). The target intensity pdf is obtained by averaging the intensity pdfs of neighbouring frames [Nar00]. The same idea of using intensity pdf transfer has been actually previously developed by Shallauer *et al.* in [Sch99, sec. 6], with also, a spatial modelling by performing grayscale pdf transfer on a block basis. In their case however, the target intensity pdf is obtained from a selected reference frame.

Some attends have been done to use parametric model. For example Skoneczny [Sko01] proposes to model the brightness distortion as a linear combination over six basis function: $t(\mathrm{I}) = \sum_{i=1}^{6} c_i\ g_i(\mathrm{I})$. The parameters c_i are estimated by standard least squares regression over the two considered frames.

Only recently, Vlachos [Vla04] gave a first attempt to derive from first principles a model for the non-linearity of the flicker distortion. In the paper, the Hurter-Driffeld *Density versus log-Exposure* characteristic is used to formulate, a quadratic model for the

non-linear correction of the intensities.

The use of the Hurter-Driffeld curve is not without reminding the Radiometric Alignment techniques.

Relation to Radiometric Self-Alignment. Flicker has indeed tight links with the problem of radiometric alignment, and in some scenarios, the radiometric alignment problem can even be seen as a special case of flicker.

Consider for example, the problem of aligning the levels of an image sequence where every frame is taken with the same camera but with different exposure time. Ideally the camera would be a linear imaging system, and the image intensity I on the picture would be related to the exposure time Δt and the irradiance E of the scene by the simple relation $\mathrm{I} = E\Delta t$. Unfortunately a camera is rarely a linear system. For photographic film for instance, the response of the film stock to variations in exposure is summarised by a characteristic curve (this is the so-called Hurter-Driffield curve). More generally, with digital or standard cameras, or after any imaging system involving development, scanning and digitisation process,*etc.* , it is possible to establish a relationship between the image irradiance of the scene, the exposure time, and the image intensity given by the camera. The relationship can be stated by introducing the radiometric response function as follows:

$$\mathrm{I} = f(E\Delta t) \tag{4.4}$$

As shown on figure 4.4, the response functions of commercial films found in the market stocks are usually not linear. Thus the problem comes down to the estimation of the radiometric response function f and/or the exposure time, if this parameter is not accessible.

Devebek and Malik [Deb97] establish clearly this problem and solve the estimation the radiometric response f and the irradiance map by least square minimisation. It is then claimed in [Mit99] that the estimation can be made more robust to noisy pictures by modelling the response curve as a sum of polynomials. Both approaches suppose that the pictures correspond to the same scene view, which eases greatly the estimation since there is no need to register the pictures beforehand.

The problem of radiometric auto-calibration can be extended to videos [Man02, Kan03, Kim04]. As seen previously, working with videos inevitably makes the estimation difficult because of the the presence of outliers due to occlusions and other common image registration issues. Note that the response function only infers the transfer between the scene irradiance and the resulting image brightness on the film. Hence it does not refer directly to the possible brightness transfer *between* frames. Interestingly Mann [Man02] proposes to consider the joint distribution of two consecutive image intensities $p(\mathrm{I}_n, \mathrm{I}_{n+1})$. The estimation of the radiometric response then involves a marginalisation of the joint probability into $p(\mathrm{I}_n)$ and $p(\mathrm{I}_{n+1})$. This is tantamount to the kind of idea developed for flicker removal by Shallauer *et al.* in [Sch99, sec. 6].

It is thus clear that the problem of radiometric alignment relates to flicker. In particular,

it properly solves the special case where the source of flickering is only due to globally varying exposure. This is however a special case of flicker, and it is well understood in the general case, that flicker may presents spatial correlations. Moreover, it is assumed in radiometric alignment that the response function is constant for the whole sequence. This makes sense for modern footages, but becomes unclear for old sequences, where chemical reactions damaged the physical film.

4.2 Spatial and Temporal Integrations

Experience with this problem shows that, spatial dependence and non-linearity are key to modelling a wide range of flicker defects. Since the causes of flicker are usually unknown and various, it is preferable to adopt a very weak prior on the distortion function.

Spatial Integration . It is thus proposed as in [Sch99] to extend the non-linear model proposed by Naranjo [Nar00] to integrate spatial variations

$$\mathrm{I}_n^o(\mathbf{x}) = t_n\left(\mathrm{I}_n(\mathbf{x}), \mathbf{x}\right) \tag{4.5}$$

where $\mathbf{x} = (x, y)$ is the pixel location, I^o the flicker free frame, and I the the observed frame. Ideally, an estimate of a transformation should be performed for each pixel $\mathbf{x}$. However, this is computationally expensive and it would require an additive spatial smoothness constraint. Alternatively, the transformations t_n^i is estimated at regularly spaced control points $\mathbf{x}_i$ in the image using interpolating splines of order 3. The splines yield an implicit smoothness constraint and the corrected pixel at $\mathbf{x}$ can be written

$$\mathrm{I}_n^o(\mathbf{x}) = \sum_{i=1}^{N} w(\mathbf{x} - \mathbf{x}_i)\; t_n^i\left(\mathrm{I}_n(\mathbf{x})\right) \tag{4.6}$$

where t_n^i is the transformation at control point $\mathbf{x}_i$ and $w(\mathbf{x})$ the interpolating 2D mask. In dimension 1, the interpolating kernels of splines of order 3 are defined as follows:

$$w(\mathbf{x} - \mathbf{x}_i) = s\left(\frac{x - x_i}{\sigma_w}\right) s\left(\frac{y - y_i}{\sigma_h}\right) \tag{4.7}$$

$$s(a) = \begin{cases} -a^2 + \frac{3}{4} & \text{if } a \leq \frac{1}{2} \\ a^2 - \frac{3}{2}a + \frac{9}{8} & \text{if } \frac{1}{2} \leq a \leq \frac{3}{2} \\ 0 & \text{if } \frac{3}{2} \leq a \end{cases} \tag{4.8}$$

where σ_w and σ_h correspond to the distances between two horizontal and vertical neighbouring control points. The number of control points on one axis will be referred to as the *flicker order*.

Figure 4.3: Example of Radiometric Calibration issue on a outdoor multi-camera footage. The image on the right is darker which results in a shift to the left in the intensity histogram (dashed green plot).

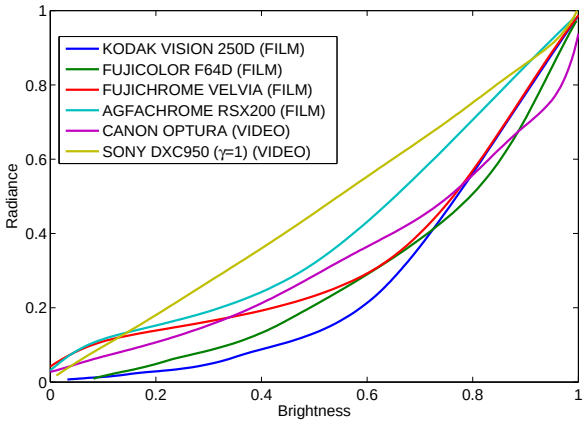


Figure 4.4: Normalised response functions of a few popular films and video cameras. (extracted from [Mit99])

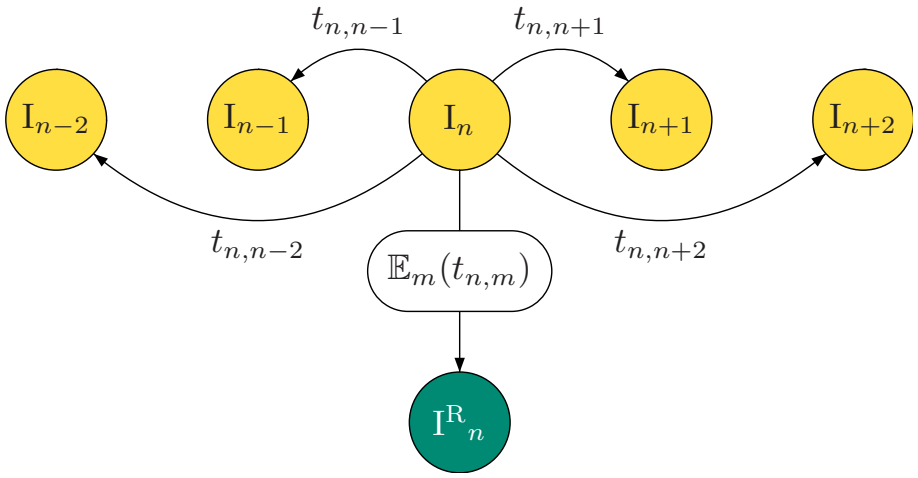


Figure 4.5: Compensation of a frame: $\{\mathrm{I}_n\}_n$ original pictures, I_n^R: restoration of I_n, $t_{n,m}$ brightness variation parameters between frames I_n and I_m.

Temporal Integration. A key deviation from previous efforts in flicker removal is that no selected key frame is supposed to be available. Thus, the model considers only the change in brightness between observed images and not the brightness change between the observed and the hidden clean images:

$$\mathrm{I}_m(\mathbf{x}) = \sum_{i=1}^{N} w(\mathbf{x} - \mathbf{x}_i)\; t^i_{n,m}\left(\mathrm{I}_n(\mathbf{x})\right) \tag{4.9}$$

where $t_{n,m}$ is the mapping from frame n to n. As illustrated in figure 4.5, the task is therefore to smooth brightness changes between frames and not necessarily reveal the true original image. Brightness variations between images I_m and I_n can be caused by: 1) intentional effects like shadows or gradual brightness changes, due to special editing effects for instance, and 2) the flicker degradation which is unintentional. The first effect is generally low frequency and exhibits slow temporal variation. The second effect is temporally impulsive and it is this signal that has to be removed.

Consider now that the mapping $t_{n,m}$ have been estimated. The estimated parameters for $t_{n,m}$ correspond to the impulsive flicker mixed with the gradual informative variations. To separate the impulsive flicker and keep the informative variation, the parameters are filtered using a temporal robust expectation:

$$\hat{t}_n(k) = \mathbb{E}_m\left[t_{n,m}(k)\right] = \arg\min_{t_n(k)} \sum_{m=n-M}^{n+M} e^{-\frac{(m-n)^2}{\sigma_w}} \rho(t_{n,m}(k) - t_n(k)) \tag{4.10}$$

where $t_{n,m}(k)$ is the k^{th} component of the look-up table $t_{n,m}$, ρ a robust function [Pre92] and σ_w a temporal scale factor. The temporal window has been fixed experimentally to 15 frames ($M = 7$). The expectation $\hat{t}_n$ is estimated using Iterative Reweighted Least Squares

(IRLS) and finally applied to I_n to generate the restored image I_n^R. The IRLS involves the following update step:

$$\hat{t}_n(k) \leftarrow \frac{\sum_m t_{n,m}(k) e^{-\frac{(m-n)^2}{\sigma_w}} \psi(t_{n,m}(k) - \hat{t}_n(k))}{\sum_m e^{-\frac{(m-n)^2}{\sigma_w}} \psi(t_{n,m}(k) - \hat{t}_n(k))} \tag{4.11}$$

The weight function chosen is the Geman one, which gives for ψ:

$$\psi(u) = \frac{1}{(1+u^2)^2} \tag{4.12}$$

Speeding Up the Correction Scheme. It is possible to dramatically reduce the complexity by estimating $t_{n,m}$ through the simple combination of successive estimations $t_{n,n+1}$. However, when the flicker is too severe that only a portion of the intensity range is occupied by an image (skewed pdf) the parameter estimates are poor. To conclude, two strategies are available: one involving exhaustive estimations but able to cope with extreme fluctuations in intensity without propagation of errors and one involving a minimal number of estimations but more liable to errors.

4.3 Estimation of the Mapping

In the case of a spatially constant brightness distortion ($\mathrm{I}_m = t_{n,m}(\mathrm{I}_n)$), it has been shown in previous chapters that the distortion function can be estimated by

$$t_{n,m}(\mathrm{I}_n) = F_m^{-1} \circ F_n(\mathrm{I}_n) \tag{4.13}$$

where F_n and F_m are the cumulative pdf of I_n and I_m. In this case the mapping is assumed to be locally constant. At each control point $\mathbf{x}^i$, the local pdf is given by[1]

$$p_n^i(u) = \frac{\sum_{\mathbf{x}|\mathrm{I}_n(\mathbf{x})=u} w(\mathbf{x}-\mathbf{x}_i)}{\sum_{\mathbf{x}} w(\mathbf{x}-\mathbf{x}_i)} = \sum_{\mathbf{x}|\mathrm{I}_n(\mathbf{x})=u} w(\mathbf{x}-\mathbf{x}_i) \tag{4.14}$$

where $\mathbf{x} \to w(\mathbf{x})$ is the spatial weighting function and $\mathbf{x}_i$ a control point. $\hat{t}_{n,m}^i$ is estimated using equation (4.13).

An interesting point about considering pdfs separately on each frames is that pdfs are not affected by local motion within the area of consideration, whereas regression methods assume that there is no local motion between both frames. Moreover, gradual changes in the scene, like appearance of new objects, are naturally handled by the filtering process. Nevertheless, for impulsive events, like large blotches, this estimator becomes very sensitive, and more robust methods need to be used. The next chapter proposes new estimation techniques to cope with severe levels of occlusion.

[1] where $\sum_{\mathbf{x}} w(\mathbf{x}) = 1$ for splines.

4.3.1 Flicker Compensation

Once the parameters of $\hat{t}_n$ have been estimated, the only remaining step is to apply the corresponding mapping to the current frame. This is a straightforward operation, but as mentioned in the re-colouring chapter, the stretching in mapping can produces very grainy pictures. Thus applying the grain reducer, proposed in the previous chapter, can then again be necessary. Moreover, this grain reducer post-process can also limit the impact of errors in the estimation.

Flicker Compensation using Graphics Hardware. Computational load can be a key issue in a restoration process involving thousands of frames. Flicker compensation is the most time consuming stage of the flicker restoration process. It has been found that in its simplest form (*i.e.* without grain reducer) the flicker compensation accounted on average for 80% of the total restoration time.

Modern computer graphics card are becoming much more programmable and contain powerful graphic processing units (GPUs). In fact they can now be considered as useful co-processors to the CPU. Flicker compensation on the GPU is only possible because of the latest advances in graphics hardware architecture especially full support for floating point accuracy. Using fragment programs it is possible to perform the necessary operations to map t_n^i on an image block and multiply it by $w(\mathbf{x} - \mathbf{x}_i)$. The vertex programs can be used to correctly position the interpolating kernels on the image. Render-to-texture and floating point data are required for the summation of $t_n^i(\mathrm{I}(\mathbf{x}))\ w(\mathbf{x} - \mathbf{x}_i)$.

The postprocessing grain reducer has not been yet implemented on a GPU, but it is well known that all the fundamental operations required for Gauss-Seidel Multigrid optimisation can be performed on the GPU [Kru03, Bol03]. A process description has been previously presented in [Goo03]. That module could be used with minimal change for this application.

Performance. Figure 4.6 shows the results of using the GPU compared to the CPU. These results were obtained on a 2.6GHz Pentium 4 machine running Windows XP, with an Nvidia GeForce FX5600 graphics card. The GPU implementation reduced the time taken for flicker compensation from 80% of the total restoration time to 55%. On average the GPU implementation is 3.5 times faster than the CPU implementation.

Altogether the full non-robust scheme processes on average 18 frames per second at flicker order 3. The full robust scheme takes 1.5 frames per second whereas Roosmalen's process takes around 2 to 3 seconds a frame on similar hardware. For heavily degraded sequences the full process can take up to 20 seconds per frame.

4.4 Practical Issues

Figure 4.7 shows the overall structure of the flicker correction system. The flicker order (i.e.

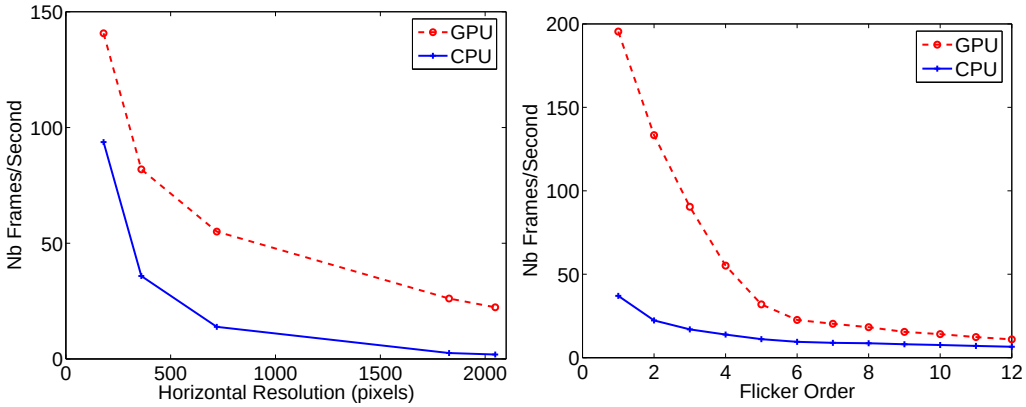


Figure 4.6: Flicker Compensation Speed CPU vs GPU. On the left with a fixed flicker order of 3 and varying frame size. On the right, with a fixed frame size of 720x576, but with varying flicker order.

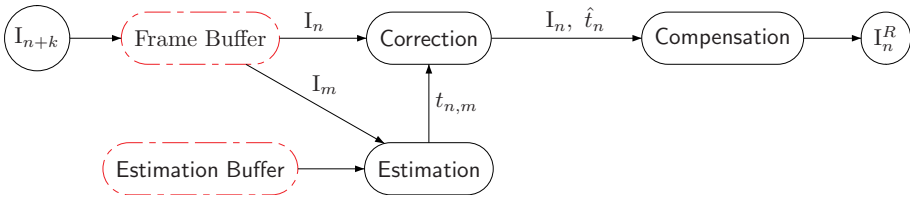


Figure 4.7: Flicker Correction Scheme.

the number of control points per dimension) typically vary from 3—for the vast majority of image sequences—to 6 or even 14 for old movies whose film has chemically changed. For flicker orders greater than 6, occlusions (blotches and local motions) requires to use the robust scheme presented in the next chapter.

Working on orders higher than 8 is not necessarily interesting since the flicker removal is then similar to a noise reducer processing. In particular, it is then difficult to estimate proper pdfs since the number of samples is too small. It is also important while working on very small areas to have a very fine estimation of the motion field. The noise remover techniques are specially designed to take these considerations into account. Thus it should be kept in mind that flicker removal should only be used for global fluctuations.

Most of the time working on colour sequences can be done by simply stabilising the luminance component of the frames. In some rare cases however, the colour channels also suffer from flicker fluctuations. In these cases, the colour mapping of the re-colouring chapter can be used. This has in practice no consequences on the framework.

4.5 Results

Sequences. The first sequence, *Snake* (see figure 4.3), composed of images from different uncalibrated cameras. This is a modern sequence actually used for post production.The second one, *Paula*, is a 8mm movie from a private repository, captured by pointing a DV camera at the projection screen; the asynchrony of the frame rates is the principle source of flicker. The third sequence is the *Tunnel* sequence [Roo99b], which was obtained by telecine. The last sequences comes from *Rory O'More*, a severely degraded movie from 1911.

Results. The results of the proposed flicker removal are presented on a separate page on figure 4.9. To have a better understanding of the temporal filter, the idea proposed here is to represent the image sequences using video slices. The frames are averaged along the horizontal axis, and thus only represented as one vertical line. This representation, displayed on figure 4.9, is particularly suited for assessing flicker, since it is then possible to detect

Evaluation by comparing the mean and the variance. Numerically assessing the performance of the systems on real degraded sequences is difficult because of the lack of objective criteria for assessing the quality of the restoration. However, as shown in [Roo99b, Ohu00, Nar00, Kok03] it is feasible to expect that a good de-flicker process would reduce the fluctuations in the mean and the variance of image intensities from frame to frame. Figure 4.8-b shows some results on the *Paula* and *Snake* sequences for the original sequence (ob), after a Naranjo de-flicker (na), after a spatially varying linear model (af), and after the de-flicker method proposed in this chapter (pm). It is visible that the filter smoothed the brightness fluctuations. Figure 4.8-a shows the importance of the non-linear treatment. Whereas (af) cannot remove completely the fluctuations, (pm) stabilises the brightness. However (na) hardly corrects the flicker.

Evaluation by comparing the MPEG4 compression performances. However the mean and the variance cannot characterise subtle differences between restorations, especially if the dirts and blotches make the mean and variance fluctuate. Hence the idea is to propose a novel criteria for assessing the quality of flicker reduced, by comparing the compression ratio given by a MPEG4 encoder (in this case the Microsoft MPEG4 encoder). Results on the *Tunnel* sequence corroborate the previous remarks : (pm) improves the compression performances by 48.6%, (af) by 45.8% and (na) by 38.4%.

As for the mean and the variance measure, this evaluation is still biased because it favours restoration processes that reduce details level, and the human based visual assessment still remains the best judge. It is however interesting to know that reducing flicker improves significantly compression performances. In particular for broadcasting over Internet, where bandwidth issues are critical.

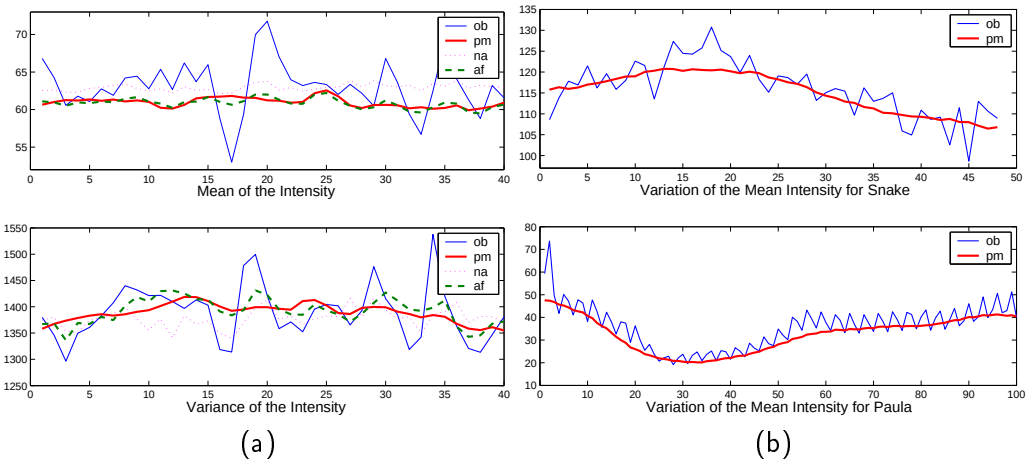


Figure 4.8: Comparison for different scenarios of the mean and variance (a) (*Tunnel*). Variations of the mean (b)(top: *Snake*, bottom: *Paula*).

Table 4.1: Compressions ratios between the original compressed sequences and the restored ones. Better restorations are obtained for higher compressions ratios.

Sequence	compression
Rory (shot 11)	17%
Rory (shot 13)	12.4%
Rory (shot 16)	15.2%
Rory (shot 19)	4.3%
Tunnel	48.6%
Paula	2%

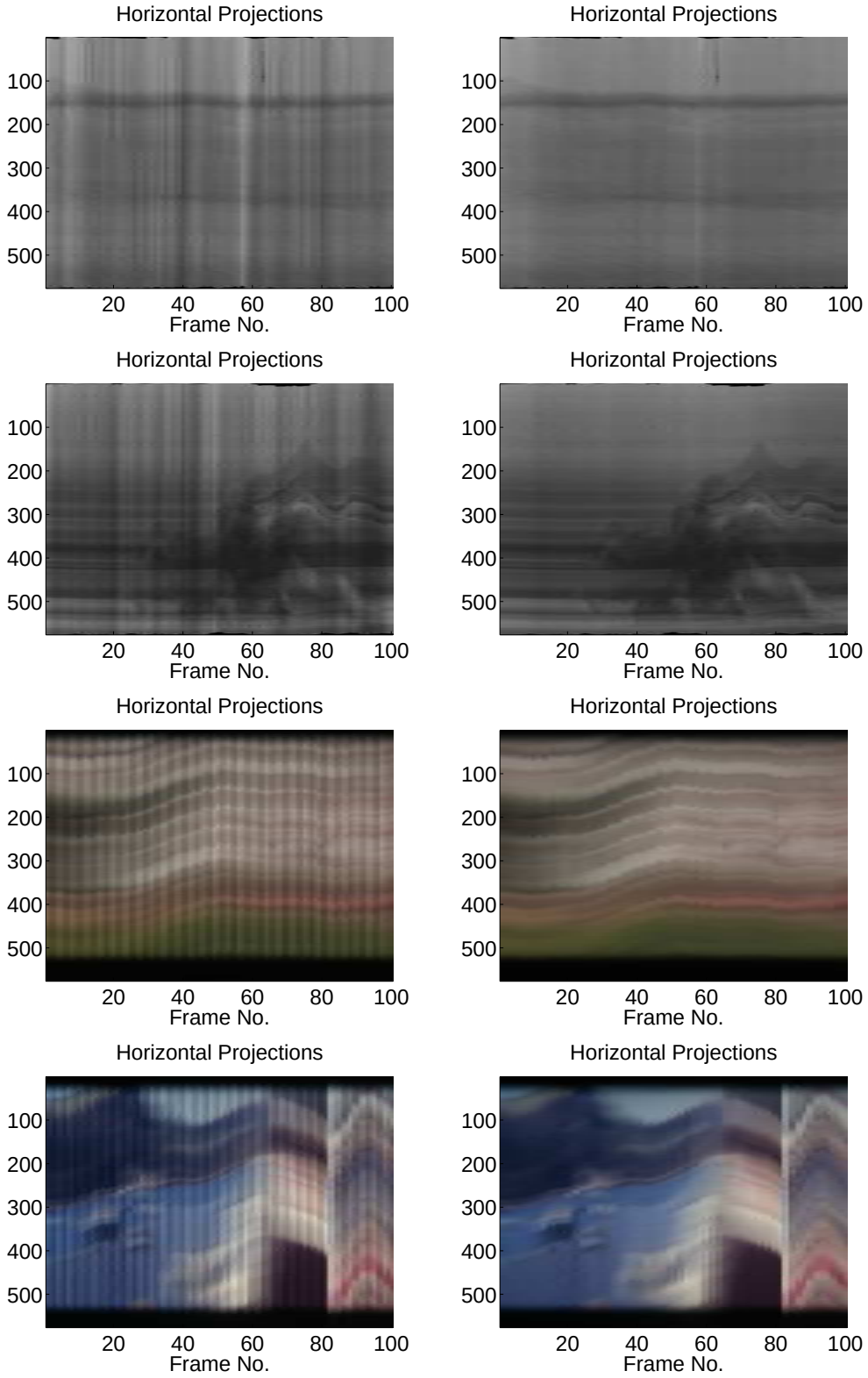


Figure 4.9: Flicker Removal Results. Each frame is represented by its horizontal projection on the vertical axis. The results on the right clearly show the brightness stabilisation of the sequences.

4.6 Conclusion

Images sequences can be affected by brightness fluctuation for many reasons. This chapter presents a new restoration process able to deal with various kinds of flicker. The algorithm is fast and coupled with the use of cheap graphics hardware, it is possible can reach near real-time performance on standard computers for most real image sequences.

Chapter 5

Robust One-Dimensional PDF Transfer

In many real situations, outliers can substantially affect the performance of the pdf transfer technique. The notion of outliers refers here to the data samples that do not follow the mapping, by opposition to *inliers* that do follow the mapping. In archive footage for example, outliers can be missing data randomly in each frame. If the size of the missing patches are large (which is frequently the case) this would bias the measurement of the image pdf away from the true underlying signal. In the colour transfer case, it is clear that large differences in content can affect the suitability of the mapping generated. Content mismatches can be considered as outlier contamination in general and that applied also when there is motion between frames in the case of flicker: the motion can cause content changes where it is not expected. An example of this problem is shown in figure 5.4. Of course it is key to establish exactly what are the outliers and inliers in the case of pdf transfer. This chapter therefore considers techniques for handling outliers in the transfer problem. Results are shown for both the flicker and colour transfer applications.

The treatment of outliers in Computer Vision and Image Processing applications is in fact a well-acknowledged key issue. It is both mandatory and also very difficult since outliers can be very numerous, with commonly more than 50% outliers in the data. The outliers can be detected and discarded by 1) modelling the inliers and considering the outliers as being non-inliers, and 2) by modelling directly the outliers. Combining both models allows to decides whether a sample is an outlier or not.

Modelling the inlier class is typically done by expressing the mapping as a linear com-

some ideas in this chapter are based on the work published in [Pit04a]

bination of basic functions:

$$t(u) = \sum_{i}^{d} a_i \phi_i(u) \tag{5.1}$$

The parameters of the model can be estimated by robust regression techniques [Hub81, Pre92]. The simplest model is to consider an affine mapping [Lai99, Ohu00, Kok03], but more complex model can be introduced [Sko01, Vla04]. There is a trade-off between the robustness of the estimation and the complexity of the model. Complex models can better fit to the actual data but can also fit to the outliers and thus make the estimation more difficult since the model does not discriminate between outliers and inliers. The approach adopted in this thesis is to explore these difficult cases where no strong prior on the mapping can help detecting the outliers.

Usually, robust regression techniques are based on the spatial registration of the pixels between frames (*e.g.* the pixel $\mathrm{I}_n(\mathbf{x})$ in frame n corresponds to the pixel $\mathrm{I}_m(\mathbf{x}+\mathbf{d})$ in frame m). Robust estimation methods discard pixels where the difference between the model and the data $\|\mathrm{I}_n(\mathbf{u}) - \sum_i a_i \phi\left(\mathrm{I}_m(\mathbf{u}+\mathbf{d})\right)\|$ is larger than some threshold. Automated methods for determining the threshold have been proposed recently [Che03, Wan04a, Dah04, Dah05]. In the present context, the difference in intensities can be large and the bias induced by the outliers important, thus such techniques are not well suited to the problem.

There is then a need for new techniques to estimate robustly a non-parametric mapping. The following paragraphs of this chapter study two attempts to robustify the estimation process. Both methods consider the transfer of pdfs in one dimension. As seen in chapter 1, one dimensional mapping can be used for any dimensional pdf transfer and is thus sufficient. The first method extends the standard solution of equation 2.11 and is thus based on the use of the original pdf $f(u)$ and the target pdf $g(v)$. The second method assumes that the correspondences between pixels are known. This means that it is possible to estimate the joint distribution $\pi(u, v)$, and to take advantage of this measure to establish the mapping.

5.1 PDF Transfer Using the Distributions Separately

The idea proposed in this section is to extend the original estimation method. The method considers both original and target pdfs f and g separately, and the joint statistic $\pi(u, v)$ is not a priori known. One important interest of the method is that the method is robust against any local motion within the region of interest of the picture. This means also that the pictures do not need to be accurately registered beforehand: an approximated global registration is most of the time sufficient.

The robust estimation is based on the minimisation of some functional, or energy, that depends both on the amount of outliers (data constraint) and the prior on the mapping (mapping smoothness). The proposed energy for a mapping is given by this linear combi-

nation:

$$E(t) = E_d(t) + \lambda_1 E_s(t) \tag{5.2}$$

where the constant parameter λ_1 tunes the smoothness of the mapping.

Data Constraint E_d. Consider that the optimal mapping t has been found. Without any presence of outlier, the two pdfs should match after mapping, and it would stand to reason that where the distributions differ indicates the presence of outliers. As previously stated, a distribution can be either represented by their pdf f and g or their cumulative pdf (cdf) F and G.

Working with pdfs is common, but requires an extra care in the estimation of the pdfs to avoid quantisation issues. The quantisation artefact is a common artefact in image and signal processing. The quantisation acts as a mapping that slightly moves the data into predefined bins. Even if not visible on the picture itself, the quantisation has a dramatic impact on the shape of the pdfs (see figure 5.1-a). It has however only little effect on the cdf representations (see figure 5.1-b). The quantisation artefacts on the pdf could be reduced by smoothing the pdf, but this would also degrade the estimation of the mapping t. Thus manipulating the cumulative pdfs is clearly preferable. The use of the cumulative pdf for comparing pdfs is in fact related to the Wasserstein distance W_p which can be expressed as:

$$W_p = \left(\int_0^1 |F^{-1}(\alpha) - G^{-1}(\alpha)|^p d\alpha \right)^{1/p}, \quad p \geq 1 \tag{5.3}$$

and for $p = 1$ the notation is equivalent to

$$W_1 = \int_{-\infty}^{\infty} |F(u) - G(u)| du \tag{5.4}$$

The chosen data energy is based on this 1-Wasserstein distance and is as follows:

$$E_d(t) = \int_u |G(t(u)) - F(u)| du \tag{5.5}$$

Mapping Smoothness E_s. The prior on the mapping is highly dependent on the application considered. In flicker for example, the nature of the mapping can greatly vary from one sequence to another. This legitimates the use of a very weak prior, and in this case, the prior is designed to minimise the amount of stretching, which can be defined by the deviation of the mapping gradient to 1. A possible measure for the amount of stretching is given by $\ln(t'(u))$, and results in the following smoothness energy:

$$E_s(t) = \int_u \left| \ln(t'(u)) \right|^2 f(u) du \tag{5.6}$$

Note that both the data energy E_d and the smoothness energy E_s have to be symmetric for the random variables U and V, *i.e.* the problem of mapping U to V and mapping V to U

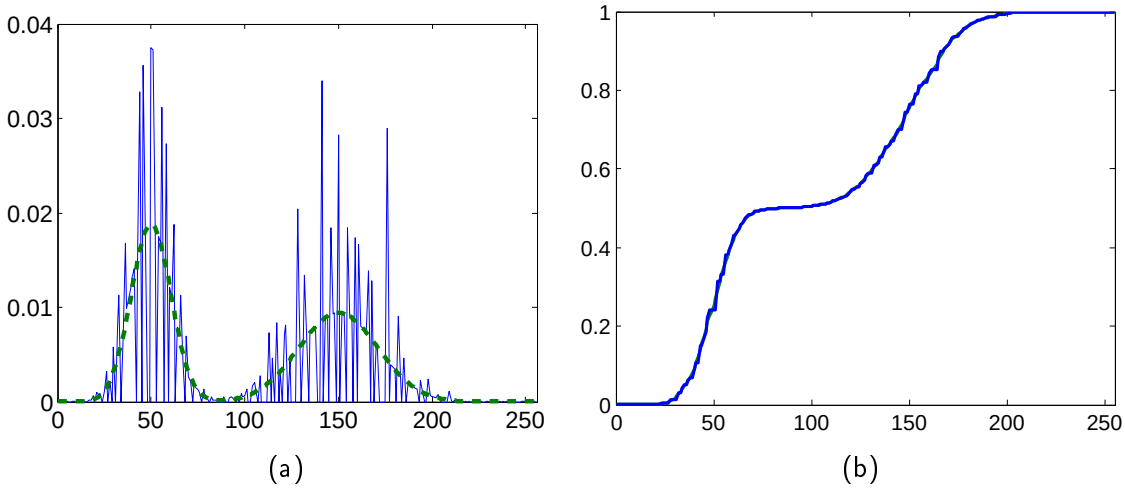


Figure 5.1: Effect of quantisation on a pdf (a) and on the cdf (b). The pdf corresponding to the quantised image is in solid blue, and the original pdf is in dotted green). It is clear that the cdf gives here a more robust representation.

should be equivalent. This is the case for the data energy E_d since the Wasserstein distance defines a proper metric and is thus symmetric ($W_1(f,g) = W_1(g,f)$). It is possible to show the same result in the case of the mapping smoothness, by noting that $f(u)du = g(v)d(v)$ and that $t^{-1\prime}(v) = 1/t'(u)$. Then the smoothness energy of the inverse problem is as follows:

$$\begin{aligned} E_s(t^{-1}) &= \int_v \left|\ln(t^{-1\prime}(v))\right|^2 g(v)dv \\ &= \int_u \left|\ln(t^{-1\prime}(v))\right|^2 f(u)du \\ &= \int_u \left|\ln(1/t'(u))\right|^2 f(u)du \\ &= \int_u \left|\ln(t'(u))\right|^2 f(u)du = E_s(t) \end{aligned} \tag{5.7}$$

which proves the symmetry of the chosen smoothness energy.

Solving for the Best Mapping. Combining both the data and smoothness energy terms results in the following overall expression for the energy:

$$E(t) = \int_u |F(u) - G(t(u))|\, du + \lambda_1 \int_u \left|\ln t'(u)\right|^2 f(u)du \tag{5.8}$$

The robust estimation of the mapping is the mapping that minimises this energy:

$$\hat{t} = \arg\inf_t E(t) \tag{5.9}$$

A discrete solution of the minimisation of equation (5.8) can be obtained by Dynamic Programming. To do so, the random variables U and V are first discretised, say to N_d

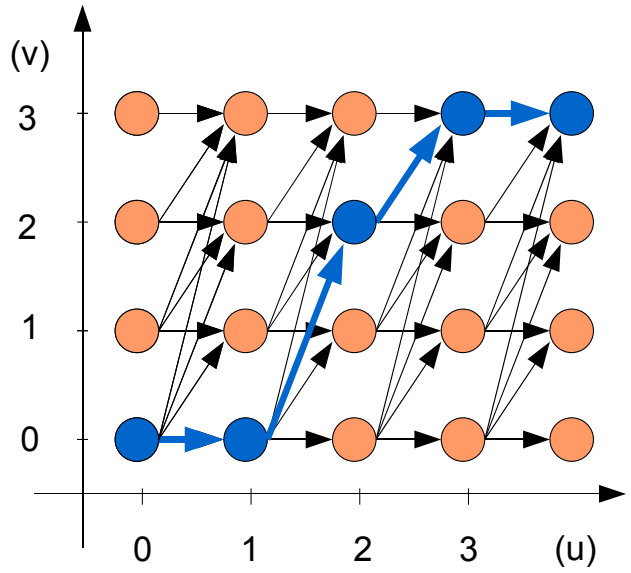


Figure 5.2: Structure of the Graph. Only upwards links between consecutive nodes are possible. A possible mapping is highlighted in blue.

values each. From all combinations of couple values (U, V), it is possible to build a graph as shown in figure 5.2. Consider that the nodes (u, v) of the graphs correspond to the case where $t(u) = v$, the minimisation problem is then equivalent to finding the cheapest path through this graph. Note that it is impossible to map $t(u)$ into two different values, thus there is no link between the node (u, v) and (u, w) for $w \neq v$. Furthermore, since the mapping should be increasing, there is no link between $(u-1, w)$ and (u, v) for $w < v$.

The energy $E(t)$, is evaluated by summing up the costs associated with the nodes of the path $\{(0, t(0)), \cdots, (N_d, t(N_d))\}$ and the costs of the links between consecutive nodes. The costs are defined to be consistent with equation (5.8):

$$\begin{cases} |F(u) - G(v)| & \text{cost for node } (u, v) \\ f(u)\ |\ln(v-w)|^2 & \text{cost for link between node}(u, v) \text{ and node } (u-1, w) \text{ with } w < v \\ +\infty & \text{cost for all other links} \end{cases} \tag{5.10}$$

The path of minimal energy is found using the Viterbi algorithm [Rab86], and this minimum path gives a discrete solution to the equation (5.8).

Higher Accuracy Estimation. However, the numerical minimisation as proposed gives poor results and needs further improvements. The reason is that the range of values for the stretching is $t'(u) \in \{0, 1, 2, 3, \cdots\}$, and favours high values of derivative whilst ignoring values of the derivative lower than 1. Ideally the range of possible stretching should be symmetric with 1, with for example $t'(u) \in \{\cdots, 1/2, 1, 2, \cdots\}$, which yields to equi-repartition of low and high derivative values.

The solution proposed here is to transform the random variables (U, V) into two new random variables (Q, R)

$$\begin{cases} Q = \dfrac{U+V}{\sqrt{2}} & ; \quad R = \dfrac{U-V}{\sqrt{2}} \\ U = \dfrac{Q+R}{\sqrt{2}} & ; \quad V = \dfrac{Q-R}{\sqrt{2}} \end{cases} \tag{5.11}$$

This approach corresponds to a rotation of the joint space by $\pi/4$, and it has already been used by Pitié and Kokaram [Pit03, Kok03] to reduce the bias in the regression of the flicker affine parameters. The idea is that the identity mapping, defined now by $Q = 0$, is a natural axis of symmetry. In particular, a result shown hereafter is that the strictly monotonous constraint of the mapping $0 < t'_{UV}(u) < +\infty$ is changed by this new coordinates system into $-1 < t'_{QR}(q) < 1$.

In the UV space, the derivative vector $[1, t'_{UV}]$ is transformed into $\frac{1}{\sqrt{2}}[t'_{UV}+1, t'_{UV}-1]$ in the QR space. Thus the derivative in the QR space t'_{UV} is related to the derivative in the QR space by:

$$t'_{UV} = \frac{t'_{QR}-1}{t'_{QR}+1} \quad \text{and} \quad t'_{QR} = \frac{t'_{UV}-1}{t'_{UV}+1} \tag{5.12}$$

and the monotonous constraint $0 < t'_{UV} < +\infty$ is thus equivalent to $-1 < t'_{QR} < +1$. With this representation however, the range of values for the discrete gradient t'_{QR} is limited to 3 values: $t'_{QR} \in \{-1, 0, 1\}$. Thus the accuracy needs to be improved by augmenting the number of states for each value q of Q by a factor M.

Using both the transformation in the QR space and the increase of states, the costs of the nodes and links of the new graph can now be defined by:

$$\begin{cases} |F\left(\frac{q+r/M}{\sqrt{2}}\right) - G\left(\frac{q-r/M}{\sqrt{2}}\right)|^2 & \text{cost for node } (q, r) \\ f\left(\frac{q+r/M}{\sqrt{2}}\right) \left|\ln\left(\frac{r-s-M}{r-s+M}\right)\right|^2 & \text{cost for link between node } (q, r) \text{ and} \\ & \quad \text{node } (q-1, s) \text{ with } -1 < \frac{r-s}{M} < 1 \\ +\infty & \text{cost for all other links} \end{cases} \tag{5.13}$$

Complexity. The graph is composed of N_d columns, each column is composed of $M \times N_d$ states and each state has $2\ M + 1$ transitions to preceding states. The overall complexity of the estimation using Viterbi is thus given by $\mathcal{O}(N_d \times (M\ N_d) \times (2\ M + 1) + N_d)$, which is roughly $\mathcal{O}(2\ N_d^2 M^2)$. This is to be compared with the non robust simple pdf transfer technique which is of $\mathcal{O}(N_d)$. Note however that the value of M is typically small (4 is sufficient).

Results can be seen on figures 5.3, 5.4 and 5.5. The original image is displayed on the top left, the target image on the bottom left and the resulting mapping using the robust estimation on the bottom middle and using the simple pdf transfer on the bottom right.

The estimated mappings are plotted on the top right (simple estimation in dashed blue, robust estimation in solid green and ground truth in solid red).

On figure 5.3, the estimation is tested against a synthetic case containing 33% of outliers. Although not perfect, the robust estimation does sensibly improve the quality of the results. In particular it attenuates effectively the amount of stretching. On figure 5.4, the method is tested on real images without outliers. The point to be made here is that the robust estimation can still estimate near-exact results if there is no outlier, even if the real stretching is extreme. On figure 5.5, the method is tested on real images with a strong presence of outliers. The robust estimation convincingly attenuates the stretching and thus reduces the grainy aspect of the recoloured picture.

5.2 PDF Transfer Using the Joint Distribution.

In this section, it is assumed that the correspondence between the pixels is known. The dense map of the correspondences can be established using Motion Estimation algorithms as in [Hor81, Bie87, Bla93, Bro04] and for a more thorough review [Tek95, Sti99]. This is an extremely difficult problem, which requires an heavy computational load. A weaker correspondence map can be obtained by simply using a rigid global motion estimation, like translational motion [Kug75, Cra04], affine [Odo95, Duf00] or homographic models. Note that it is well acknowledge that no algorithm so far can estimate a perfect correspondence map and that most of the time, the presence of outliers still remains an issue.

Estimation of the Mapping Unlike in the previous paragraph, the idea is not to minimise the amount of outliers but rather to maximises the amount of inliers, *i.e.* the number of couples $(u, v = t(u))$ that follow the mapping model. With the same smoothness constraint as previously defined, the energy to maximise is then defined by:

$$E(t) = \int_{u,v=t(u)} \pi(u,v) du dv - \lambda_2 \int_u |\log(t(u))| f(u) du \tag{5.14}$$

The maximisation of the energy can be obtained by discretisation and then by using the Viterbi algorithm. The costs associated with the nodes and the links between nodes are defined by:

$$\begin{cases} \pi(u,v) & \text{cost for node } (u,v) \\ f(u)\ |\ln(v-w)|^2 & \text{cost for link between node } (u,v) \text{ and node } (u-1,w) \end{cases} \tag{5.15}$$

The cost associated with the node (u, v) is not strictly correct since it should be $\pi(u, v)dudv$. Unfortunately $dv = dt(u)$ is dependent on the mapping t, and this cannot be integrated in this graph framework. The term is approximated by assuming a low stretching and thus $dv \approx du$.

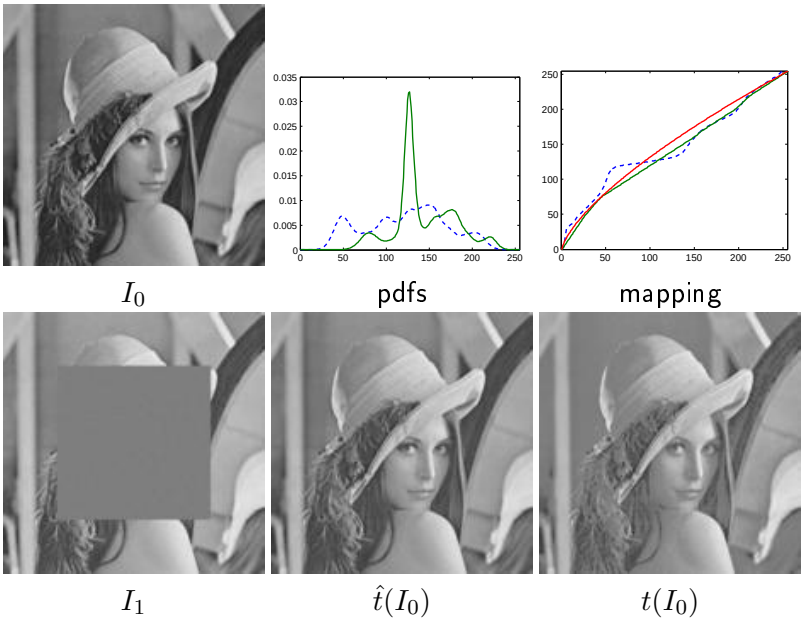


Figure 5.3: Estimation Results for an artificial example. On the top right, in red the ground truth mapping, in solid green the robust mapping $\hat{t}$ and in dashed blue the non-robust mapping t.

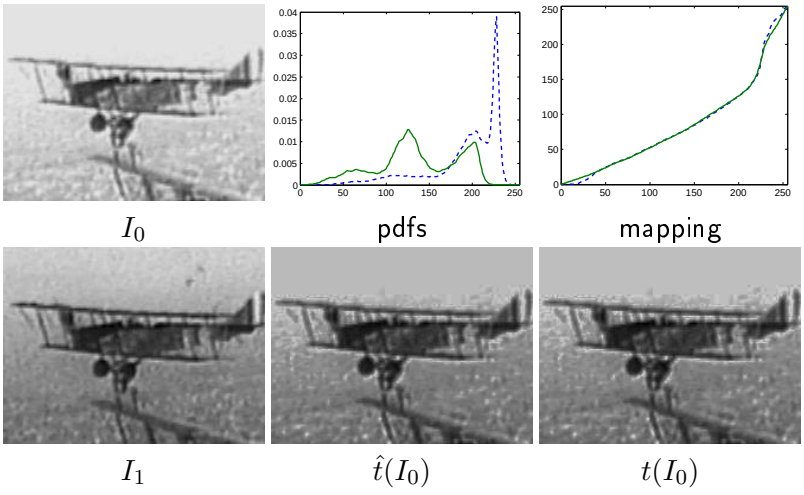


Figure 5.4: Estimation Results for a real example without outliers.

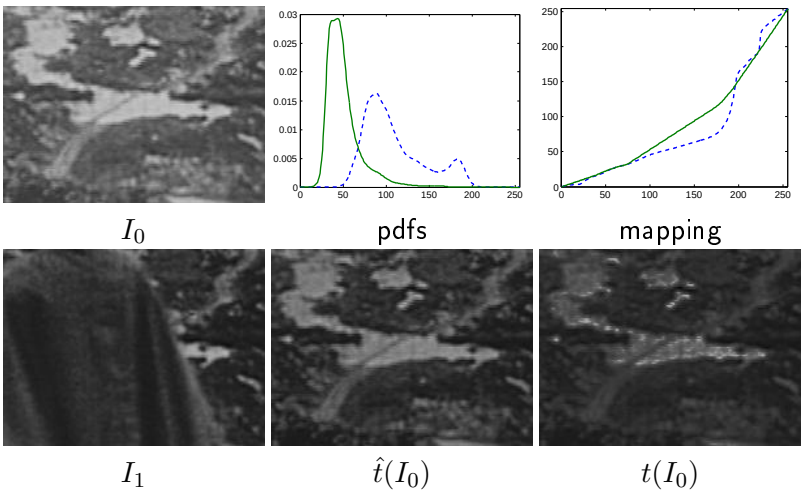


Figure 5.5: Estimation Results for a real example with outliers.

Once again, a big improvement can be gained by rotating the space by $\pi/4$ and using more states. The resulting graph has the following costs:

$$\begin{cases} \pi\left(\frac{q+r/M}{\sqrt{2}}, \frac{q-r/M}{\sqrt{2}}\right) & \text{cost for node } (q,r) \\ f\left(\frac{q+r/M}{\sqrt{2}}\right) \left|\ln\left(\frac{r-s-M}{r-s+M}\right)\right|^2 & \text{cost for link between node } (w,z) \text{ and} \\ & \text{node } (q-1,r) \text{ with } -1 < \frac{r-s}{M} < 1 \\ +\infty & \text{cost for all other links} \end{cases} \tag{5.16}$$

Inliers Enhancement. The estimation still remains difficult in cases where the amount of outliers is extreme. Consider the case study of the two images I_0 and I_1 presented on figure 5.6. The image I_0 is occluded by a large object of constant colour u_0. The corresponding joint pdf, displayed on (c), is thus composed into two parts: the inlier part which manifests as the $u \to v = t(u)$ curve, and the occlusion part which manifests as a spurious vertical ridge $x = u_0$. Sadly, as seen on figure 5.6-(e), some of the probabilities on this outlier line are greater than the corresponding inliers one (i.e. $\pi(u_0, t(u)) > \pi(u, t(u))$), and the estimation presented before tends to follow this outlier ridge (see figure 5.8).

The difficulty thus arises from the nature of the spurious structure in the joint pdf. The idea is then to use a pre-processing filter to simplify the ridge structure of the outliers. Simply weighting out parts that are too far away from the identity line $v = u$ is a classical robust regression approach but it becomes inefficient when the mapping t makes inlier pairs deviate a lot. It is then necessary to examine closer at the nature of the joint distribution. Consider that the joint pdf can be expressed as a combination of an inlier part $\pi_i(u,v)$ and an occlusion/outlier part $\pi_{\mathcal{O}}(u,v)$:

$$\pi(u,v) = (1-\alpha)\ \pi_i(u,v) + \alpha\ \pi_{\mathcal{O}}(u,v) \quad (\alpha \in [0,1]) \tag{5.17}$$

Ideally the inlier joint distribution $\pi_i(u,v)$ is null everywhere except for the curve $(u, t(u))$ where it is maximum column and row wise. Since the occluding object usually presents an independent distribution from the original picture, the joint probability of outliers $\pi_{\mathcal{O}}(u,v)$ could be ideally considered has separable:

$$\pi_{\mathcal{O}}(u,v) = \pi_{\mathcal{O}}(u)\pi_{\mathcal{O}}(v) \tag{5.18}$$

To help the mapping estimation, it would be intuitively beneficial to explore a weight $\gamma(u,v)$ on the joint distribution which would 1) enhance the parts of the joint pdf that are maximum row and column wise to enhance the inliers, and 2) to reduce the separable part of the pdf to reduce the amount of outliers. In an attempt to achieve this, the following weight function is proposed:

$$\gamma(u,v) = \pi(u|v)\ \pi(v|u) = \frac{\pi(u,v)^2}{\pi(u)\ \pi(v)} \tag{5.19}$$

Initialising $\pi^{(0)}(u,v)$ at $\pi(u,v)$, the weight could then be used iteratively:

$$\pi^{(n+1)}(u,v) = K^{(n+1)}\ \pi(u,v)\ \gamma^{(n)}(u,v) \tag{5.20}$$

where $K^{(n+1)}$ is a normalising constant.

To better understand the effect of using iteratively this weight on the joint pdf, consider again the simple case study presented in figure 5.6 where an object of colour $u_0 = 128$ is occluding an object in the original picture. It is shown hereafter that the method can actually remove the outlier ridge if and only if the following condition is satisfied:

$$(C): \quad \forall v,\ p(v_0|u_0) > p(v|u_0) \quad \text{with } v_0 = t(u_0) \tag{5.21}$$

Note that this condition still allows the outlier colour to be more important than the inlier colour. For example, it is possible to have $p(u, v = t(u)) < p(u_0, v)$ (see figure 5.6-e). Under condition (C), the following inequalities arise:

$$\begin{aligned}
\gamma(u_0,v_0) &= \pi(u_0|v_0)\ \pi(v_0|u_0) && (5.22)\\
\gamma(u_0,v_0) &= 1 \cdot\ \pi(v_0|u_0) && (5.23)\\
\forall v,\ \exists \epsilon > 0\ /\ \gamma(u_0,v_0) &> (1+\epsilon)\ \pi(v \neq v_0|u_0) && (5.24)\\
\gamma(u_0,v_0) &> (1+\epsilon)\ \pi(v \neq v_0|u_0)\ \pi(u_0|v \neq v_0) && (5.25)\\
\gamma(u_0,v_0) &> (1+\epsilon)\ \gamma(u_0, y \neq v_0) && (5.26)
\end{aligned}$$

This means that under (C), at iteration 0, the attenuation factor is always more important in the inlier part (u_0, v_0) than in the outlier vertical line $(u_0, v \neq v_0)$. Thus (C) is still satisfied after the first iteration, and after each following iteration:

$$\forall v,\ \exists \epsilon > 0\ /\ 0 \leq (1+\epsilon)^k\ \pi(u_0, y \neq v_0)^{(k)} < \pi(u_0,v_0) \leq 1 \tag{5.27}$$

This implies that the outlier ridge is completely removed since $\lim_{k\to\infty}\pi^{(k)}(u_0, v \neq v_0) = 0$ (see the results on figure 5.7).

Although this case-study examines a simple scenario of monochromatic occlusion, it illustrates the ability of the method to handle cases where the level of occlusion is severe. The condition (C) is indeed satisfied in most situations. Consider that the area 'under' the occlusion in I_1 has the same colour statistics $\pi_{\mathcal{O}}(v)$ than the rest of the picture $p_1(v)$ ($\pi_{\mathcal{O}}(v) = p_1(v)$. It follows then that

$$\begin{aligned}
\pi(u_0,v_0) &= (1-\alpha)p_1(v_0) + \alpha p_1(v_0) = p_1(v_0) && (5.28)\\
\pi(u_0, v \neq v_0) &= \alpha p_1(v) && (5.29)
\end{aligned}$$

Thus the condition (C) is fulfilled if

$$\alpha < \frac{p_1(v_0)}{\max_v p_1(v)} \tag{5.30}$$

In other words, the proportion of the occlusion α can be important if the colour of the occluding object is already frequent in the picture. The method even works regardless of the level of occlusion if the occlusion colour ($\alpha = 1$) is the most frequent in the rest of the picture ($p_1(v_0) = \max_v p_1(v)$).

In practice of course, the assumption made here do not necessarily apply exactly. This means that sometimes, inlier parts could be attenuated whereas outlier parts could be enhanced. The process however results in a joint pdf without any ridge structure. This is especially interesting when used in association with the robust estimation discussed earlier in this section, since the mapping will not be biased by the presence of a ridge.

Figure 5.8 shows the behaviour of the method in some real cases. The left column shows the original joint pdf $\pi(u, v)$ and its corresponding robust mapping estimation. The middle column diplays the joint pdf after outlier attenuation and its corresponding robust estimation . The right column shows the original joint pdf overlaid with both estimations. As seen on the first row, both estimations are able to handle correctly situations without outliers. The last two rows show the positive effect of the proposed outlier attenuation in the presence of significant outliers.

5.3 Conclusion

Accurately detecting outliers in a dataset is a very difficult problem. This problem is compounded by the fact the that in the general case, neither the inlier class, nor the outlier class, can be clearly modelled. The two estimation methods proposed here are a first attempt towards robust estimation of a non-parametric mapping. This problem has not yet been specifically addressed in the Computer Vision community; the proposed method provides valid results when there are no outliers, and significantly improves the quality of the mapping in the presence of outliers by comparison to the standard pdf transfer technique mentioned in the previous chapters.

This chapter also demonstrates the benefit of using cumulative pdf representations instead of pdf representations, to alleviate quantisation artefacts. It also analyses the impact of picture occlusions on the joint pdf to assist separation of the outlier and inlier classes by an iterative method.

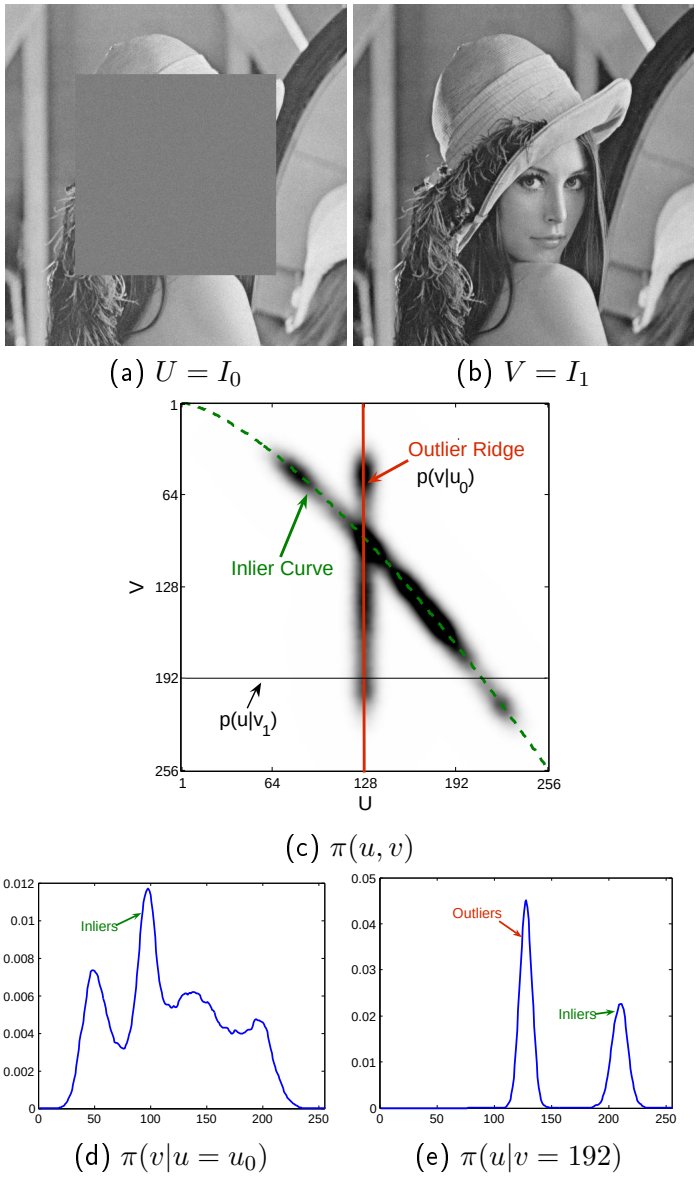


Figure 5.6: Case Study of a monochromatic occlusion. On the first row the original and the example pictures. The outlier block covers 33% of the original picture. On the middle row, the joint pdf of intensity pixels which is composed of an inlier part and an outlier vertical ridge located at the colour of the occlusion ($u_0 = 128$). On the last row, the conditional probability $\pi(v|u = u_0)$ corresponding to the outlier ridge, and $p(u|v = 192)$ corresponding to the horizontal slice marked on joint pdf figure. Note that for $p(u|v = 192)$, the outlier peak is higher than the inlier peak.

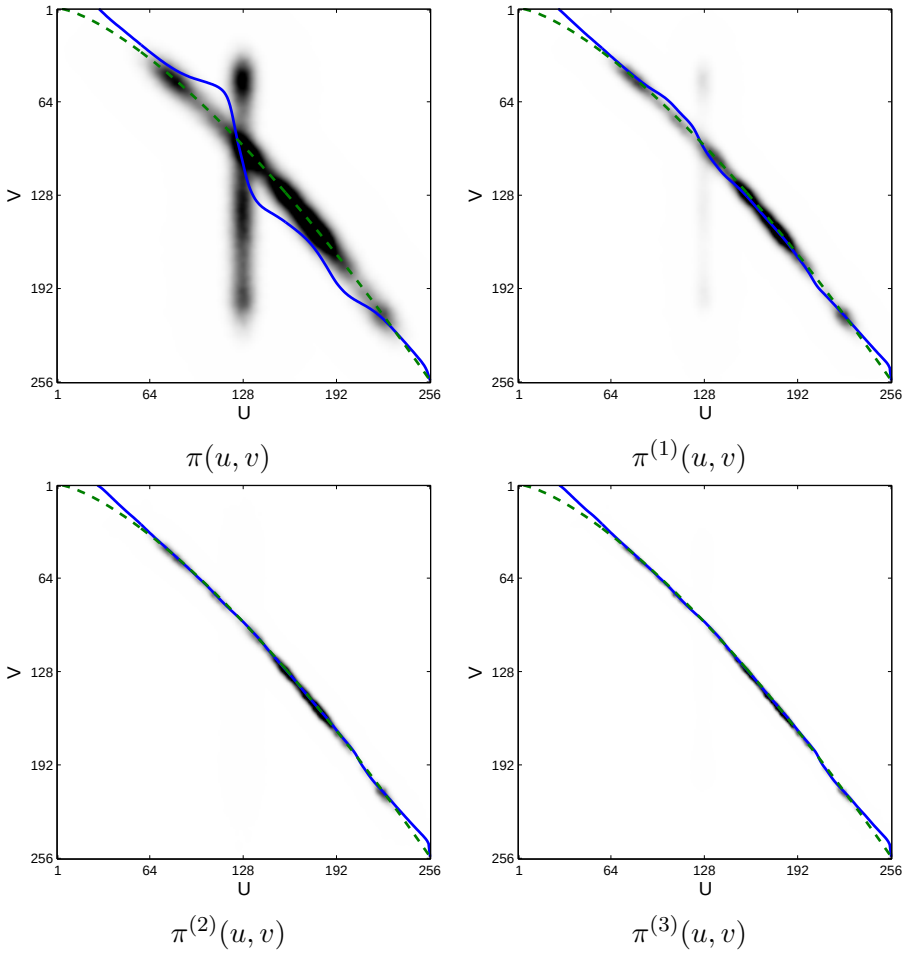


Figure 5.7: Results on the Case Study. The results of the joint pdf at iteration k are overlaid with the standard estimation of the pdf transfer mapping in solid blue, and the corresponding true mapping in dashed green. The outlier ridge disappears in less than 3 iterations.

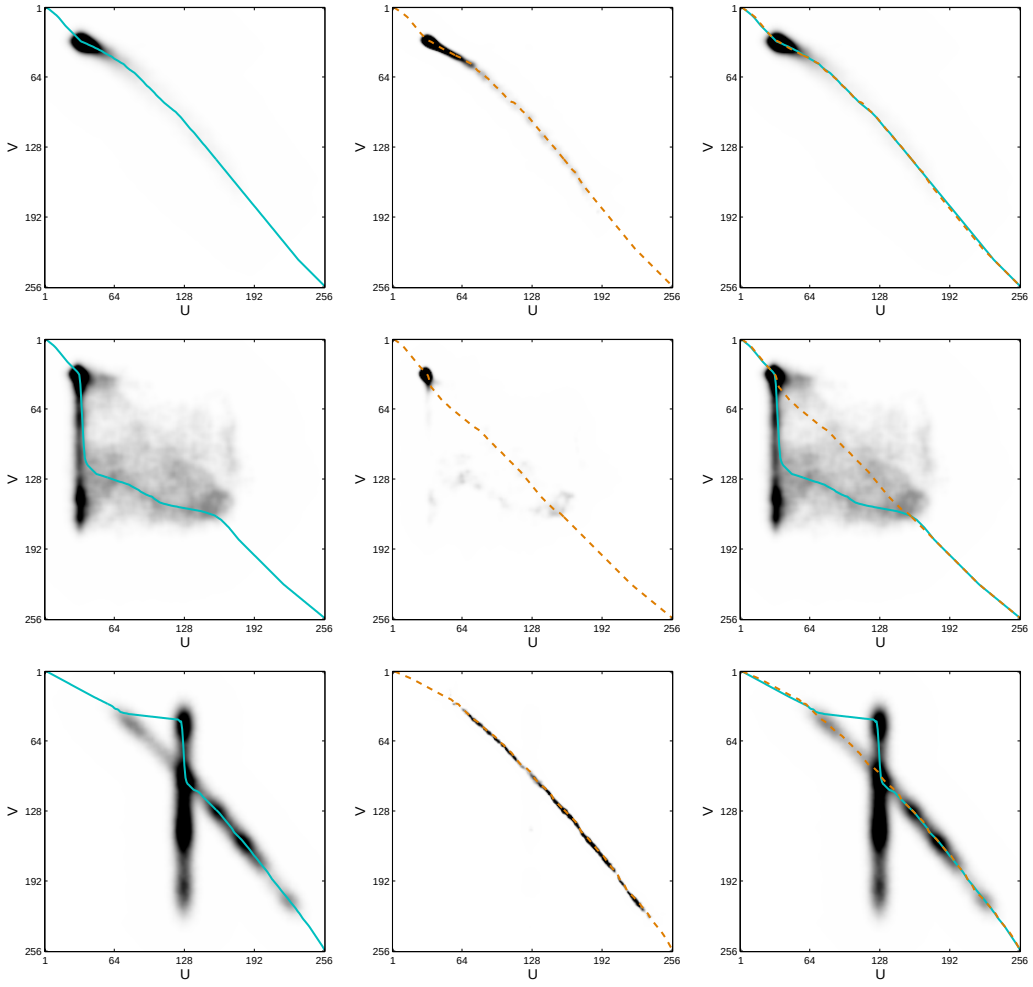


Figure 5.8: Estimation Results for real examples. On the left column the original joint pdf $\pi(u, v)$ and its corresponding robust mapping estimation. On the middle column, after outlier attenuation with corresponding robust estimation . On the right, original joint pdf overlaid with both estimations.

Part II

Tracking Applications

Chapter 6

Contour Following using Particle Filter Algorithms

The next two chapters present two different applications for video editing that involve probabilistic tracking techniques. The concept of probabilistic tracking and distribution transfer are closely related, in the sense that tracking an object is finding a mapping for the position of the object. The framework involved in both cases are however very different. In distribution transfer, the focus is given on finding a wrapping of the cumulative pdf of the distribution, whereas in tracking, the actual pdfs are directly manipulated. For instance, most probabilistic tracking methods are based on Maximum A Posteriori (MAP). The core of these methods is to estimate the position of the object by finding the position that maximises the pdf of the object presence probability.

The applications presented in this chapter and the following one employ two probabilistic methods derived from the Bayesian framework. The first chapter considers the problem of semi-automated tracing application for delineating objects. This is an on-line application and is well suited to the stochastic technique called particle filters. The next chapter is dedicated to off-line applications and is based on the deterministic method Viterbi.

6.1 Contour Tracing

Manual or semi-automatic contour following is an important task in image editing. The tracing of object contours in general is also seen as an important task in early vision [Mar82]. Cut-out tools that assist the user in following a contour, can be seen in Adobe Photoshop for instance. Automated or semi-automated contour following is complicated by the ambi-

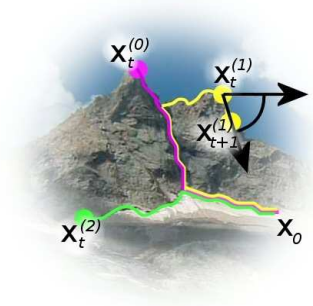
parts of this chapter are based on the work published in [Pit04b]

guity of any contour in an image. Not only is it difficult to track exactly the position of a contour because of poor image contrast and noise, but also it is impossible to foresee the contour chosen by the user on the basis of semantics.

Pérez *et al* [Pér01] have proposed a robust technique—called *JetStream*—for contour following that handles this ambiguity by sampling from the posterior distribution for the contour location. It is based on the use of a Particle Filter and its operation can be understood as explained in the following section.

Probabilistic Tracing Approach using Particle Filters

The approach proposed in JetStream [Pér01] to extract a contour can be understood by using an analogy with manual tracing. Starting from a point $\mathbf{x}_0$, the pencil draws a contour by following the edge of the picture. The current position of the pencil at time t is denoted $\mathbf{x}_t$. Tracing the contour can then be understood as *tracking* the pencil. The growing contour is represented by an ordered sequence $\mathbf{x}_{0:t} \equiv (\mathbf{x}_0 \dots \mathbf{x}_t)$.



Let θ_{t+1} be the angle formed by the segment $[\mathbf{x}_i; \mathbf{x}_{i+1}]$ with the horizontal axis and assume that the points are equally spaced by a step d. To simplify the problem, the pencil speed is supposed to be constant and therefore d is set to $d = 1$.

$$\mathbf{x}_{t+1} = \mathbf{x}_t + d \begin{bmatrix} \cos(\theta_{t+1}) \\ \sin(\theta_{t+1}) \end{bmatrix} \tag{6.1}$$

The idea of using Particle Filters for tracing is understood more easily with the help of the adjacent figure. While following a contour in the mountain picture, the pencil encounters bifurcations and edge junctions. To select the most likely path, the idea is to try all possible paths and to decide afterward which one is the best. In the mountain picture example, growing contours $\mathbf{x}_{0:t}^{(0)}$ (in pink), $\mathbf{x}_{0:t}^{(1)}$ (in yellow) and $\mathbf{x}_{0:t}^{(2)}$ (in green) correspond to 3 different possible tracings all originating from the same starting point $\mathbf{x}_0$. The Particle Filter framework—described properly in the next section—proposes to grow simultaneously a number of possible contours—also called *particles*. The particles can take separate decisions when they reach an edge junction. The framework decides whether a particle should grow further, duplicate itself, or stop, depending on its performance.

JetStream, though an elegant solution to a combinatorially difficult problem, suffers from an inability to handle sudden changes in direction without the use of a switching process. In effect, upon encountering a corner, the idea is to propose unconstrained direction possibilities in the expectation that one of the proposed direction will regain a contour 'lock'. This chapter resolves the problem by designing a directional probability density function (pdf) that is better able to control the evolution of the contour. Because of the reliability of this pdf it is then possible to relieve the need for heavy control on contour

smoothness. The particle filter framework is presented next and the new design explained as problems are highlighted.

6.2 Probabilistic Contour Tracking Framework

6.2.1 Standard Approach using Particle Filters

Recall that the ordered sequence $\mathbf{x}_{0:t} \equiv (\mathbf{x}_0 \dots \mathbf{x}_t)$ represents the 2D points of the curve being tracked. This chain is assumed to be a Markov Chain of order 2, ie. $p(\mathbf{x}|\mathbf{x}_{0:t}) = p(\mathbf{x}|\mathbf{x}_t, \mathbf{x}_{t-1})$. Given the observed image represented by a vector $\mathbf{y}$, a probabilistic approach to tracking proceeds by manipulating the posterior $p(\mathbf{x}_{0:t+1}|\mathbf{y})$ to estimate the most probable next position $\mathbf{x}_{t+1}$. This distribution can be written in a recursive form:

$$p(\mathbf{x}_{0:t+1}|\mathbf{y}) = p(\mathbf{x}_{t+1}|\mathbf{y}, \mathbf{x}_{0:t})\ p(\mathbf{x}_{0:t}|\mathbf{y}) \tag{6.2}$$

This form admits a solution which manifests as the propagation of densities from point to point on each contour. Bayes rule combined with the Markovian hypothesis on the contour leads to the following expression for the posterior:

$$p(\mathbf{x}_{0:t+1}|\mathbf{y}) \propto \prod_{i=2}^{t+1} p(\mathbf{x}_i|\mathbf{x}_{i-1}, \mathbf{x}_{i-2})\ p(\mathbf{y}|\mathbf{x}_i, \mathbf{x}_{i-1}) \tag{6.3}$$

It is then possible to show that the following recursion arises:

$$p(\mathbf{x}_{0:t+1}|\mathbf{y}) = p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{x}_{t-1})\ p(\mathbf{y}|\mathbf{x}_{t+1}, \mathbf{x}_t)\ p(\mathbf{x}_{0:t}|\mathbf{y}) \tag{6.4}$$

The term $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{x}_{t-1})$ corresponds to the *prior* on the contour and $p(\mathbf{y}|\mathbf{x}_{t+1}, \mathbf{x}_t)$ to the *data model*.

Although the prior and the data model might have an analytical expression, this expression presents usually no simple closed form. Sequential Monte Carlo methods (also called *particle filters*) provide however a flexible and easy way of propagating an approximation of this posterior distribution. In this framework the posteriors are approximated in a grid-based fashion by a finite set $(\mathbf{x}_{0:t}^{(m)})_{m=1\dots M}$ of M samples or *particles*:

$$p(\mathbf{x}_{0:t}|\mathbf{y}) \approx \sum_{m=1}^{M} w_t^{(m)} \delta(\mathbf{x}_{0:t} - \mathbf{x}_{0:t}^{(m)}) \tag{6.5}$$

where $\delta(.)$ denotes the Dirac delta measure which is 1 at 0 and zero otherwise; $w_t^{(m)}$ the importance weight attached to particle $\mathbf{x}_{0:t}^{(m)}$. Note that the particles correspond to contours $(\mathbf{x}_{0:t}^{(m)})$ and not to single 2D points. The posterior approximation can be propagated in time by the generic boostrap filter (or Sequential Importance Resampling (SIR) Particle Filter) [Aru02, Dou00] as proposed for instance in *JetStream*. At each time iteration,

the weights are chosen using the principle of *importance sampling* [Aru02, Dou00]. As it is known, it can be difficult to draw directly samples from the posterior $p(\mathbf{x}_{0:t}|\mathbf{y})$. However, it is usually possible to find as a first step a proposal pdf, that is called *importance density*, from which samples can be easily drawn. In the bootstrap filter the proposal is simply the prior density $p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t-1}^{(m)})$ and the weights are therefore given by the likelihood [Aru02, Dou00]:

$$w_{t+1}^{(m)} \propto \frac{p(\mathbf{x}_{t+1}^{(m)}, \mathbf{x}_t^{(m)}|\mathbf{y})}{p(\mathbf{x}_{t+1}^{(m)}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t-1}^{(m)})} = p(\mathbf{y}|\mathbf{x}_{t+1}^{(m)}, \mathbf{x}_t^{(m)}) \tag{6.6}$$

To avoid that the weight distribution becomes more and more skewed and leads to the degeneracy of the particles, the bootstrap filter adds a *selection* step. In this crucial step the M growing contours are drawn from the normalised weight distribution. The idea is that 'good' contours will be statistically replicated whereas 'bad' one will be deleted.

From these approximations of the posterior distribution $p(\mathbf{x}_{0:t}|\mathbf{y})$, an approximation of the Maximum A Posteriori can be derived by taking the 'best' contour.

6.2.2 Exact Importance Sampling

A good choice for the proposal is key to the success of the particle filter algorithm. In *JetStream*—as in many tracking algorithms—the importance distribution is however constrained by the smoothness of the particle's trajectory. For instance, the trajectory of the contour cannot deviate by more than a few degrees. A special case is made when particles reach a corner: particles are then allowed to take any direction. With such hypotheses the position of the next particle is strongly restricted and at the price of sometimes missing sharp turns in the contour as shown in figure 6.5. This problem arises due to the difficulty in designing a prior that both plays the role of a good proposal, able to restrict the search area, and that also gives enough flexibility to model the dynamics of the contour.

A deviation from the boostrap filter approach, is to reconsider equation (6.4) and choose directly as the proposal

$$q(\mathbf{x}_{t+1}|\mathbf{y}, \mathbf{x}_{0:t}^{(m)}) = \frac{p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t-1}^{(m)})\ p(\mathbf{y}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t+1})}{\int_{\mathbf{x}_{t+1}} p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t-1}^{(m)})\ p(\mathbf{y}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t+1})\ d\mathbf{x}_{t+1}} \tag{6.7}$$

The proposal is thus optimal and results in a perfect sampling of the posterior, without any additional constraint on the prior function. With this proposal, the weights can now be written as follows

$$w_{t+1}^{(m)} \propto \frac{p(\mathbf{x}_{t+1}|\mathbf{y}, \mathbf{x}_{0:t}^{(m)})}{q(\mathbf{x}_{t+1}|\mathbf{y}, \mathbf{x}_{0:t}^{(m)})} \tag{6.8}$$

$$= \int_{\mathbf{x}_{t+1}} p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t-1}^{(m)})\ p(\mathbf{y}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t+1})\ d\mathbf{x}_{t+1} \tag{6.9}$$

Algorithm 3 Outlines of the Oriented Particle Spray

1: **Initialisation.** $t = 0$, manually set $\mathbf{x}_0^{(m)} = \mathbf{x}_0$

2: **Importance Sampling Step**.
For each particle m, do:

- **Prediction:**
$$\mathbf{x}_{t+1}^{(m)} \sim p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t-1}^{(m)})p(\mathbf{y}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t+1}) \tag{6.10}$$
- **Weighting:**
$$w_t^{(m)} = \int_{\mathbf{x}_{t+1}} p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t-1}^{(m)})p(\mathbf{y}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t+1})d\mathbf{x}_{t+1} \tag{6.11}$$

3: **Selection Step.**
Resample with replacement M contours from the set $\left(\mathbf{x}_{0:t+1}^{(m)}; m = 1, \ldots, M\right)$ according to the normalised importance weights $w_t^{(m)} / \sum_m w_t^{(m)}$.

The difficulty lies now in drawing samples $\mathbf{x}_{t+1}^{(m)}$ directly from the proposal. By continuity of the tracing, only neighbouring pixels are considered, and the pdfs involved are thus of dimension 1 (i.e. the dimension of the neighbourhood). Working in dimension 1 makes possible to consider a simple discretisation of the 1D pdf. Thus it only remains to find an expression for the prior $p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t-1}^{(m)})$ and the likelihood $p(\mathbf{y}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t+1})$. These are studied hereafter in section 6.3 and 6.4.

The final outline of the contour tracker is summarised by the algorithm 3.

6.3 The Prior on the Contours

Since the prior does not serve as a proposal, the constraint on the dynamic of the contour can be relaxed. Thus the only constraint which is chosen is that a tracing cannot reverse over its own path. This problem—trivial in appearance—has to be handled carefully to avoid that the particles try to rediscover their exact reverse trajectory. To better understand the constraint, the prior can then be rewritten using trajectory angle θ,

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t-1}^{(m)}) = p(\theta_{t+1}|\theta_t^{(m)}) \tag{6.12}$$

The solution adopted is to disallow angles diametrically opposed to the previous direction angle taken by the particle.

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(m)}, \mathbf{x}_{t-1}^{(m)}) = \phi_b(\text{dist}(\theta_{t+1}, \theta_t^{(m)})) \tag{6.13}$$

where ϕ_b is a kernel function based on the distance between unwrapped angles as represented in figure 6.3.

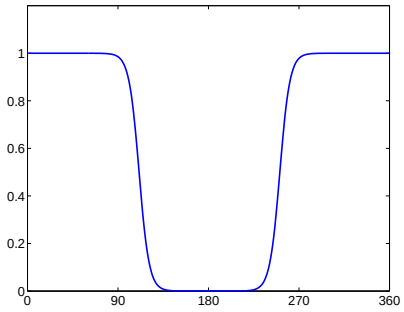


Figure 6.1: Values of the kernel for $\theta \to \phi_b(\text{dist}(\theta, 0))$.

6.4 Likelihood

Introducing the angle notation as previously, the likelihood can be re-expressed as

$$p(\mathbf{y}|\mathbf{x}_{t+1}, \mathbf{x}_t^{(m)}) = p(\mathbf{y}|\theta_{t+1}, \mathbf{x}_t^{(m)}) \tag{6.14}$$

which stands for the probability that at pixel $\mathbf{x}_t^{(m)}$ an edge goes along the direction θ_{t+1}. The likelihood presented in *JetStream* relies mainly on the simple definition of the edge: the angle of the edge is defined[1] by $\theta = \text{atan2}(I_y, I_x)$ and its norm by $N = \sqrt{I_x^2 + I_y^2}$, where I_x and I_y are the derivatives of the picture I. This definition presents a strong drawback: it assumes that only one edge passes by the pixel of consideration. In consequence, this approach cannot cope with corners, or junctions. Even if *JetStream* attempts to handle this problem by using a Harris corner detector beforehand, figure 6.5 shows that it still might fail in its tracking, especially if there is no clear corner. Instead of using a two stage strategy, it is proposed here to fully integrate the orientation of the contours in the likelihood function. To do so, the quantity $p(\mathbf{y}|\theta_{t+1}, \mathbf{x}_t^{(m)})$ still relies on the angular edge direction, but here the edge direction is estimated by an approach similar to the Steerable Filters [Sim95, Per95] and more specifically in [Yu99]. The method presented here differs in the expression of the filter bank which is derived. In particular, this method integrates directly the interpolation between pixels, which avoids aliasing in the filters.

Assume that the probability that at pixel $\mathbf{x}_t^{(m)}$, the direction θ_{t+1} corresponds to an edge is proportional to the absolute variation of the angular intensity, i.e.:

$$p(\mathbf{y}|\theta_{t+1}, \mathbf{x}_t^{(m)}) \propto \left| \frac{dI_\theta}{d\theta} \right| \tag{6.15}$$

where the intensity in direction $\theta \in [0; 2\pi]$ I_θ is equal to:

$$I_\theta = \int_{\rho>0} I(\rho, \theta) g(\rho) \, \mathrm{d}\rho \tag{6.16}$$

[1] atan2 is $\tan^{-1}$ with unwrapped angles.

(ρ, θ) is a pixel coordinate location in polar coordinates, with origin at the current contour point. The integral is just the sum of pixels along the direction θ. $g(\rho)$ is a smoothing kernel (a gaussian for instance), which ensures that pixels closer to the origin are more important than those further away. Note that $\rho > 0$ is required to have a meaningful direction metric.

To interpolate I_θ to all values of θ it is taken advantage of the periodicity of I_θ (since the function would repeat every 360^{deg}) and thus of the Fourier series of I_θ:

$$I_\theta = \sum_{n=0}^{n=N} \mathrm{H}_n e^{jn\theta} \tag{6.17}$$

and respectively for its derivative:

$$p(\mathbf{y}|\theta_{t+1}, \mathbf{x}_t^{(m)}) \propto \left| \frac{dI_\theta}{d\theta} \right| = \left| \sum_{n=0}^{n=N} n \; j \; \mathrm{H}_n e^{jn\theta} \right| \tag{6.18}$$

The Fourier coefficients can be computed with:

$$\mathrm{H}_n = \int_{\phi=0}^{2\pi} \int_{\rho=0}^{+\infty} I(\rho, \phi) w_n(\rho, \phi) \; \rho \mathrm{d}\phi \mathrm{d}\rho \tag{6.19}$$

where

$$w_n(\rho, \phi) = \frac{1}{\rho} g(\rho) e^{jn\phi} \tag{6.20}$$

The continuous values of $I(\rho, \theta)$ can be obtained by interpolation from the image grid. The interpolation corresponds to a convolving of the sampled picture $\mathrm{I}(x, y)$ with an interpolation kernel k. To simplify notations, consider the Cartesian coordinates as follows

$$\begin{cases} (u, v) & \equiv & (\rho, \phi) = (\sqrt{u^2 + v^2}, \mathrm{atan2}(v, u)) \\ (x, y) & \equiv & (r, \psi) = (\sqrt{x^2 + y^2}, \mathrm{atan2}(y, x)) \end{cases} \tag{6.21}$$

Then yields the following expression for H_n:

$$\mathrm{H}_n = \int_{u,v} (\mathrm{I} * k)(u, v) w_n(u, v) \; \mathrm{d}u\mathrm{d}v \tag{6.22}$$

$$= \int_{u,v} \left(\sum_{x,y} \mathrm{I}(x, y) k(u - x, v - y) \right) w_n(u, v) \; \mathrm{d}u\mathrm{d}v \tag{6.23}$$

$$= \sum_{x,y} \mathrm{I}(x, y) \int_{u,v} k(u - x, v - y) w_n(u, v) \; \mathrm{d}u\mathrm{d}v \tag{6.24}$$

By expressing the interpolation kernel in this way, it becomes possible to derive a complete framework for calculation of the direction information. In summary, $\mathrm{H_n}$ can be computed

Algorithm 4 Summary of the algorithm for computing the likelihood $p(\mathbf{y}|\theta_{t+1}, \mathbf{x}_t^{(m)})$

1: **Offline computations:**

$$h_n(x,y) \propto \int_{\phi=0}^{2\pi}\int_{\rho=0}^{+\infty} \exp\left(-\frac{\rho^2+r^2-2r\rho\cos(\psi-\phi)}{2\sigma_k^2} - \frac{\rho^2}{2\sigma_g^2} + jn\phi\right) \mathrm{d}\phi\mathrm{d}\rho \quad (6.29)$$

with the normalising constant:

$$C = \frac{1}{2\pi\sigma_k^2\sqrt{2\pi\sigma_g^2}} \quad (6.30)$$

2: Online computations:

$$\mathrm{H}_n = \sum_{x,y} \mathrm{I}(x,y) h_n(x,y) \quad (6.31)$$

$$p(\mathbf{y}|\theta_{t+1}, \mathbf{x}_t^{(m)}) \propto \left|\frac{dI_\theta}{d\theta}\right| = \left|\sum_{n=0}^{n=N} jn\mathrm{H}_n e^{jn\theta}\right| \quad (6.32)$$

by the use of a filter bank, whose masks $h_n(x,y)$ can be computed offline, as follows:

$$\mathrm{H}_n = \sum_{x,y} \mathrm{I}(x,y) h_n(x,y) \quad (6.25)$$

$$h_n(x,y) = \int_{u,v} k(u-x, v-y) w_n(u,v)\ \mathrm{d}u\mathrm{d}v \quad (6.26)$$

The integrals can be approximated using numerical techniques (in this study using MATLAB). Here is a possible implementation for the kernels k and g:

$$g(\rho) = \frac{1}{\sqrt{2\pi\sigma_g^2}} \exp\left(-\frac{\rho^2}{2\sigma_g^2}\right) \quad (6.27)$$

$$k(u-x, v-y) = \frac{1}{2\pi\sigma_k^2} \exp\left(-\frac{\rho^2+r^2-2r\rho\cos(\psi-\phi)}{2\sigma_k^2}\right) \quad (6.28)$$

Figure 6.2 shows examples of 11-tap filters h_n.

Examples of Likelihood. Figure 6.3 shows an example of angular variations of the intensity. On the right the values of $\left|\frac{dI_\theta}{d\theta}\right|$ correspond to the pdf of the contour directions at the centre of the picture on the left. This was obtained for $\sigma_g = 2.25$, $\sigma_k = 0.7$ at order $N = 10$. On the left side, the red lines correspond to the lobes of $\left|\frac{dI_\theta}{d\theta}\right|$.

6.5 Results and Remarks

Figure 6.5 shows some simulations of *JetStream* (on the left) and the Oriented Particle Spray (on the right). *JetStream* tends to overshoot sharp angles of the contours whereas

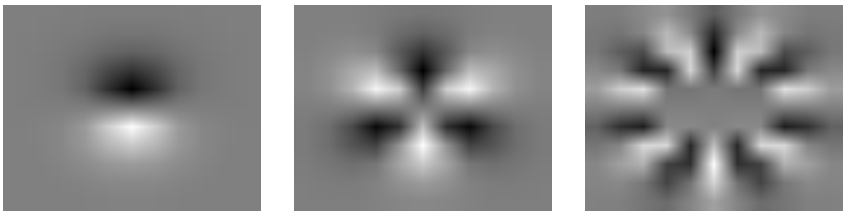


Figure 6.2: Examples of 11-tap filters $h_n(x, y)$ for $n = 1$, $n = 3$ and $n = 7$.

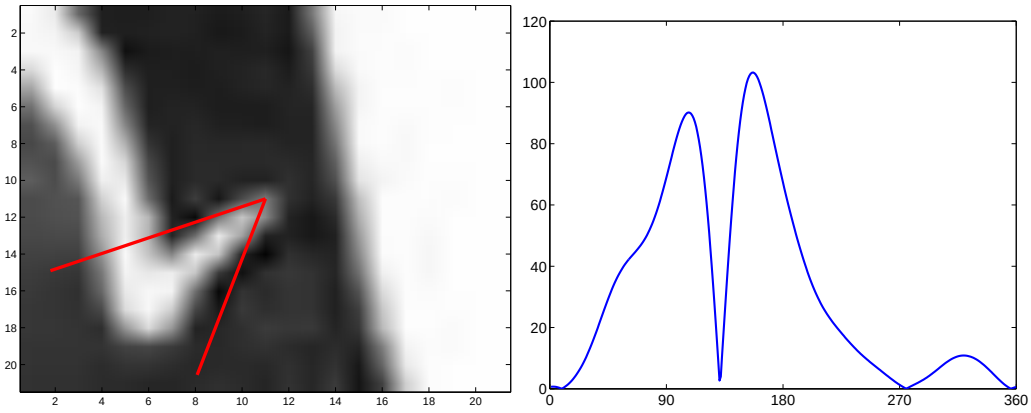


Figure 6.3: Example of an image (on the left) and the corresponding values of $\left|\frac{dI_\theta}{d\theta}\right|$ for θ in $[0°; 360°]$. On the left the red lines correspond to the directions of maximum variations (lobes of $\left|\frac{dI_\theta}{d\theta}\right|$).

the proposed method can follow them correctly, for a computational time equivalent to *JetStream* (the simulations were performed under Matlab). This comparison has been carried out without user interaction that is an essential tool in a contour tracing application. For real application, user interaction is actually inevitable. A further development of this algorithm could be also to automatically extract *all* relevant contours of a picture by letting branches to grow separately after edge junctions.

As a concluding remark, it is important to point out that first-time users of probabilistic contour tracers face an immediate difficulty: the impossibility of reproducing the same results. This inability is inherent to the use of particle filters and other MCMC methods, which are by essence stochastic. Not only results can not be reproduced, but also there is always a non-null probability that the tracking simply fails. In fact, it transpires from practical experience with contour delineation, that users prefer using deterministic approaches.

The problem is that finding an equivalent deterministic solution is very often difficult, if not impossible. In contrast with particle filtering, deterministic methods are indeed less flexible and thus less prone to complex modelling. This is actually one the greatest strength of MCMC methods. With MCMC methods it is very easy to implement complex models for the distribution, without having to think too much about *how* to infer the statistics. This is because the MCMC methods ensure that the space is sampled (or explored) in the best statistical way.

In this chapter however, the problem has already been simplified, and an optimal importance density could be established for the 1-dimensional posterior $p(\mathbf{x}_{0:t}|\mathbf{y})$. The posterior can then be inferred at a low computational cost. It stands to reason that instead of using particle filters to propagate the posterior, it would be preferable to use deterministic variants of the Kalman Filter. This would probably simplify the user interaction, but this is left for further research.

Following this idea, the object of the next chapter is to point out that in many situations, it is actually possible to simplify the problem in such a way that the use of particle filters can be efficiently replaced by the Viterbi algorithm.

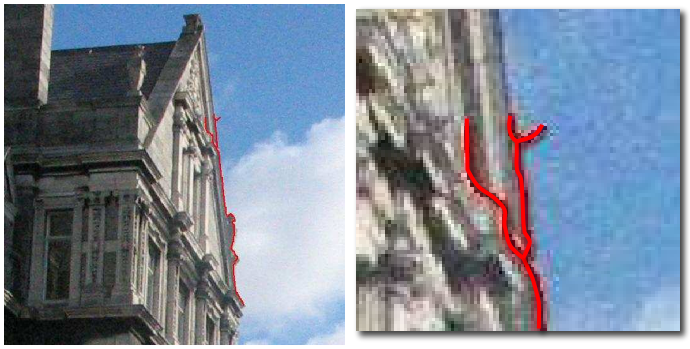


Figure 6.4: Example of the Oriented Particle Spray in action, with on the right a zoom on the multiple hypotheses tracking.

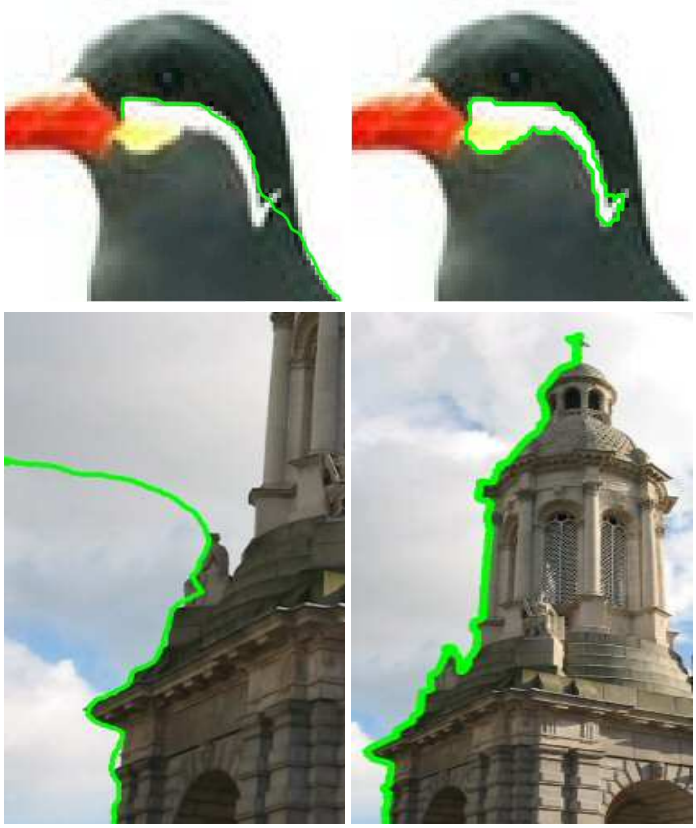


Figure 6.5: Contour tracings for *JetStream* on the left column and the Oriented Particle Spray on the right.

Chapter 7

Off-line Multiple Object Tracking using Candidate Selection and the Viterbi Algorithm

This chapter presents a probabilistic framework for off-line multiple object tracking. At each timestep, a small set of deterministic *candidates* is generated which is guaranteed to contain the correct solution. Tracking an object within video then becomes possible using the Viterbi algorithm. In contrast with particle filter methods where candidates are numerous and random, the proposed algorithm involves a few candidates and results in a deterministic solution. Moreover, only off-line applications where past and future information is exploited. This chapter shows that, although basic and very simple, this candidate selection allows the solution of many tracking problems in different real-world applications and offers a good alternative to particle filter methods for off-line applications.

7.1 Introduction

Tracking visual objects in image sequences is a key task for a wide range of applications in different domains (traffic surveillance, video summarisation, etc.). It has been extensively studied and many methods have been proposed. Particle filter based methods have become very popular indeed. They are powerful, simple and can handle complex situations in particular multiple objects tracking [Hue02, Oku04]. They are specially suitable for applications where on-line processing is required. In many such applications past information is used to determine the current position of the tracked object(s).

parts of this chapter are based on the work published in [Pit05b]

The First Key Idea in this chapter is to acknowledge that in many applications, tracking could be performed off-line. Information retrieval in sport footage [ici03] and video indexing are typical examples of such applications. In this context, a global analysis of the video can be performed to extract object paths, that is, visual features are first extracted from all the frames and then analysed in a second step. In such a scenario exploiting both the past and future information could lead to useful gains.

The Second Key Idea in this chapter is to consider the possibility of generating at each timestep, a candidate set of solutions that is guaranteed to contain at least one solution that is *correct*. In that case a deterministic process can yield the MAP estimate for tracking. This may seem wishful, yet it is worthwhile to consider this alternative route to tracking because such simple scenarios do indeed exist and can arise from realistic problems. Given the difficulties posed by the correct application of particle filters, in particular the problem of degeneracy of particles, it is useful to consider alternative strategies where those are viable. The success of this approach depends entirely on the process for generating candidates, it must be simple, and reliable enough that the candidates always contain the *correct* state. It is interesting to note that Kernel Particle Filter in [Cha03] have introduced the notion of pre-processing as a means of improving particle diversity in the particle filter for a tracking problem. The reader can consider that this chapter takes that idea one step further and proposes that if the candidate selection stage is reliable enough (which it can be) sampling can be avoided.

Organisation of the chapter. An overview of the methodology for off-line object tracking is presented in section 2. It is explained how, by defining a suitable candidate selection and a set transition probabilities, tracking an object within the video becomes equivalent to finding the most likely path in the candidate trellis using the Viterbi algorithm [The89].

Although basic and very simple the candidate selection process allows the solution of many tracking problems in different real-world applications. It also allows the easy integration of specific rules related to the object motion. The chapter presents in sections 3 and 4 two applications that represent domains in which tracking is amenable to this kind of idea.

The first application considered aims at detecting the arms of a child in a psychological assessment exercise. Tracking is used only to take temporal information into account and avoid false detections due to occlusions. It is a simple application that allows to introduce the framework. The second application is more challenging and concerns player detection and tracking in soccer video footages. Problems of introduction of a new players in the scene, disappearance of a tracked player and occlusions have to be dealt with.

7.2 Overview of the Methodology

Consider that $\mathbf{x}_n$ is the random variable corresponding to the object position $\mathbf{x}$ (which may be 1 or 2D depending on the application), where $n \in [1; N]$, and N is the number of frames of the sequence. Bayes theorem states that the *posterior* distribution of the object position throughout the sequence $\mathbf{x}_{1:N}$ can be written as

$$p(\mathbf{x}_{1:N}|\mathbf{y}_{1:N}) \propto p(\mathbf{x}_{1:N})\ p(\mathbf{y}_{1:N}|\mathbf{x}_{1:N}) \tag{7.1}$$

where $p(\mathbf{x}_{1:N})$ corresponds to the *prior* on the object positions and $p(\mathbf{y}_{1:N}|\mathbf{x}_{1:N})$ corresponds to the *likelihood* for the object positions given the data model $\mathbf{y}_{1:N}$—which corresponds here to the frames of the sequence.

Assume that the likelihood can be computed independently on each frame $p(\mathbf{y}_{1:N}|\mathbf{x}_{1:N}) = \prod_n p(\mathbf{y}_n|\mathbf{x}_n)$. In general there are a large number of possible states (each pixel location in each image, and in each frame). Reducing the number of states will reduce the computational load. The idea is to propose a limited number of states as *candidates* from some pre-process. One option is to generate these candidates as the peaks of the likelihood $p(\mathbf{y}_n|\mathbf{x}_n)$. The likelihood presenting r peaks is then approximated by the following grid-based distribution:

$$p(\mathbf{y}_n|\mathbf{x}_n) \propto \sum_{i=1}^{r} p(\mathbf{y}_n|\mathbf{x}_n^{(i)})\ \delta(\mathbf{x}_n^{(i)} - \mathbf{x}_n) \tag{7.2}$$

The candidates solutions of the tracking follow some *rules* depending on the application (feasible moves, scenarios of occlusions, ...). These rules are encapsulated in the prior function which gives the transition probabilities between candidates. Then from the rules and the candidates Viterbi [The89] can be applied to extract the most likely path, which is actually a Maximum a Posteriori estimation. For more than one object, it is possible to apply iteratively Viterbi and remove the corresponding candidates from the set of candidates. The *posterior* of the successive tracks is decreasing and the number of objects to track can be automatically determined by thresholding the *posterior*.

The success of the method depends on the simplicity of the object detector which is performed on the whole picture. A few particle filter trackers [Ter04, Oku04] propose a similar approach by sampling part of the particles directly from the likelihood. For instance [Oku04] uses Adaboost to detect the entrance of new players.

A similar candidate refinement scheme has been used in Kernel Particle Filter [Cha03]. But here candidates are fully *deterministic* as the resulting tracking. This implies also that the proposed method requires much less candidates.

7.3 Application to a Simple Case Study

This section shows how the framework can be applied in a real application. The application aims at detecting the position of the hands of a child performing a psychological exercise [Joy04] as presented in figure 7.3.

Candidates are found by projecting the colour skin segmentation [Joy04] of the frames on the horizontal axis and taking the main peaks of the projection as candidates for the hands positions (see figure 7.3). The likelihood of these candidates is proportional to the value of the projection. The presence of the instructor can generate spurious peaks in the skin colour projection and up to the 5 most important peaks are selected.

Transition probabilities are set to prevent large displacements of the hands:

$$p(x_n|x_{n-1}) \sim \mathcal{N}(0,3) \tag{7.3}$$

Once the hand candidate positions have been collected, the Viterbi algorithm is applied to extract one hand trajectory. To track the other hand, it suffices to remove the candidates corresponding to the first track and then to apply again Viterbi on the reduced set of candidates. The figure 7.3 shows some example of results (see also [Pit05a]).

7.4 Application to Multiple Objects Tracking

This section proposes a more difficult tracking application: the tracking of soccer players. Many works have been published on the subject and one can refer for instance to [OK02] for an example of tracking based on particle filter method.

The following paragraphs describe a possible adaptation of the framework. In particular, it is necessary to explain how to extract candidate positions of the players and how to set the rules explaining the dynamic of the players.

7.4.1 Player Candidate Positions

Playground Extraction. The playground can be efficiently extracted using a colour segmentation of the pitch followed by simple morphological operations to fill in holes and remove spurious detections.

Player Detection. To detect the position of the players on the playground, it is relevant to use the colour as a feature to characterise players. As shown in figure 7.4 a colour segmentation will result in a map of blobs corresponding to the players. It is possible then using a Mean Shift procedure to extract the centre of mass of these blobs and locate the player. This method has to be related to the colour histogram based mean-shift techniques used in [Jaf03, Com03] which are known to be robust. The method is also fast, because the mean shift can be done on downsized pictures, and generates a small set of candidates

for each frame (typically less than 20 candidates).

7.4.2 Set of Rules

Once a set of position candidates has been obtained, it remains to encode allowable player motion,*i.e.* how it can appear or disappear from the field of view and how it can be occluded by other players.

Player Motion. The player motion can be realistically limited to 50km/h. Even though the frames are not registered, such a speed can be bounded by a player displacement of 50 pixels for a PAL resolution (720x576).

$$\begin{cases} p(\mathbf{x}_n|\mathbf{x}_{n-1}) = 0 \text{ if } \|\mathbf{x}_n - \mathbf{x}_{n-1}\| > 50 \\ p(\mathbf{x}_n|\mathbf{x}_{n-1}) = 1 \text{ if } \|\mathbf{x}_n - \mathbf{x}_{n-1}\| \leq 50 \end{cases} \tag{7.4}$$

Player Apparition/Disappearance. To allow apparition and disappearance of players at any time of the video, two abstract states positions $\mathbf{x}_0$ and $\mathbf{x}_\infty$ are added. $\mathbf{x}_0$ indicates that the player is not yet visible and $\mathbf{x}_\infty$ indicates that the player is not any more visible. It is assumed that a player cannot appear more than once, which means that a typical state sequence is of the form $(\mathbf{x}_0, \ldots, \mathbf{x}_0, \mathbf{x}_n, \cdots, \mathbf{x}_\infty, \ldots, \mathbf{x}_\infty)$. In particular it is not possible to have the state $\mathbf{x}_\infty$ followed by $\mathbf{x}_0$ and vice versa; this leads to the following transition probability:

$$\begin{cases} p(\mathbf{x}_n = \mathbf{x}_0|\mathbf{x}_{n-1} = \mathbf{x}_\infty) & = 0 \\ p(\mathbf{x}_n = \mathbf{x}_\infty|\mathbf{x}_{n-1} = \mathbf{x}_0) & = 0 \end{cases} \tag{7.5}$$

In this framework, a player that has disappeared cannot appear again. However it is possible to image a post-process that would assign different tracks to a single player.

The probability that a candidate corresponds to a player that has just appeared or is about to disappear is dependent on the distance of the candidate to the border of the frame. For instance, a candidate cannot appear more than 50 pixels away of the borders:

$$p(\mathbf{x}_n \notin \{\mathbf{x}_0, \mathbf{x}_\infty\}|\mathbf{x}_{n-1} = \mathbf{x}_0) = 0 \quad \text{if } \operatorname{dist_t o_b oders}(\mathbf{x}_n) > 50 \tag{7.6}$$

Occlusions by another Team-Mate. In this case the colour detection only spots one player, and this single candidate position corresponds to two different tracks. To overcome this problem, instead of removing candidates from the pool each time Viterbi has been run for a player, the candidate positions are kept but penalised by reducing their likelihood (division by 3). Penalising the previously selected candidates avoids to generate multiple instance of the same track but still allows for temporary overlap of the tracks.

Occlusions by an Opponent. If a player is occluded by an opponent, its colour is also occluded and the player cannot be detected. In this situation the candidate positions of

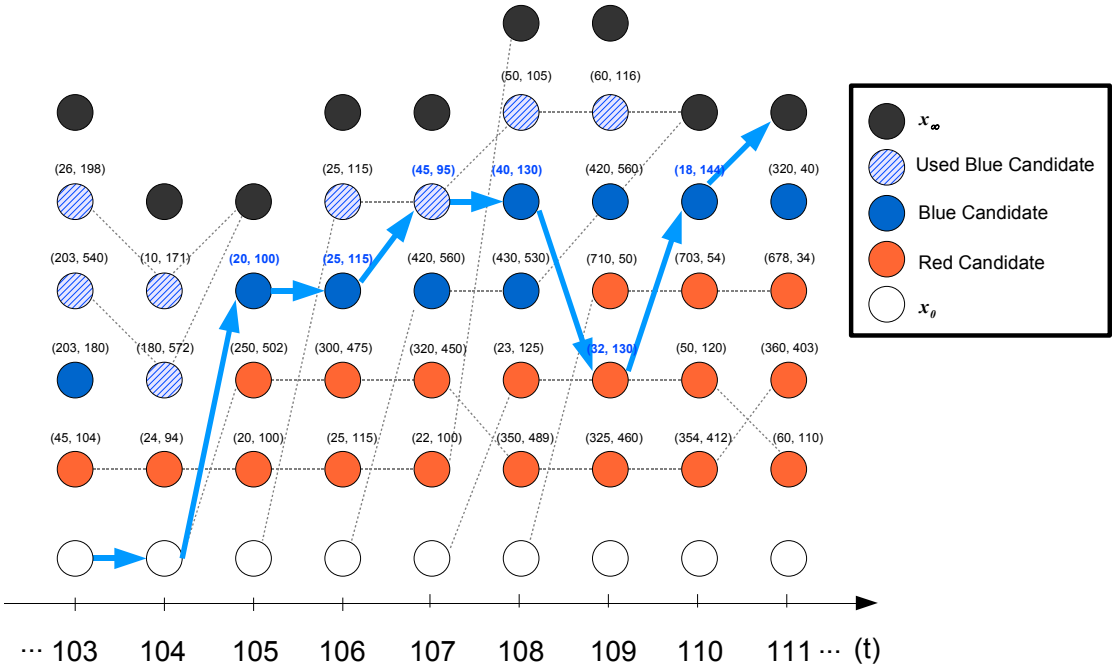
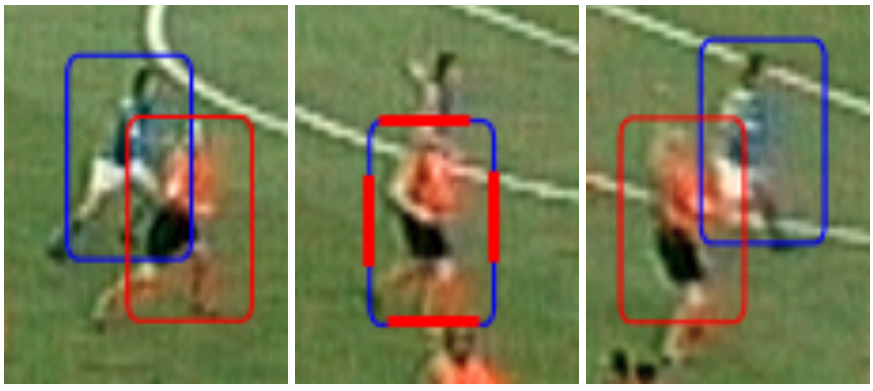


Figure 7.1: Proposed Bayesian Graph for tracking objects. The circles correspond to the states of the graph. At each instant (t) there can be a variable number of states. The white state on the bottom line is $\mathbf{x}_0$, and corresponds to an object that has not yet appeared. The black state on the top line is $\mathbf{x}_\infty$, and corresponds to an object that has not disappeared. The red and blue states correspond to the blue and red candidate positions. Finally the dashed blue states correspond to the blue candidates that have already been used for a previous tracking.



(a) (b) (c)

Figure 7.2: Example of player occlusion: the blue player on (b) is fully occluded. The tracking method assigns temporary the position of the red player to the blue one.

the opponent team is used instead. Practically the candidate positions of the other team is added to the current set of candidates, but with a much lower likelihood (division by 3). Figure 7.2 shows the results obtained for such a scenario. The blue player is not found on the middle frame and is temporary assigned to the position of the red player.

Post-Processing Rules. Since the model is based on a Markov Chain of order 1, post processing rules allow for integrating richer features to filter the results. One can decide if a player visible only on a few frames is worth being tracked. One can also set, as mentioned earlier in the chapter, a threshold for the *posterior*. If the *posterior* of the tracking is too small, the object is insignificant and the multitracking process stops.

Figure 7.1 shows a tracking example combining the situations that have just been described. The circles correspond to the states of the graph. Note that there can be a variable number of states for each frame (n). The white state on the bottom line is $\mathbf{x}_0$, and corresponds to an object that has not yet appeared. The black state on the top line is $\mathbf{x}_\infty$, and corresponds to an object that has not disappeared. The red and blue states correspond to the blue and red candidate positions. Finally the dashed blue states correspond to the blue candidates that have already been used for a previous tracking. The apparition happens on frame 105, and the player is close to the border ($x = 20 < 50$). The player is occluded by another team-mate on frame 107 and by an opponent on frame 109. The player leaves the field of view at frame 111.

Figure 7.6 shows some results for the tracking of the blue team and video material is available online at [Pit05a]. It is noteworthy that the results are obtained deterministically and can be reproduced identically, whereas with particle filter methods, the results are partly random and will differ slightly in applications to the same footage with the same initial conditions, with a non-zero possibility of outright failure in any given instance.

7.5 Conclusion

This chapter showed that in some applications, when the image data is such that the object detection task is quick and robust, 1) random candidate generation in particle filters can be efficiently replaced by a deterministic candidate selection that results in deterministic solutions and 2) for off-line applications the Viterbi algorithm can be applied to exploit all available temporal information.

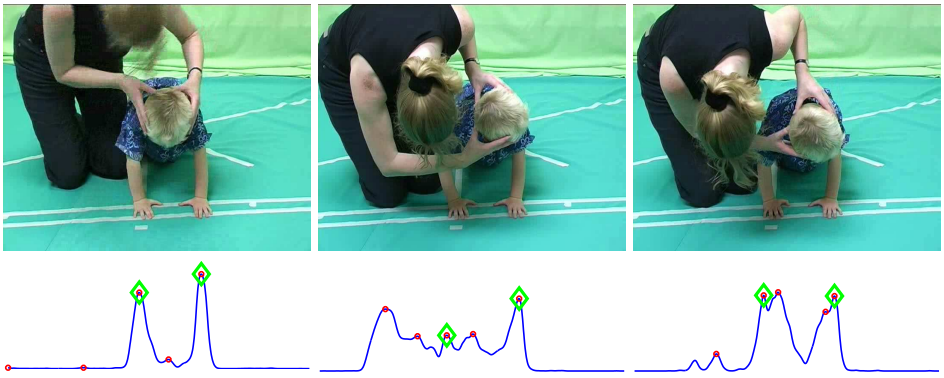


Figure 7.3: Detection of the hands positions (green diamonds) of a child performing a psychological exercise [Joy04]. The peaks of the skin colour projection give the candidate positions (red circles).

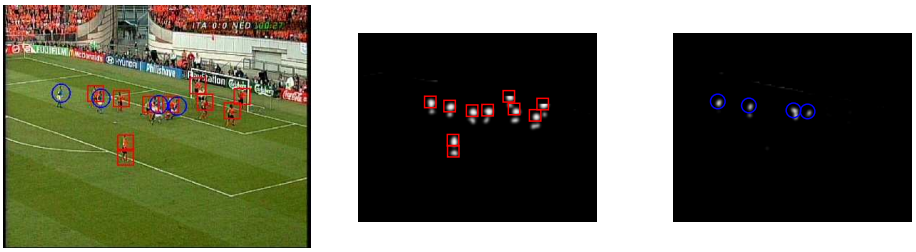


Figure 7.4: Example of Player Detection using colour segmentation and MeanShift to find the blobs centres.

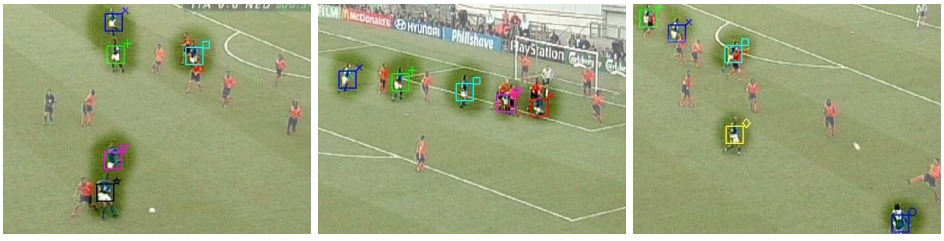


Figure 7.5: Tracking in action on the soccer sequence. Only the blue team is tracked, other areas are in lower contrast. Videos are available at [Pit05a]. (Image courtesy of Rádio e televisão Portugal)

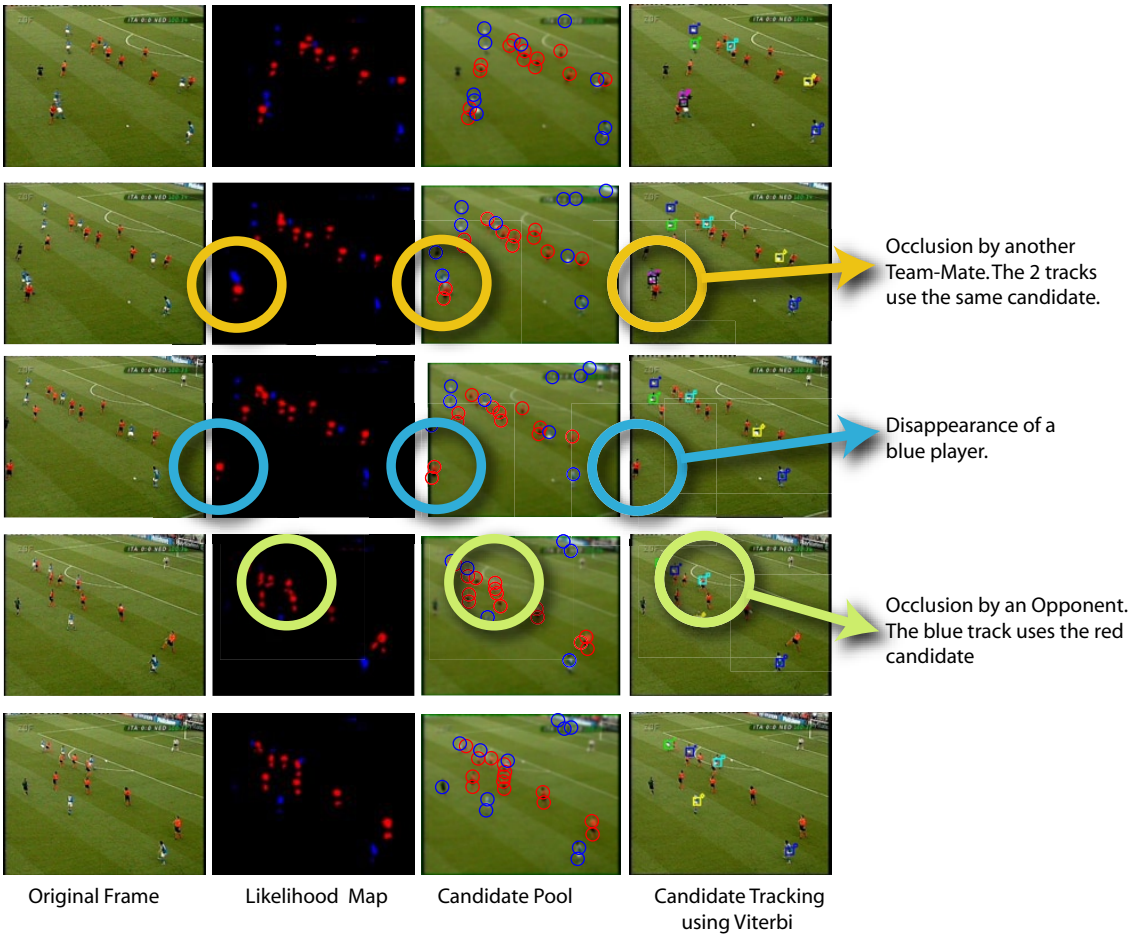


Figure 7.6: Tracking in action on the soccer sequence. Only the blue team is tracked, other areas are in lower contrast. Videos are available at [Pit05a]. (Images courtesy of Rádio e televisão Portugal)

Part III

Image Simplification

Chapter 8

Towards Image Simplification

Most tools in video post-production consist of altering the raw videos in a seamless way, such that the resulting video looks natural. This chapter explores the idea that videos could also be re-expressed in a different visual form that is not necessary close to reality, but which convey the same content. Rendering non-photographic pictures has raised some interest in the computer graphic community, especially to design filters [Mig03] that simulate an artistic style. The statistics transfer technique shown in the first chapters can be used for instance to render non-photographic images (see figures 2.1 and 2.2). However these approaches are still limited to low-level pixel processing. The idea proposed in this chapter is to push further this concept by designing a non-photographic manipulation that focuses on the *content* of the video, whilst simplifying its representation.

Cartoonisation for Video Broadcasting. With the rise of digital visual media, there is a demand for broadcasting videos on a wide variety of devices. But sadly this has not yet materialised. One aspect of the problem is the bandwidth issue: embedded devices usually have bandwidth limitations but video streaming requires at least 64KB/s, which is often impossible or just too expensive. Another aspect is the size of the display. If too small, the user does not discern objects in cluttered environment. For example in golf events broadcasting, the ball simply disappears in the resized movie. And if the display is too large, compression artefacts become apparent.

The original idea proposed in this chapter is to simplify the video by extracting the important information and re-expressing that in a different visual form. An attractive form is the *cartoon* representation as shown in figure 8.1. Several cartoon-like formats exist already and are widely used (for instance SVG and flash). They code the scene in terms of

layers, contours and animations of the layers. By converting the natural video into one of these content-aware video formats, it is then possible to enhance the viewability of the pictures by throwing away details while enhancing important features. These representations have also the benefit of being vector-based and thus scalable, which means that pictures can be displayed without any loss of quality on any screen. *Simplification* therefore, has two implications depending on the application. For streaming, *media simplification* admits a more efficient use of limited bandwidth while for display adaptation the concept allows for comfortable viewing on a wide range of terminal types.

Several works have been achieved to cartoonise videos. One recent attempt in the domain has been presented by [Wan04b, Wan04c]. The method is a semi-automated process, which includes in a first step an implicit video segmentation followed by a manual interaction to track contours across the video. Another recent effort [DeC02] in re-expressing images in a cartoon-like aspect makes use of hardware eye tracking system to find areas of interest and then enhance these areas whilst flattening details of the background. Note that these methods rely heavily on human interaction.

A fully automated approach has been explored in [Kok05]. The strategy adopted is to restrict the media content to deal with one application, like tennis or snooker for instance. It is then possible to develop robust mechanisms for extracting the useful information, like the position of the player, and expressing it in a different visual form that occupies an extremely low bandwidth. This kind of content aware media processing requires access to high level features, which can be extracted for instance by tracking objects as described in chapter 7.

What Is Simplification Anyway? The notion of simplification in this chapter refers to the process of replacing similar instances of an object by a unique representation of this object.

The idea of simplification is based on the observation that images and videos, like most data content, present highly redundant structures. For example, an object moving across multiple frames is likely to be almost identical on every frame. The idea is that the object on each of these frames actually correspond to the same unique object. A simplification of the video could be expressed in replacing all instances of this object by a unique representation of this object.

How is simplifying any different from compressing? These two concepts both reduce redundancy within the data stream. The key difference lies in the objective of these processes. In compression, removing redundant structures aims at reducing the description length of the data stream, but the main point is that the resulting video/image should still remain as close as possible to the original raw stream. In simplification, the output of the image or video is not required to be unchanged, but instead, it is only required to leave the interpretation of the *content* unchanged.

Approach. It is clear that with manual intervention, simplification can be achieved with impressive results [Wan04c, DeC02]. However, this work explores the notion that the

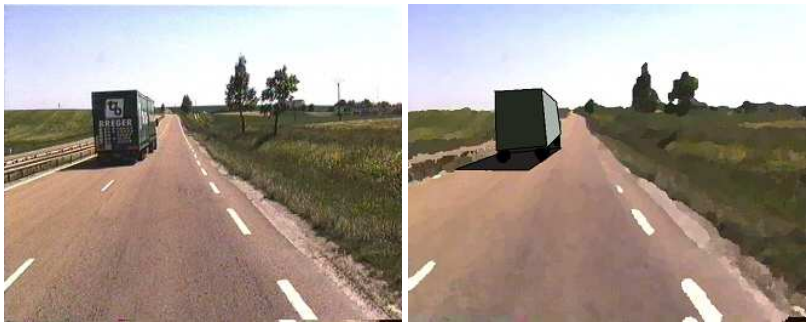


Figure 8.1: Example of the video cartoonisation concept for traffic condition surveillance footages.

intrinsic statistical properties of image patches can be exploited for this purpose. The reason for doing this to avoid the heavy contextual knowledge required for pre-processing tasks like segmentation. This idea is called 'implicit Statistical Simplification'.

8.1 Implicit Statistical Simplification

The idea of Implicit Statistical Simplification is to avoid explicit statement of the 'object finding' problem and instead exploit intrinsic, clusterable features of images. This leads to perceptually simplified images of the type required.

Denote $S = \{\mathbf{u}_i\}_{i \leq M}$ as the set of the observed samples and denote c_i as the cluster grouping the samples similar to the sample of position i. By definition each sample of the cluster c_i points to different instances of the same generic object. That object is represented here by $\overline{\mathbf{u}}_i$. Note that the representation object $\overline{\mathbf{u}}_i$ is not unique and could be defined for instance as the average value of the cluster. The set of these representative elements that constitutes the whole image is denoted as $\overline{S} = \{\overline{\mathbf{u}}_i\}_{i \leq M}$.

Finding the clusters corresponds to the non trivial problem of segmentation. One solution is to reduce the sample pdf to a sum of well chosen Dirac's that correspond to the clusters and then assign samples to these classes. The reduction step can be understood by the 1D case proposed in figure 8.3. The pdf $p(\mathbf{u})$ of the observed signal (on the left, in blue) presents two peaks, which are reduced into a sum of two Dirac's $p(\overline{\mathbf{u}})$. A attractive implicit solution for finding these Dirac's is to use the MeanShift filtering segmentation technique. MeanShift filtering is a clustering method originally developed by Fukunaga [Fuk90] and recently successfully applied to computer vision by Cheng [Che95] and then by Comaniciu and Meer [Com02]. It consists of labelling each sample by the nearest peak in the pdf. The peaks are found by gradient ascent on the density estimation function and the resulting clusters are then delimited by valleys of point densities. This results in an implicit framework where the number of classes is not predefined.

8.2 MeanShift Filtering

For a set of N-dimensional samples $S = \{\mathbf{u}_j\}_{j=1...M}$, the corresponding fixed bandwidth kernel density estimate is defined by:

$$\hat{p}(\mathbf{u}) = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{h^N}K\left(\frac{\mathbf{u}-\mathbf{u}_j}{h}\right) \tag{8.1}$$

where $K(\mathbf{u}) = k(\|\mathbf{u}\|^2)$ represents the kernel and h is the bandwidth controlling the smoothness of the approximation. A typical choice for the kernel is the Epanechnikov kernel (see chapter 2, figure 2.8). The segmentation aims at finding for each sample the nearest local maxima in the pdf. The direction of the nearest maxima can be given by the *sample mean shift*, referred to as MeanShift and denoted as $M_h(\mathbf{u})$:

$$M_h(\mathbf{u}) = \frac{\sum_{j=1}^{M}\mathbf{u}_j\ g\left(\left\|\frac{\mathbf{u}-\mathbf{u}_j}{h}\right\|^2\right)}{\sum_{j=1}^{M} g\left(\left\|\frac{\mathbf{u}-\mathbf{u}_j}{h}\right\|^2\right)} - \mathbf{u} \tag{8.2}$$

where $g(u) = -k'(u)$. Then by observing that g is also a kernel function, the MeanShift can be shown to be collinear to the gradient of the pdf:

$$M_h(\mathbf{u}) = \frac{h^2}{N+2}\frac{\nabla\hat{p}(\mathbf{u})}{\hat{p}(\mathbf{u})} \tag{8.3}$$

The nearest peak is then found by moving iteratively the sample by the MeanShift vector $M_h(\mathbf{u})$. For a sample at position i, the procedure is initialised at $\mathbf{u}^{(0)} = \mathbf{u}_i$ and the iterations for the Epanechnikov kernel can be simplified as follows:

$$\overline{\mathbf{u}}_i^{(n+1)} = \frac{1}{|\mathcal{C}_h(\overline{\mathbf{u}}_i^{(n)})|}\sum_{j\in\mathcal{C}_h(\mathbf{u}_i^{(n)})}\mathbf{u}_j \tag{8.4}$$

where $\mathcal{C}_h(\overline{\mathbf{u}}_i^{(n)})$ is the hypersphere of radius h and centred on $\overline{\mathbf{u}}_i^{(n)}$, and containing $|\mathcal{C}_h(\overline{\mathbf{u}}_i^{(n)})|$ data samples. In other words, the iteration consists in iteratively moving to the centre of mass of the neighbouring samples. Figure 8.2 illustrates the MeanShift procedure. At convergence, $\overline{\mathbf{u}}_i^{(n)}$ reaches the nearest peak of the pdf [Com02]:

$$\lim_{n\to\infty}\overline{\mathbf{u}}_i^{(n)} = \overline{\mathbf{u}}_i \tag{8.5}$$

Figure 8.3 shows how this could lead to signal simplification. Each observed 'noisy' value is replaced with the corresponding cluster centre. The signal is then effectively simplified into two values 0 and 1. A limitation of the method is that it is impossible to recover signal granularities that are smaller than twice the noise standard deviation

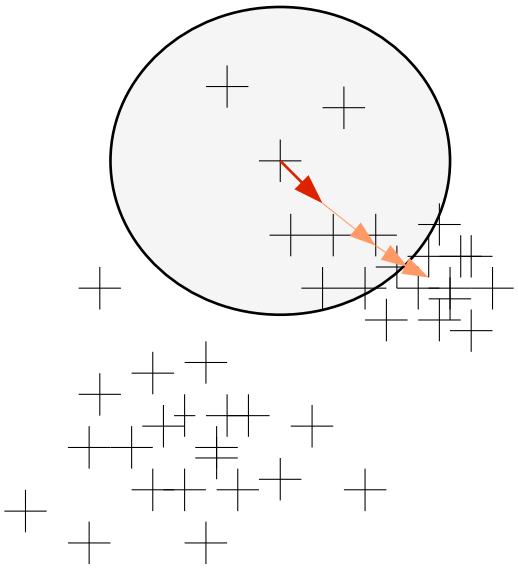


Figure 8.2: MeanShift iterations. The circle delineates $\mathcal{C}_h(\mathbf{u})$, the hypersphere of radius h and centred on $\mathbf{u}$. The red arrow represents the MeanShift vector that moves the sample to the centre of mass of the neighbouring samples. The orange arrows represent the MeanShift moves for the following iterations.

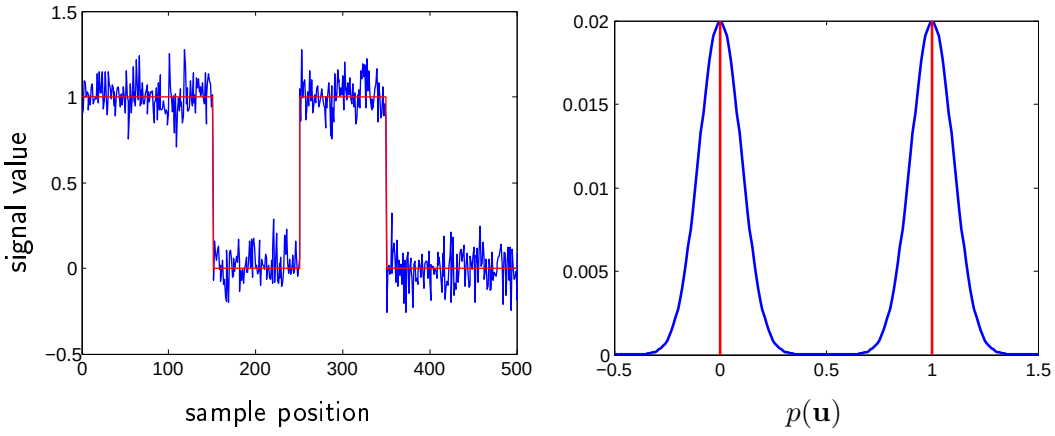


Figure 8.3: Example of Simplification using MeanShift Filtering. The pdf of the signal values is reduced to two Dirac's centred in 0 and 1.

σ. This is illustrated on figure 8.4 for $\sigma = .5$. In this case the pdf (on the right) only presents one peak centred in $\mathbf{u} = .5$ and the resulting filtered signal (in red on the left) is constant. This is a serious drawback which is partly fixed in [Com02] by introducing the spacial information in the segmentation. The idea proposed in [Com02] is to take the sample position as a feature of the sample. In their case, a pixel sample contains then 5 components: 3 colour components and 2 spatial coordinates. This means that both the colour feature and the spatial coordinates are moving in the MeanShift update.

The sample position is an important information that ought to be taken into account. It is indeed widely accepted in image processing that pictures are piecewise smooth and that using spatial smoothness priors for the segmentation is key to obtain natural looking results. However the results of taking the position as a feature as in [Com02] are unclear. Consider for example the signal on figure 8.5. It transpires from this example that similar samples are not necessary spatially close to each other, and that the centre of mass of the coordinates is not significant in this case. It is thus necessary to find a way to integrate the spatial smoothness in the MeanShift segmentation, by considering the sample position as a *parameter* and not as a feature.

8.3 Smoothness Prior

Generic Prior. Smoothness priors are usually used as generic priors that would be applicable for any image. One example of such a prior is the Ising model that simply penalises discontinuous labelling. For neighbouring samples i and j, the Ising prior is as follows:

$$-\ln p\left(\overline{\mathbf{u}}_i|\overline{\mathbf{u}}_j, i, j\right) = \lambda\ \delta(\overline{\mathbf{u}}_i \neq \overline{\mathbf{u}}_j) \tag{8.6}$$

This gives a local smoothness to the labelling whose strength can be tuned with λ. Note that for a continuous framework, as in Gaussian Markov Random Fields, the Ising model is equivalent to

$$-\ln p\left(\overline{\mathbf{u}}_i|\overline{\mathbf{u}}_j, i, j\right) = \lambda\ \|\overline{\mathbf{u}}_i - \overline{\mathbf{u}}_j\|^2 \tag{8.7}$$

It has been proved however [Des95, Mor96] that the Ising model alone performs poorly for segmentation purposes. In fact, establishing a smoothness prior that would be efficient for any kind of images turns out to be a very difficult task. This is due to the high dimensionality, complexity and variety of images. A recent advance in this domain is the Fields of Experts framework (FoE) [Rot05a] which proposes generic prior model for natural images. The prior is trained on a database of generic images but can also be tuned by using more pertinent databases [Rot05b].

Image Specific Prior. To relax the difficulty of modelling a generic prior, another solution is to link the label interactions with the features observed for this particular picture, *i.e.* to consider $p\left(\overline{\mathbf{u}}_i|\overline{\mathbf{u}}_j, i, j, \mathbf{u}_i, \mathbf{u}_j\right)$ instead of $p\left(\overline{\mathbf{u}}_i|\overline{\mathbf{u}}_j, i, j\right)$. Label discontinuities happen usually

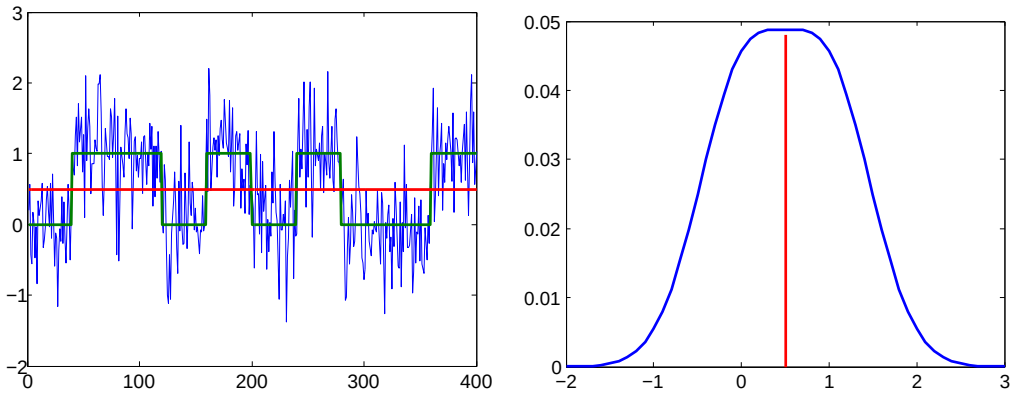


Figure 8.4: The granularity of the MeanShift Filtering output is limited by the noise level. For $\sigma = .5$, the pdf displays only one peak.

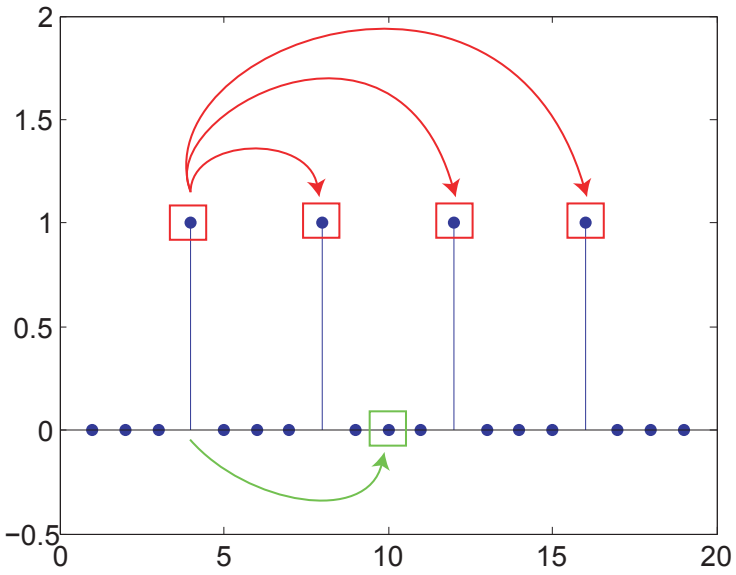


Figure 8.5: Example of using coordinates as features. The signal is in blue. The cluster of sample 4 contains the samples $\{4, 8, 12, 16\}$ (highlighted with red boxes). The centre of mass for these sample coordinates is 10, but 10 does not belong to the cluster.

at feature discontinuities (*e.g.* at object edges), and reciprocally, label discontinuities are less likely on homogeneous areas. This is well known and previously Geman and Geman [Gem84] proposed in their seminal paper to use two Markov Random Fields (MRF): one for the image itself (called *image process*) and an additional MRF that notifies the presence of edges (*line-process*) and turns off or on the smoothness constraint accordingly.

In practical cases, this issue can be simplified by extracting the edge map from the original noisy/non-segmented image and thus avoiding the joint estimation. The smoothing term then purely depends on the observed data. For instance [Kok98, chapter 2] and [Bla04, Rot04, Boy04] propose to use a prior of the form:

$$-\ln p\left(\overline{\mathbf{u}}_i|\overline{\mathbf{u}}_j, i, j, \mathbf{u}_i, \mathbf{u}_j\right) = \lambda\ \delta(\overline{\mathbf{u}}_i \neq \overline{\mathbf{u}}_j)\|\mathbf{u}_i - \mathbf{u}_j\|^2 \tag{8.8}$$

Implicit Prior. Using this image specific prior instead of the generic Ising model can improve dramatically the segmentation. The method proposed in this chapter pushes further this idea by considering a purely implicit prior. The smoothness is uniquely inferred by analysing the patch statistics of the image itself, in a similar way as the work of Efros *et al.* [Efr99]. Consider the signal on figure 8.6. The signal (in blue) is composed of a periodic pattern. This gives a prior information that could be used in the segmentation. On this figure for example, the representation of the segmented signal is a repetition of the averaged pattern. This leads to clearly simplified representation. The proposed method is thus based on grouping samples that present similar neighbourhood.

8.4 MeanShift with Implicit Prior

The core of the method relies on the MeanShift algorithm. The key deviation from the standard MeanShift is that instead of finding the peaks of $p(\mathbf{u})$, it is desired to restrict the pdf to the pdf of samples that have similar neighbourhood $\mathcal{N}$. The method is thus to find the peaks of $p(\mathbf{u}_i = \mathbf{u}|\mathcal{N}_i)$ as illustrated on figure 8.7.

$$\begin{cases} \forall\ i\ ,\ \overline{\mathbf{u}}_i^{(0)} := \mathbf{u}_i \\ \\ \overline{\mathbf{u}}_i^{(n+1)} := \dfrac{\sum_j \mathbf{u}_j\ p\left(\mathcal{N}_i|\mathcal{N}_j\right)\ K\left(\frac{\mathbf{u}_j - \overline{\mathbf{u}}_i^{(n)}}{h}\right)}{\sum_j p\left(\mathcal{N}_i|\mathcal{N}_j\right)\ K\left(\frac{\mathbf{u}_j - \overline{\mathbf{u}}_i^{(n)}}{h}\right)} \end{cases} \tag{8.9}$$

where $p\left(\mathcal{N}_i|\mathcal{N}_j\right)$ gives a similarity probability between neighbourhoods. The neighbourhoods $\mathcal{N}_j$ collect the neighbouring samples of position j in S, *excluding* $\mathbf{u}_j$ itself, and $\mathcal{N}_i$ collects the neighbourhood of sample position i, excluding $\mathbf{u}_i$ itself. Since the notion of neighbourhood is only defined for a particular size, it should be developed according to the

neighbourhood diameter ρ:

$$p\left(\mathcal{N}_i|\mathcal{N}_j\right) = \int_{\rho>1} p\left(\mathcal{N}_{i,\rho}|\mathcal{N}_{j,\rho},\rho\right) p\left(\rho\right) d\rho \tag{8.10}$$

1) The first term $p\left(\mathcal{N}_{i,\rho}|\mathcal{N}_{j,\rho},\rho\right)$ corresponds to the probability that for a diameter ρ, both neighbourhood $\mathcal{N}_{j,\rho}$ of sample j and neighbourhood $\mathcal{N}_{i,\rho}$ of sample i are similar. Consider that both $\mathcal{N}_{i,\rho}$ and $\mathcal{N}_{j,\rho}$ are blocks of diameter ρ, encapsulating the neighbouring samples minus the sample itself. If the samples lie on a lattice of dimension L (*e.g.* $L=2$ for images), then these blocks contain roughly $m_\rho = \rho^L - 1$ samples. The similarity between neighbourhoods can be measured by estimating the standard deviation of the neighbourhood difference:

$$SD\left(\mathcal{N}_{i,\rho},\mathcal{N}_{j,\rho}\right) = \sqrt{\frac{1}{N\ m_\rho} \sum_{(k,l)\in\mathcal{N}_{i,\rho}\times\mathcal{N}_{j,\rho}} \|\mathbf{u}_k - \mathbf{u}_l\|^2} \tag{8.11}$$

Consider that the observed signal is corrupted by a white noise of standard deviation σ. The neighbourhood differences results then in adding up the noise. Thus if $i \neq j$, the standard deviation of the noise on the neighbourhood difference should be $\sqrt{2}\sigma$. Ideally then, neighbourhoods could be considered as similar if $SD \leq \sqrt{2}\sigma$. The SD estimate depends however on the number of values involved m_ρ. In particular, for a standard deviation of $\sqrt{2}\sigma$, the SD estimate should be compared to σ_ρ [Weib]:

$$\sigma_\rho = \sqrt{2}\sigma\sqrt{\frac{m_\rho - 1}{m_\rho}} \tag{8.12}$$

and the standard deviation of the SD estimate is given [Weib] by:

$$\lambda_\rho = \mathrm{Std}\left[SD\right] \approx \sqrt{\frac{\left(\sqrt{2}\sigma\right)^2}{2m_\rho}} \tag{8.13}$$

The similarity between blocks should then be measured by testing that $SD \leq \sigma_\rho$. This binary test can be relaxed by using a sigmoid function:

$$p\left(\mathcal{N}_{i,\rho}|\mathcal{N}_{j,\rho},\rho\right) \approx \frac{1}{\mathcal{L}_0}\mathcal{L}\left(\frac{SD\left(\mathcal{N}_{i,\rho},\mathcal{N}_{j,\rho}\right) - \sigma_\rho}{\lambda_\rho/2}\right) \tag{8.14}$$

where $\mathcal{L}(u) = \exp(-u)/(1+\exp(-u)) = 1 - \mathrm{sig}(u)$. The normalising factor $\mathcal{L}_0$ is:

$$\mathcal{L}_0 = \frac{\lambda_\rho}{2}\ \ln\left(1 + \exp\left(\frac{\sigma_\rho}{\lambda_\rho/2}\right)\right) \tag{8.15}$$

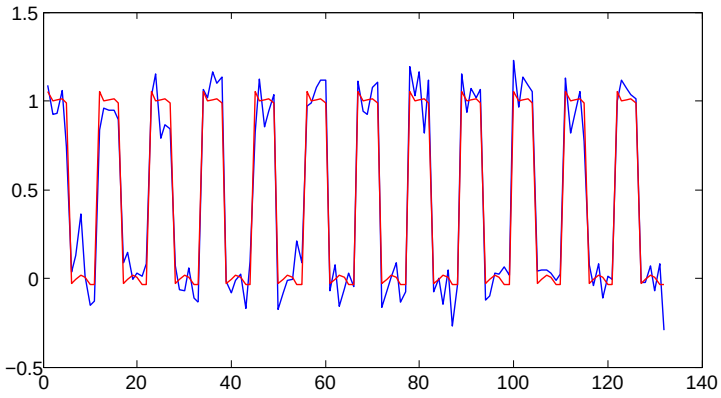


Figure 8.6: Example of Simplification by averaging repeated blocks.

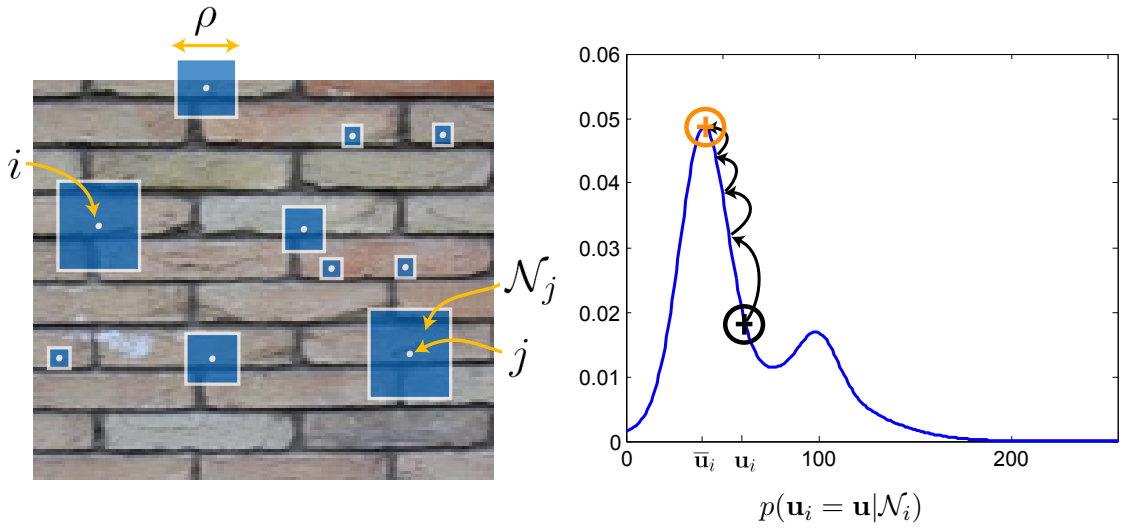


Figure 8.7: Implicit Statistical Simplification Outline. On the left: compare the neighbourhood of sample i to the neighbourhoods of other samples j, for different neighbourhood diameters ρ. On the right: $p(\mathbf{u}_i = \mathbf{u}|\mathcal{N}_i)$, the pdf of samples j that have similar neighbourhoods to i. The representative element $\bar{\mathbf{u}}_i$ is then found by MeanShift iterations on $p(\mathbf{u}_i = \mathbf{u}|\mathcal{N}_i)$.

2) The second element $p(\rho)$ weights the similarity between neighbourhoods with respect to the size of the neighbourhood. There is no prior knowledge on the size to expect, but it is still possible to assume that the problem is scale invariant, which yields in

$$p\left(\rho\right) \propto 1 \tag{8.16}$$

To avoid exploring all values of ρ, a significant speed up can be gained by considering powers of the scale, *i.e.* instead of summing for $\rho = 1, 2, 3, 4, 5 \ldots$, it is explored $\rho' = \exp_2(\rho)$, *i.e.* $\rho' = 1, 2, 4, 8, 16, 32, \ldots$. Bearing in mind that $d\rho' = d\exp(\rho) = \rho' d\rho$, the probability becomes:

$$p'(\rho') \propto \rho' \tag{8.17}$$

Implementation. The overall algorithm is detailed on page 112. The first consists in finding the neighbourhood similarities. The step is the MeanShift itself. The implementation is straightforward and can be found in the appendix A.3. Some special care is however required when manipulating neighbourhoods. The solution adopted here is to loop over the possible displacements $\mathbf{d}$, where the range of displacements can be limited to $\max() = D$. The SD estimate for a neighbourhood diameter of ρ is then obtained by convolution with a spatial filter which defines the neighbourhood:

$$SD(\mathbf{x}, \rho)^2 = G_\rho^{-0} * \left(S * \delta_\mathbf{d} - S\right)^2 \tag{8.18}$$

The Dirac distribution $\delta_\mathbf{d}$ shifts the signal S by $\mathbf{d}$. In the following results (in 1D), the spatial shape of the neighbourhood is as follows:

$$G_\rho^{-0}(\mathbf{x}) \propto \begin{cases} 0 & \text{if } \mathbf{x} = 0 \\ \exp\left(-\mathbf{x}^2/(2\rho^2)\right) & \text{else} \end{cases} \tag{8.19}$$

Because the filter is not separable, it is computationally expensive to apply it for images. A solution is to decompose the estimation as follows:

$$SD(\mathbf{x}, \rho)^2 = \left(G_\rho * \left(S * \delta_\mathbf{d} - S\right)^2 - G_\rho(0)\left(S * \delta_\mathbf{d} - S\right)^2\right) / (1 - G_\rho(0)) \tag{8.20}$$

with $G_\rho(\mathbf{x}) \propto \exp\left(-\mathbf{x}^2/(2\rho^2)\right)$ being separable.

The overall algorithm leads of course to ridiculously slow computations since it has to compare the signal with itself for each sample ($\mathcal{O}(M^3)$, for M samples). A 720x576 picture requires for instance around 8 minutes of processing on a PIII 800MHz for displacements up to 10 pixels. It is however hopeful that a GPU implementations could be very effective for such a parallelisable process.

8.5 Results

Figures 8.8 and 8.9 show results for 1D signals. The comparison of the method to standard MeanShift in figure 8.8) shows that the method in contrary to MeanShift can retrieve signal granularities that are smaller than 2σ. The strength of using an implicit prior is apparent in figure 8.9. The clean signal defined as $s(n) = (-1)^n \sin(n/20)$ is extremely non continuous. The method is however successful in making the most of the signal intrinsic redundancy.

Figure 8.10 shows the results of the method on images. Several values for σ and h are tested. It transpires that the bandwidth h used in the MeanShift step is not as influential as in standard MeanShift. This means that the prior distribution $p(\mathbf{u}_i = \mathbf{u}|\mathcal{N}_i)$ is mainly mono-modal, and that the neighbourhood information alone is very pertinent.

Figure 8.11 shows some results in colour. Note that the geometrical structure is particularly well restored. Figure 8.12 and 8.13 show results for traffic images. An important point to be made here is that this method is *not* a noise reducer but only aims at simplifying the picture. Hence evaluating the PSNR would not make any sense here. However it is interesting to measure the compression improvements, *i.e.* in this case the ratio between the original JPEG compression (with a quality set to 75) of the result image and the original image.

8.6 Conclusion

This chapter proposes a way of simplifying signals. The simplification aims at replacing similar objects by only one representation of this object. The algorithm that is proposed is a modification of the MeanShift algorithm, where a similarity measure between sample neighbourhoods weights the importance of samples in the MeanShift update. This way the use of neighbourhood statistics leads to a purely implicit formulation of the signal prior. Results for 1D signal and images show that the method is indeed very effective at simplifying a wide variety of signals.

The approach of this chapter is to see to what extend a purely implicit prior can help the segmentation. It would be however useful to integrate some generic parametric prior in the model. This would help in cases where the number of samples is too small and when a good model for the signal can be established beforehand. One improvement could be for instance to consider a Wavelet decomposition of the Signal and force the smoothness constraint. As shown for texture synthesis [Gal05], this would also speed up the search for similar neighbourhoods as the Wavelet decomposition is scale independent.

Algorithm 5 MeanShift with Implicit Prior

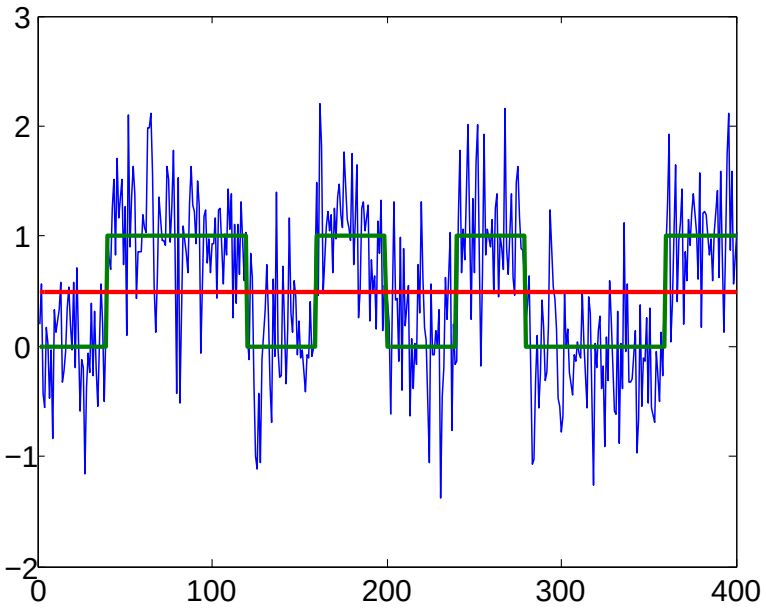
Inputs: σ, h and S

// **Initialisation**

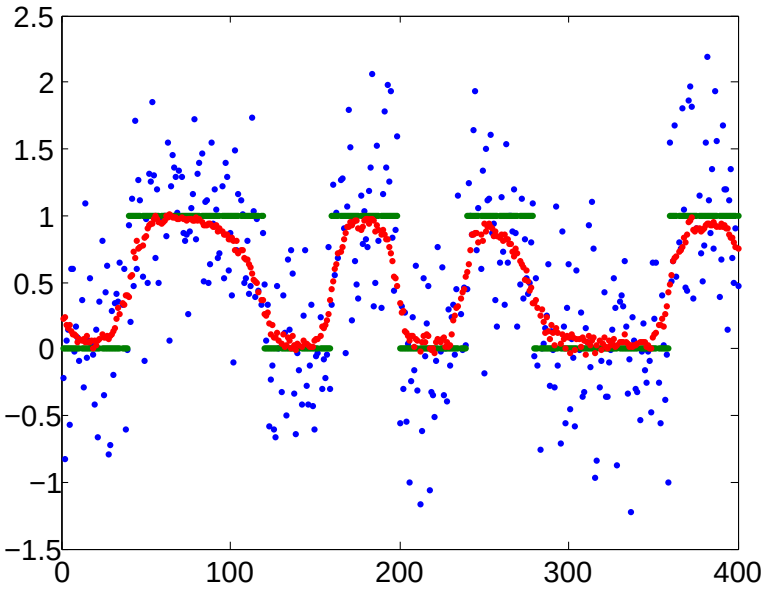
for all samples i **do**
  for all samples j **do**
    $p_{ji} \leftarrow 0$
    for neighbourhood diameter $\rho = 1, 2, 4, 8, \ldots$ **do**
      $m_\rho \leftarrow \rho^L - 1$
      $\sigma_\rho \leftarrow \sqrt{2}\sigma\sqrt{(m_\rho - 1)/m_\rho}$
      $\lambda_\rho \leftarrow \sqrt{2}\sigma/\sqrt{2m_\rho}$
      $\mathcal{L}_0 \leftarrow \lambda_\rho/2 \ \ln\left(1 + \exp\left(\frac{\sigma_\rho}{\lambda_\rho/2}\right)\right)$
      $p_{ji} \leftarrow p_{ji} + \rho \cdot \frac{1}{\mathcal{L}_0}\mathcal{L}\left(\frac{SD(\mathcal{N}_{i,\rho} - \mathcal{N}_{j,\rho}) - \sigma_\rho}{\lambda_\rho}\right)$
    end for
  end for
end for

// **MeanShift**

repeat
  $n \leftarrow 0$, $\overline{S}^{(0)} = S$
  for all samples i **do**
    for all samples j **do**
      $k_{ji} \leftarrow K\left(\frac{\mathbf{u}_j - \overline{\mathbf{u}}_i^{(n)}}{h}\right)$
    end for
    $\overline{\mathbf{u}}_i^{(n+1)} \leftarrow \left(\sum_j \mathbf{u}_j \ p_{ji} \ k_{ji}\right) / \left(\sum_j \ p_{ji} \ k_{ji}\right)$
  end for
until convergence



(a) MeanShift results (in red)



(b) MeanShift with Implicit Prior results (in red)

Figure 8.8: MeanShift with Implicit Prior. Results are in red, the observed noisy signal in blue and the clean original signal in green.

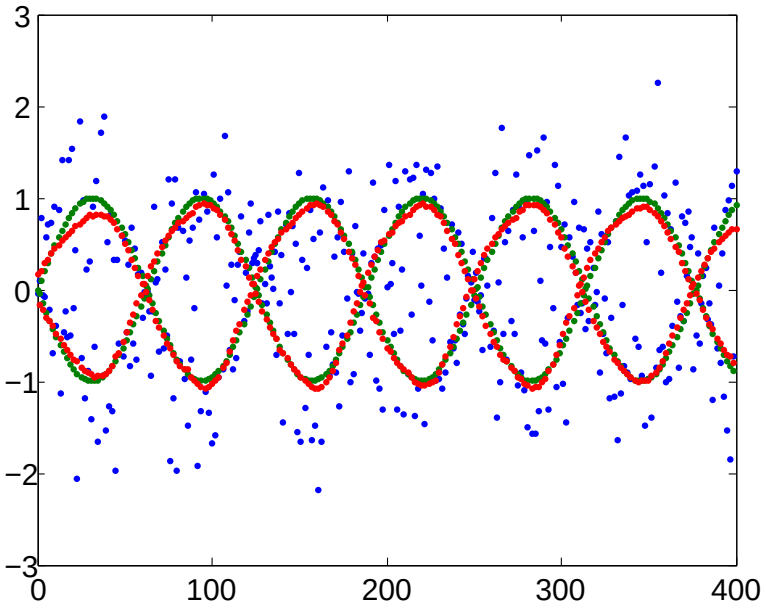


Figure 8.9: MeanShift with Implicit Prior. Results are in red, the observed noisy signal in blue and the clean original signal is in green. The original signal, highly non continuous, is defined as $s(n) = (-1)^n \sin(n/20)$.

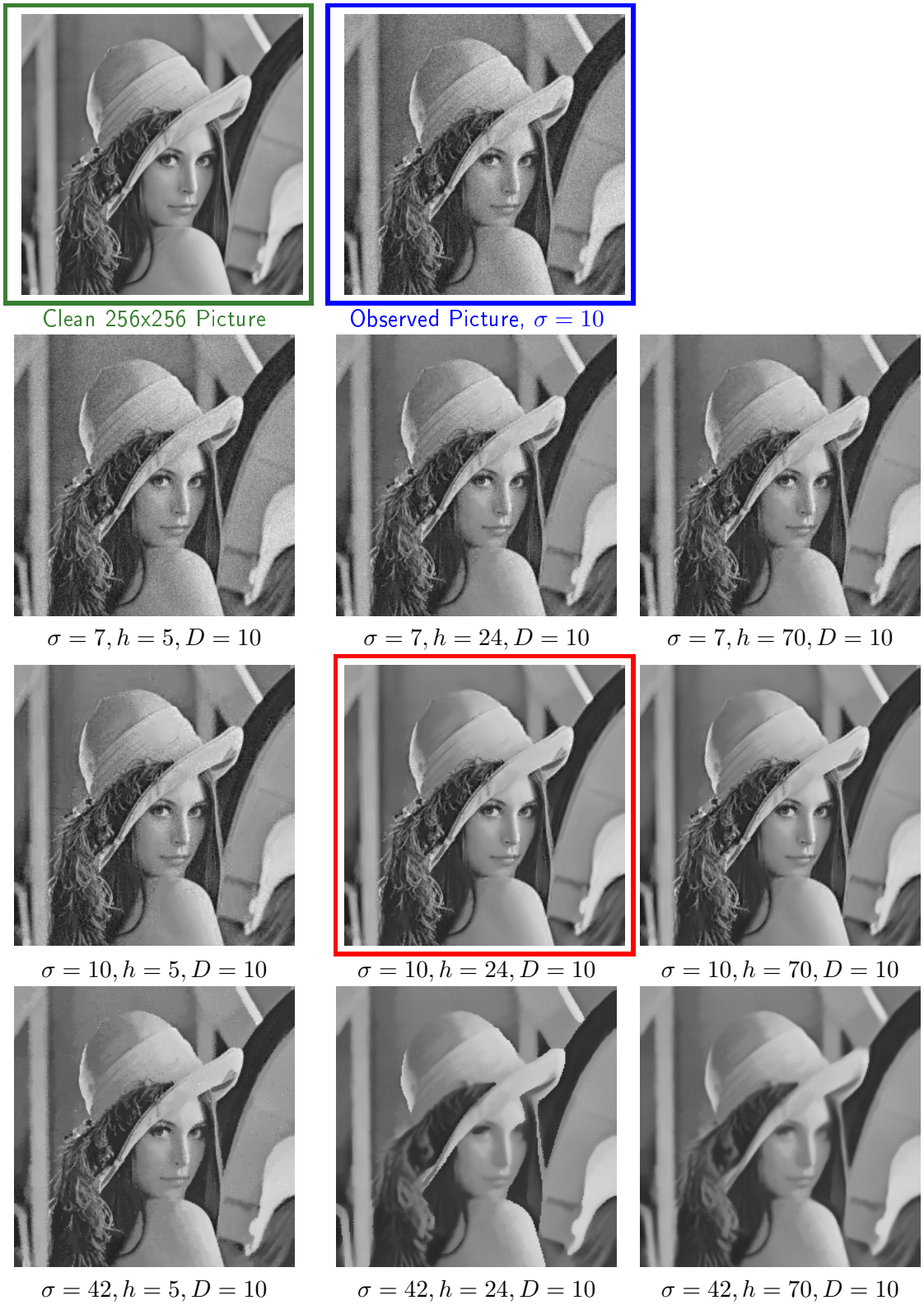


Figure 8.10: Influence of parameters on the Image Simplification. The bandwidth used in the MeanShift step appears to be not as influential that in standard MeanShift.

Figure 8.11: Image Simplification. $\sigma = 45, h = 80, D = 30$. (original image credits: Berkeley Segmentation Database, http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/)

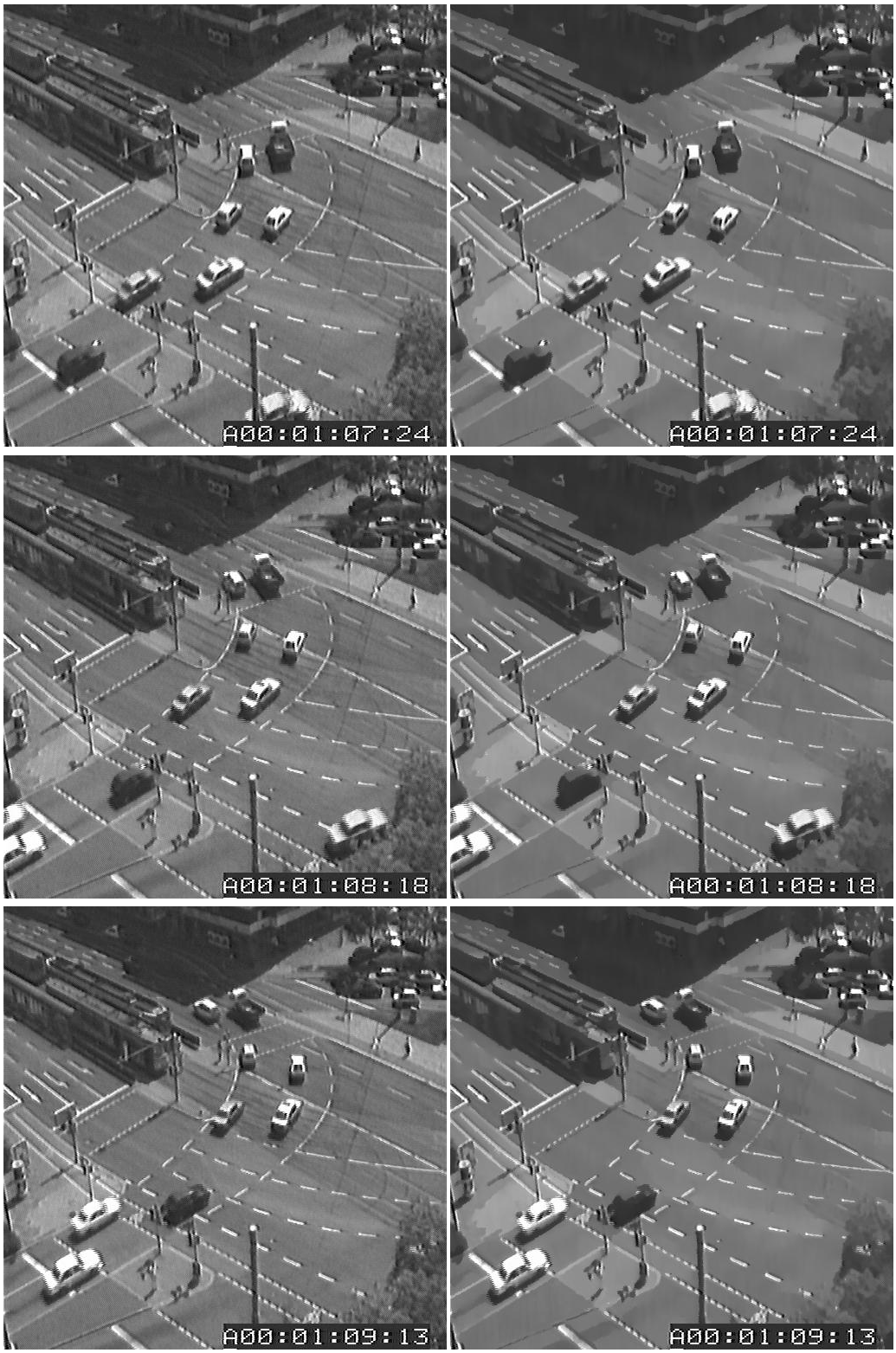


Figure 8.12: Image Simplification. $\sigma = 30, h = 10, D = 10$. **The compression ratios are** 0.8084, 0.8112 **and** 0.8154. (original image credits: Durlacher-Tor sequence, http://i21www.ira.uka.de/image_sequences/)

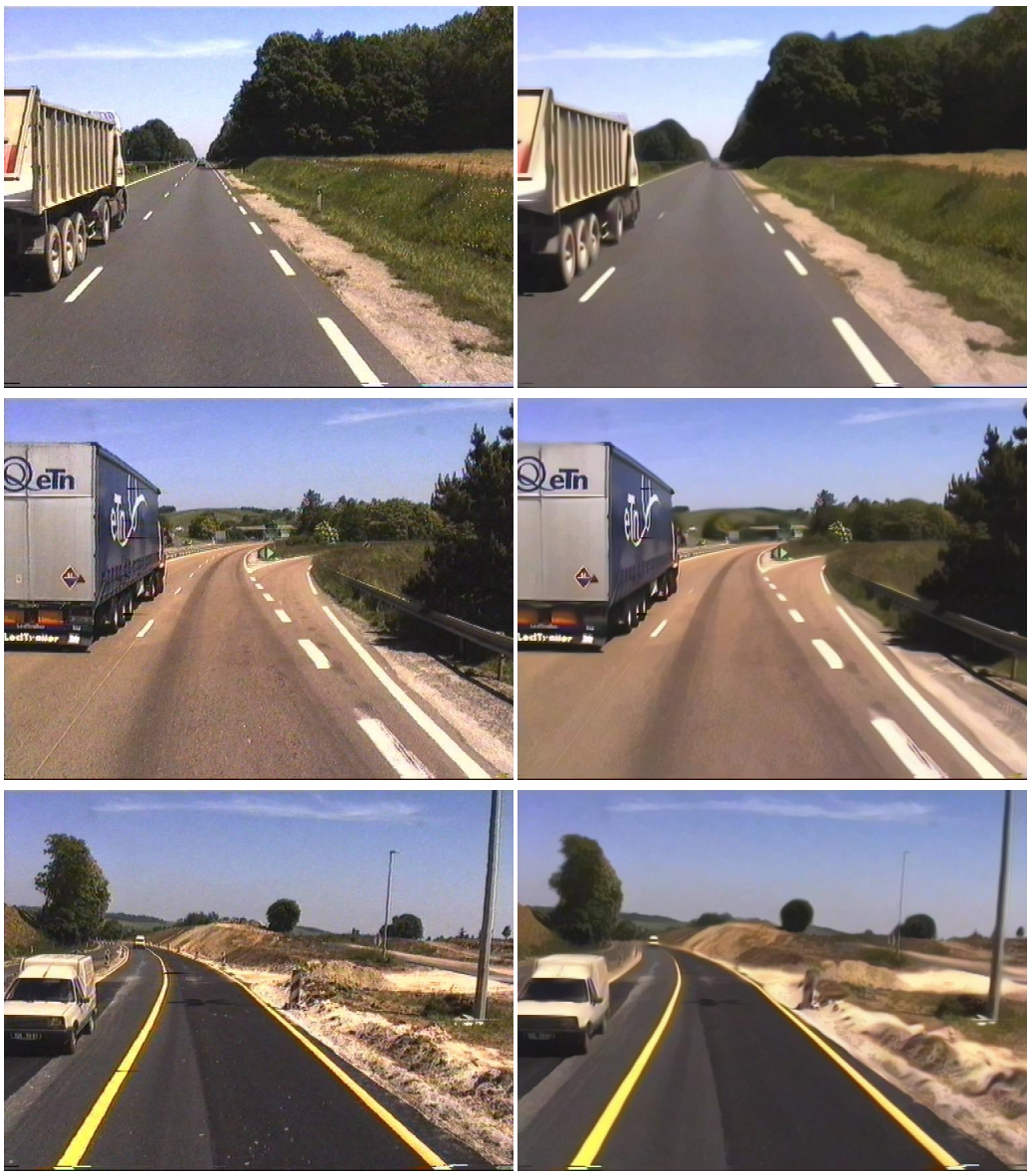


Figure 8.13: Image Simplification. $\sigma = 40, h = 80, D = 10$. **The compressions ratios are** 0.5598, 0.6073, 0.5576. (original image credits: Laboratoire Central des Ponts et Chaussées)

Chapter 9

Closing Remarks

This thesis presents several contributions in three post-production areas where signal processing statistical methods can be applied. In the first part, a new N-dimensional pdf transfer technique is used to transfer aspects of an image to another. In the second part, the image content is extracted by using two statistical trackers. Eventually the last part shows that statistics of image patches can be used to simplify the content the picture itself. In conclusion of this work, two remarks emerge from dealing with image processing tools in post-production.

Non-Parametric Approaches. The idea of re-using the statistics of real images to grade images is a simple but powerful method for rendering images. The method is part of the larger domain of non-parametric image processing. In a sense then, the first part of the thesis is closely related to the last chapter, since in both cases the statistics manipulated are extracted from the picture itself and not from a hand tuned prior. The main difficulty in image processing is that there is no available valid model for images. Non-parametric methods get around this problem by restricting the range of manipulations to manipulations based on real image statistics. The results are thus guaranteed to be visually acceptable, though not necessarily exact. These kind of approaches gained much success in texture synthesis and it transpires from the results of this work that they could be efficiently used for other applications.

Human In Control. The last remark of this conclusion is that users should be ultimately in control of algorithms. Whilst fully automated tools are eventually what the end-user is dreaming of, the reality of the post-production industry is that fully automated tools rarely give the exact desired results. To be useful, a tool has thus to provide the end-user a full

control on every step of its realisation. Although drawing object mattes is a painstaking task, artists simply prefer doing this on every frame of a movie if this results for sure in what they are looking for. This motivates too the approach adopted in chapter 7 to focus on deterministic algorithms. Since stochastic methods cannot provide a full control on the results and are therefore not well adapted to interactive tools. It would be thus beneficial to explore new ways of integrating the user in the algorithms, like for instance using techniques from relevance feed-back to minimise the interaction. But this is left for further research.

The End.

Chapter A

Appendix

A.1 Sampling PDFs

Consider pdfs of random variables that can take k different values. The pdfs are characterised by

$$q \in \mathcal{P}_k \Longleftrightarrow \begin{cases} \forall\ i \in [1;k]\ , \quad q(i) \geq 0 \\ \sum_{i=1}^{k} q(i) = 1 \end{cases} \tag{A.1}$$

The problem here presents similarities with picking points uniformly on a sphere [Mul59, Mar72]. The set of k-state pdfs $\mathcal{P}_k$ can indeed be seen as a unit hypersphere in a L^1 space: $\mathcal{P}_k = \left\{\mathbf{q} \in (\mathbb{R}^+)^k \ :\ \|\mathbf{q}\|_1 = 1\right\}$, where $\|q\|_1 = \sum |q_i|$ refers to the L^1 norm (see figure A.1). Thus the idea is to follow the same outlines of the point picking methods. The method proposed by Muller [Mul59] is particularly attractive and easy. The method to pick a random point on a hypersphere is to generate k Gaussian random variables $z_1, \ldots, z_k$. Then the distribution of the vectors

$$\frac{1}{\sqrt{z_1^2 + \ldots + z_k^2}} \begin{pmatrix} z_1 \\ \vdots \\ z_k \end{pmatrix} \tag{A.2}$$

is uniform over the surface of the hypersphere.

Using the same arguments as proposed in the point picking method yields a method for picking k-state pdfs uniformly over the pdf space $\mathcal{P}_k$. The method is to generate k

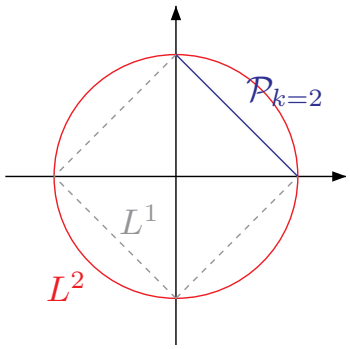


Figure A.1: The k-state pdf space $\mathcal{P}_k$ corresponds to the positive quadrant of the L^1-hypersphere.

exponential random variables $(z_1, z_2, \cdots, z_k)$. Then the distribution of the vectors

$$\frac{1}{z_1 + \ldots + z_k} \begin{pmatrix} z_1 \\ \vdots \\ z_k \end{pmatrix} \tag{A.3}$$

is uniform over the pdf space $\mathcal{P}_k$. It is probable that the proof of this method exists somewhere else in the mathematical literature, but it could not be found. The proof follows the same steps as the point picking method. The difference is the choice of a exponential distribution instead of a normal distribution.

Theorem 2 (Uniform pdf Sampling). *Let $\mathcal{P}_k = \left\{\mathbf{q} \in (\mathbb{R}^+)^k \ : \ \|\mathbf{q}\|_1 = 1\right\}$ be the space all pdfs for random variables taking n values and let the elements of $\mathbf{z} \in (\mathbb{R}^+)^k$ have independent exponential distributions (with same scale parameter λ). Then $\mathbf{q} = \mathbf{z}/\|\mathbf{z}\|_1$ is uniformly distributed over $\mathcal{P}_k$.*

proof. By independence, the joint density for $\mathbf{z}$ is:

$$\begin{aligned} p(z_1, z_2, \cdots, z_k) &= p(z_1)p(z_2)\cdots p(z_k) && \text{(A.4)} \\ &= \lambda^k e^{-\lambda(z_1+z_2+\cdots+z_k)} && \text{(A.5)} \\ &= \lambda^k e^{-\lambda(\|z\|_1)} && \text{(A.6)} \end{aligned}$$

for any region A in $\mathcal{P}_k$,

$$\Pr(\mathbf{z} \in A \subset \mathcal{P}_k) = \int_A \lambda^k \exp\left(-\lambda\|\mathbf{z}\|_1\right) \tag{A.7}$$

This integral is clearly isotropic since it depends only on the length of $\mathbf{z}$ and not on its direction. For $\mathbf{z} \in A \subset \mathcal{P}_k$, $\mathbf{z}$ is restricted such that $\|\mathbf{z}\|_1 = 1$, then $\Pr(\mathbf{z} \in A \subset \mathcal{P}_k)$ is constant. Hence $\mathbf{q}$ is uniformly distributed over $\mathcal{P}_k$. □

A.2 Directions

Table A.1: Optimised Rotations for $N = 2$

No.	1	2	3	4
x	1.000000 0.000000	0.707107 -0.707107	0.382683 0.923880	0.382683 -0.923880
y	0.000000 1.000000	0.707107 0.707107	0.923880 -0.382683	0.923880 0.382683
No.	5	6	7	8
x	0.195079 0.980788	0.555580 -0.831463	0.831476 -0.555560	0.195102 -0.980783
y	0.980788 -0.195079	0.831463 0.555580	0.555560 0.831476	0.980783 0.195102
No.	9	10	11	12
x	0.728428 -0.685123	0.999531 0.030622	0.410795 -0.911728	-0.354213 -0.935165
y	-0.685123 -0.728428	0.030622 -0.999531	0.911728 0.410795	0.935165 -0.354213
No.	13	14	15	16
x	0.165027 0.986289	0.814104 0.580720	0.974365 0.224971	0.848059 -0.529902
y	0.986289 -0.165027	-0.580720 0.814104	-0.224971 0.974365	0.529902 0.848059

Table A.2: Optimised Rotations for $N = 3$

No.	1	2
x	1.000000 0.000000 0.000000	0.333333 0.666667 0.666667
y	0.000000 1.000000 0.000000	0.666667 0.333333 -0.666667
z	0.000000 0.000000 1.000000	-0.666667 0.666667 -0.333333
No.	3	4
x	0.577350 0.211297 0.788682	0.577350 0.408273 0.707092
y	-0.577350 0.788668 0.211352	-0.577350 -0.408224 0.707121
z	0.577350 0.577370 -0.577330	0.577350 -0.816497 0.000029
No.	5	6
x	0.332572 0.910758 0.244778	0.243799 0.910726 0.333376
y	-0.910887 0.242977 0.333536	0.910699 -0.333174 0.244177
z	-0.244295 0.333890 -0.910405	-0.333450 -0.244075 0.910625
No.	7	8
x	-0.109199 0.810241 0.575834	0.759262 0.649435 -0.041906
y	0.645399 0.498377 -0.578862	0.143443 -0.104197 0.984158
z	0.756000 -0.308432 0.577351	0.634780 -0.753245 -0.172269
No.	9	10
x	0.862298 0.503331 -0.055679	0.982488 0.149181 0.111631
y	-0.490221 0.802113 -0.341026	0.186103 -0.756525 -0.626926
z	-0.126988 0.321361 0.938404	-0.009074 0.636722 -0.771040
No.	11	12
x	0.687077 -0.577557 -0.440855	0.463791 0.822404 0.329470
y	0.592440 0.796586 -0.120272	0.030607 -0.386537 0.921766
z	-0.420643 0.178544 -0.889484	-0.885416 0.417422 0.204444

A.3 Simple Matlab Code for 1D Signal

```
function R = Simplifier_1D(S, D, sigma, h)

M = length(S);
R = S;

rho_list = [1 2 4 8 16 32]; % collect the neighbourhing sizes

for it=1:10

    t = zeros(size(S));
    tw = zeros(size(S));

    for j = -D:D

        % to take care of range issues
        j_first_0 = max(j, 0) + 1;
        j_first_d = max(-j, 0) + 1;
        j_M  = M - abs(j);

        s0 = S(j_first_0:j_first_0+j_M-1);
        sd = S(j_first_d:j_first_d+j_M-1);
        r0 = R(j_first_0:j_first_0+j_M-1);
        rd = R(j_first_d:j_first_d+j_M-1);

        % image difference
        diff = abs(s0-sd).^2;

        % loop over spatial scale h
        p_h = zeros(1, j_M);

        for rho_i = 1:length(h_list)
            rho = rho_list(rho_i);
            m_rho = rho - 1;
            sd_rho = sqrt(smoothgauss(diff, rho)); % SD estimate
            lambda_rho = sqrt(2)*sigma/sqrt(2*m_rho);
            sigma_rho = sqrt(2)*sigma*sqrt((m_rho-1)/m_rho);

            p_h = p_h + rho * block_kernel(sd_rho, sigma_rho, lambda_rho);
        end

        % mean shift factor
        diff_0 = abs(r0 - sd);
        kdiff = kernel(diff_0, h);
        p_h  = p_h .* kdiff;

        t(j_first_0:j_first_0+j_M-1) = t(j_first_0:j_first_0+j_M-1) + p_h .* sd;
        tw(j_first_0:j_first_0+j_M-1) =  tw(j_first_0:j_first_0+j_M-1) + p_h;
    end

    % normalising and update
    R = t ./ tw;
end

function pv = kernel(v, hv)
v = v./hv;
pv = exp(- v.*v/2)/ sqrt(2*pi)/hv;  % gaussian

function pv = block_kernel(s, sh, lh)
f = 2;
pv = (1 - 1./(1+exp(-f*(s - sh)/lh)))/ (log(1 + exp(f*sh/lh))*lh/f);

function sr = smoothgauss(s, scale)
u = -3*scale:3*scale;
h = exp(-(u/scale).^2/2);
```

```
h((length(u) + 1)/2)=0;  % the sample is NOT in its neighbourhood
h = h/sum(h);
sr = filter2(h,s,'same');
```

Bibliography

[Aru02] S. Arulampalam, S. Maskell, N. Gordon and T. Clapp., A tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking, 2002.

[Bie87] J. Biemond, D. E. Boekee, L. Looijenga and R. Plompen, A pel-recursive Wiener based displacement estimation algorithm, *Signal Processing*, 13:pp. 399–412, 1987.

[Bla93] M. J. Black and P. Anandan, A Framework for the Robust Estimation of Optical Flow, in *Proceedings of Fourth International Conference on Computer Vision*, pp. 231–236, May 1993.

[Bla04] A. Blake, C. Rother, M. Brown, P. Perez and P. Torr, Interactive Image Segmentation using an adaptive GMMRF model, in *Proc. European Conf. Computer Vision*, 2004.

[Bol03] J. Bolz, I. Farmer, E. Grinspun and P. Schroder, Sparse Matrix Solvers on the GPU: Conjugate Gradients and Multigrid, *ACM Transactions on Graphics*, 22(3):pp. 917–924, July 2003.

[Boy04] Y. Boykov and V. Kolmogorov, An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9, pp. 1124–1137, Sept. 2004.

[Bro04] T. Brox, A. Bruhn, N. Papenberg and J. Weickert, High Accuracy Optical Flow Estimation Based on a Theory for Warping, in *Proceedings ECCV*, pp. 25–36, 2004.

[Cha03] C. Chang and R. Ansari, Kernel Particle Filter: Iterative Sampling for Efficient Visual Tracking, in *IEEE International Conference on Image Processing (ICIP)*, 2003.

[Cha04] Y. Chang, K. Uchikawa and S. Saito, Example-based color stylization based on categorical perception, in *Proceedings of the 1st Symposium on Applied per-*

ception in graphics and visualization (APGV), pp. 91–98, ACM Press, 2004, ISBN 1-58113-914-4.

[Che95] Y. Cheng, Mean Shift, Mode Seeking, and Clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:pp. 790–799, 1995.

[Che03] H. Chen and P. Meer, Robust Regression with Projection Based M-estimators., in *International Conference on Computer Vision*, pp. 878–885, Nice, France, October 2003.

[Com01] D. Comaniciu, V. Ramesh and P. Meer, The Variable Bandwidth Mean Shift and Data-Driven Scale Selection, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV'01)*, pp. 438–445, Vancouver, Canada, 2001.

[Com02] D. Comaniciu and P. Meer, Mean Shift: A Robust Approach Toward Feature Space Analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):pp. 603–619, May 2002.

[Com03] D. Comaniciu, V. Ramesh and P. Meer, Kernel-Based Object Tracking, *IEEE Trans Pattern Anal Machine Intell*, 25(5):pp. 564–575, 2003.

[Cra04] A. Crawford, H. Denman, F. Kelly, F. Pitié and A. C. Kokaram, Gradient Based Dominant Motion Estimation with Integral Projections for Real Time Video Stabilisation, in *IEEE International Conference on Image Processing (ICIP'04)*, Singapor, October 2004.

[Dah04] R. Dahyot, N. Rea, A. Kokaram and N. Kingsbury, Inlier Modeling for Multimedia Data Analysis, in *IEEE International Workshop on Multimedia Signal Processing*, Siena Italy, Sep. 2004.

[Dah05] R. Dahyot and S. Wilson, Robust Scale Estimate for the Generalized Gaussian Probability Density Function, *Tech. rep.*, Department of Statistics, University of Dublin, Trinity College, Ireland, Aug. 2005.

[Deb97] P. E. Debevec and J. Malik, Recovering high dynamic range radiance maps from photographs, in *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 369–378, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1997, ISBN 0-89791-896-7.

[Dec97] E. Decencière, *Restauration automatique de films anciens*, Ph.D. thesis, Ecole Nationale Supérieure des Mines de Paris, December 1997.

[DeC02] D. DeCarlo and A. Santella, Stylization and abstraction of photographs, in *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 769–776, ACM Press, New York, NY, USA, 2002, ISBN 1-58113-521-1.

[Des95] X. Descombes, J.-F. Mangin, E. Pechersky and M. Sigelle, Fine Structures Preserving Markov Model for Image Processing, in *9th Scandinavian Conference on Image Analysis SCIA'95*, 1995.

[Dou00] A. Doucet, N. de Freitas and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, 2000.

[Duf00] F. Dufaux and J. Konrad, Efficient, Robust, and Fast Global Motion Estimation for Video Coding, *IEEE Transaction on Image Processing*, 9(3):pp. 497–501, March 2000.

[Efr99] A. A. Efros and T. K. Leung, Texture Synthesis by Non-parametric Sampling, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV'99)*, pp. 1033–1038, Corfu, Greece, September 1999.

[Efr01] A. Efros and W. Freeman, Image quilting for texture synthesis and transfer, in *Proceedings of ACM SIGGRAPH*, pp. 341–346, 2001.

[Eva99] L. Evans, Partial differential equations and Monge-Kantorovich mass transfer, *Current Developments in Mathematics 1997*, pp. 65–126, 1999.

[Fat02] R. Fattal, D. Lischinski and M. Werman, Gradient domain high dynamic range compression, in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques (SIGGRAPH '02)*, pp. 249–256, ACM Press, New York, NY, USA, 2002, ISBN 1-58113-521-1.

[Fuk90] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 1990.

[Gal05] C. Gallagher and A. Kokaram, Nonparametric Wavelet Based Texture Synthesis, *Proceedings of IEEE International Conference on Image Processing (ICIP), Genova, Italy*, September 2005.

[Gem84] S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):pp. 721–741, Novembre 1984.

[Goo03] N. Goodnight, C. Woolley, G. Lewin, D. Luebke and G. Humphreys, A Multigrid Solver for Boundary Value Problems Using Programmable Graphics Hardware, in *Graphics Hardware 2003*, pp. 102–111, July 2003.

[Her01] A. Hertzmann, C. Jacobs, N. Oliver, B. Curless and D. Salesin, Image Analogies, in *Proceedings of ACM SIGGRAPH*, 2001.

[Hor81] B. Horn and B. Schunck, Determining optical flow, *Artificial Intelligence*, 17:pp. 185–203, 1981.

[Hub81] P. Huber, *Robust Statistics*, John Wiley and Sons, 1981.

[Hue02] C. Hue, J.-P. Le Cadre and P. Pérez, Sequential Monte Carlo methods for multiple target tracking and data fusion, *IEEE Trans on Signal Processing*, 50(2):pp. 309–325, February 2002.

[ici03] Special Session on Sports Video Analysis (chaired by I. Sezan and B.Li), in *IEEE International Conference on Image Processing (ICIP)*, Sep 2003.

[Jaf03] G. Jaffré and A. Crouzil, Non-Rigid Object Localization from Color Model using Mean Shift, in *IEEE International Conference on Image Processing (ICIP)*, 2003.

[Ji04] Y. Ji, H.-B. Liu, X.-K. Wang and Y.-Y. Tang, Color Transfer to Greyscale Images using Texture Spectrum, in *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, pp. 4057–4061, Shanghai, 2004.

[Jia04] J. Jia, J. Sun, C.-K. Tang and H.-Y. Shum, Bayesian Correction of Image Intensity with Spatial Consideration, in *Proceedings of the 8th European Conference on Computer Vision (ECCV'04)*, pp. 342–354, 2004.

[Jin01] H. Jin, P. Favaro and S. Soatto, Real-Time Feature Tracking and Outlier Rejection with Changes in Illumination, in *ICCV*, pp. 684–689, Jul. 2001.

[Joy04] L. Joyeux, E. Doyle, H. Denman, A. Crawford, A. Kokaram and R. Fuller, Content Based Access for a massive database of human observation video, in *Workshop on Multimedia and Image Retrieval*, pp. 46–52, October 2004.

[Kan03] S. B. Kang, M. Uyttendaele, S. Winder and R. Szeliski, High dynamic range video, *ACM Transactions on Graphics*, 22(3):pp. 319–325, 2003.

[Kim04] S. J. Kim and M. Pollefeys, Radiometric Alignment of Image Sequences, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 645–651, 2004.

[Kok98] A. Kokaram, *Motion Picture Restoration: Digital Algorithms for Artefact Suppression in Degraded Motion Picture Film and Video*, Springer Verlag, 1998.

[Kok03] A. C. Kokaram, R. Dahyot, F. Pitié and H. Denman, Simultaneous Luminance and Position Stabilization for Film and Video, in *Visual Communications and Image Processing (VCIP)*, San Jose, California USA, January 2003.

[Kok05] A. Kokaram, F. Pitié, R. Dahyot, N. Rea and S. Yeterian, Content Controlled Image Representation for Sports Streaming, in *IEEE workshop on Content Based Multimedia Indexing (CBMI'05)*, Riga, June 2005.

[Kru03] J. Kruger and R. Westermann, Linear Algebra Operators for GPU Implementation of Numerical Algorithms, *ACM Transactions on Graphics*, 22(3):pp. 908–916, July 2003.

[Kug75] C. D. Kuglin and D. C. Hines, The phase correlation image alignment method, in *Proceedings of IEEE International Conference on Cybernetics and Society*, pp. 163–165, 1975.

[Lai99] S.-H. Lai and M. Fang, Robust and Efficient Image Alignement, in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 167–172, Fort Collins, Colorado, June 1999.

[Luc01] L. Lucchese and S. K. Mitra, a new Method for Color Image Equalization, in *Proceedings of the IEEE International Conference on Image Processing (ICIP'01)*, pp. 133–136, 2001.

[Man02] S. Mann, C. Manders and J. Fung, Painting with looks: photographic images from video using quantimetric processing, in *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pp. 117–126, ACM Press, New York, NY, USA, 2002, ISBN 1-58113-620-X.

[Mar72] G. Marsaglia, Choosing a point from the surface of a sphere, *Annals of mathematical statistics*, 43:pp. 645–646, 1972.

[Mar82] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, W.H. Freeman, 1982.

[Mig03] M. Mignotte, Unsupervised statistical sketching for non-photorealistic rendering models., in *IEEE International Conference on Image Processing (ICIP'03)*, pp. 573–576, 2003.

[Mit99] T. Mitsunaga and K. Nayar, Radiometric Self Calibration, in *Proceedings of Computer Vision and Pattern Recognition (CVPR'99)*, pp. 374–380, 1999.

[Mor96] D. Morris, X. Descombes and J. Zerubia, The Ising/Potts model is not well suited to segmentation tasks, in *IEEE Digital Signal Processing Workshop*, Sept. 1996.

[Mor03] J. Morovic and P.-L. Sun, Accurate 3D image colour histogram transformation, *Pattern Recognition Letters*, 24(11):pp. 1725–1735, 2003.

[Mul59] M. E. Muller, A note on a method for generating points uniformly on n-dimensional spheres, *Commun ACM*, 2(4):pp. 19–20, 1959.

[Nar00] V. Naranjo and A. Albiol, Flicker reduction in old films, in *Proceedings of the 2000 International Conference on Image Processing (ICIP'00)*, vol. 2, pp. 657–659, September 2000.

[Odo95] J.-M. Odobez and P. Bouthemy, Robust multiresolution estimation of parametric motion models., *Journal of Visual Communication and Image Representation*, 6(1):pp. 348–365, 1995.

[Ohu00] T. Ohuchi, T. Seto, T. Komatsu and T. Saito, A robust method of image flicker correction for heavily-corrupted old film sequences, in *Proceedings of the 2000 International Conference on Image Processing (ICIP'00)*, September 2000.

[OK02] H.-W. OK, Y. Seo and K. Hong, Multiple Soccer Players Tracking by Condensation with Occlusion Alarm Probability, in *International Workshop on Statistical Methods for Vision Processing (in conjunction with ECCV)*, 2002.

[Oku04] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little and D. G. Lowe, A Boosted Particle Filter: Multitarget Detection and Tracking, in *European Conference on Computer Vision (ECCV)*, 2004.

[Pap00] M. Pappas and I. Pitas, Digital Color Restoration of Old Paintings, *Transactions on Image Processing*, (2):pp. 291–294, Feb. 2000.

[Per95] P. Perona, Deformable Kernels for Early Vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):pp. 488–499, 1995.

[Pic03] E. Pichon, M. Niethammer and G. Sapiro, Color histogram equalization through mesh deformation, in *IEEE International Conference on Image Processing (ICIP'04)*, pp. II 117–120, 2003.

[Pit02] F. Pitié, 2002, Video Material for Flicker Removal.
URL www.sigmedia.tv/research/restoration/flicker/

[Pit03] F. Pitié, R. Dahyot and A. C. Kokaram, Suppression du bruit de pompage des vidéos, in *Proceedings 19ème Colloque sur le Traitement du Signal et des Images, GRETSI'03*, Paris, France, September 2003.

[Pit04a] F. Pitié, R. Dahyot, F. Kelly and A. C. Kokaram, A New Robust Technique for Stabilizing Brightness Fluctuations in Image Sequences, in *2nd Workshop on Statistical Methods in Video Processing In conjunction with ECCV 2004 Prague*, Prague, Czech Republic, may 2004.

[Pit04b] F. Pitié, A. C. Kokaram and R. Dahyot, Oriented Particle Spray: A New Probabilistic Contour Tracing with Directional Information, in *Irish Machine Vision and Image Processing (IMVIP'04)*, Dublin, Ireland, September 2004.

[Pit05a] F. Pitié, Video Material, Website, 2005.
URL www.sigmedia.tv/publications/publis/ICIP2005/fpitie/

[Pit05b] F. Pitié, S.-A. Berrani, R. Dahyot and A. Kokaram, Off-line Multiple Object Tracking using Candidate Selection and the Viterbi Algorithm, in *IEEE International Conference on Image Processing (ICIP'05)*, Genoa, September 2005.

[Pit05c] F. Pitié, A. Kokaram and R. Dahyot, N-Dimensional Probability Density Function Transfer and its Application to Colour Transfer, in *International Conference on Computer Vision (ICCV'05)*, Beijing, October 2005.

[Pit05d] F. Pitié, A. Kokaram and R. Dahyot, Towards Automated Colour Grading, in *IEE European Conference on Visual Media Production (CVMP'05)*, London, December 2005.

[Pit06] F. Pitié, PDF Transfer - Thesis Material, 2006.
URL www.sigmedia.tv/research/PDFTransfer/

[Pre92] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, New York, NY, USA, 1992, ISBN 0521437148.

[Pér01] P. Pérez, A. Blake and M. Gangnet, Jetstream: Probabilistic Contour Extraction with Particles, *Proc Int Conf on Computer Vision (ICCV)*, II(5):pp. 524–531, 2001.

[Pér03] P. Pérez, A. Blake and M. Gangnet, Poisson image editing, *ACM Trans Graph*, 22(3):pp. 313–318, 2003.

[Rab86] L. R. Rabiner and B. H. Juang, An Introduction to Hidden Markov Models, *IEEE ASSP Mag*, pp. 4–16, 1986.

[Rei01] E. Reinhard, M. Ashikhmin, B. Gooch and P. Shirley, Color Transfer between Images, *IEEE Computer Graphics Applications*, 21(5):pp. 34–41, 2001.

[Ric95] P. Richardson and D. Suter, Restoration of historic film for digital compression: a case study, in *Proceedings of the International Conference on Image Processing (ICIP'95)*, pp. 49–52, october 1995.

[Roo99a] P. van Roosmalen, *Restoration of archived film and video*, Ph.D. thesis, Delft University of Technology, October 1999.

[Roo99b] P. van Roosmalen, R. Lagendijk and J. Biemond, Correction of intensity flicker in old film sequences, *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7):pp. 1013–1019, 1999.

[Rot04] C. Rother, V. Kolmogorov and A. Blake, GrabCut - Interactive Foreground Extraction using Iterated Graph Cuts, *Proc ACM Siggraph*, 2004.

[Rot05a] S. Roth and M. J. Black, Fields of Experts: A Framework for Learning Image Priors, in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Beijing, China, 2005.

[Rot05b] S. Roth and M. J. Black, On the Spacial Statistics of Optical Flow, in *IEEE International Conference on Computer Vision (ICCV'05)*, 2005.

[Rud98] D. Ruderman, T. Cronin and C. Chiao, Statistics of Cone Responses to Natural Images: Implications for Visual Coding, *Journal of the Optical Society of America*, (8):pp. 2036–2045, 1998.

[Sch99] P. Schallauer, A. Pinz and W. Haas, Automatic Restoration Algorithms for 35mm Film, *Videre: Journal of Computer Vision Research*, 1(3):pp. 60–85, summer 1999.

[Sil86] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, 1986, ISBN 0-412-24620-1.

[Sim95] E. P. Simoncelli and H. Farid, Steerable Wedge Filters, in *ICCV*, pp. 189–194, 1995.

[Sko01] S. Skoneczny, Defects Removing in Old Movies, in *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis (ISPA'01))*, pp. 38–42, Croatia, June 2001.

[Sti99] C. Stiller and J. Konrad, Estimating Motion in Image Sequences, *IEEE Signal Processing Magazine*, 16(4):pp. 70–91, July 1999.

[Tek95] A. M. Tekalp, *Digital Video Processing*, Signal Processing Series, Prentice-Hall, 1995.

[Ter04] B. Terwijn, J. Porta and B. Kröse, A Particle Filter to Estimate non-Markovian States, in *International Conference on Intelligent Autonomous Systems, IAS'04*, 2004.

[The89] C. W. Therrien, *Decision estimation and classification: an introduction to pattern recognition and related topics*, John Wiley & Sons, Inc., 1989, ISBN 0-471-83102-6.

[Vla04] T. Vlachos, Flicker Correction for Archived Film Sequences, *IEEE Transactions on Circuits and Systems for Video Technology*, 14(4):pp. 508–516, april 2004.

[Wan04a] H. Wang and D. Suter, Robust Adaptive-Scale Parametric Model Estimation for Computer Vision, *Transactions on Pattern Analysis and Machine Intelligence*, 26(11), November 2004.

[Wan04b] J. Wang, B. Thiesson, Y. Xu and M. Cohen, Image and Video Segmentation by Anisotropic Kernel Mean Shift., in *European Conference on Computer Vision (ECCV'04)*, pp. 238–249, 2004.

[Wan04c] J. Wang, Y. Xu, H.-Y. Shum and M. F. Cohen, Video tooning, *ACM Trans Graph*, 23(3):pp. 574–583, 2004.

[Weia] E. W. Weisstein, Radon Transform, from MathWorld–A Wolfram Web Resource.
URL http://mathworld.wolfram.com/RadonTransform.html

[Weib] E. W. Weisstein, Standard Deviation Distribution, from MathWorld–A Wolfram Web Resource.
URL http://mathworld.wolfram.com/StandardDeviationDistribution.html

[Wei98] J. Weickert, B. ter Haar Romeny and M. Viergever, Efficient and Reliable Schemes for Nonlinear Diffusion Filtering, *IEEE Transactions on Image Processing*, 7(3):pp. 398–410, march 1998.

[Wel02] T. Welsh, M. Ashikhmin and K. Mueller, Transferring Color to Greyscale Images, in *Proceedings of ACM SIGGRAPH*, pp. 227–280, San Antonio, 2002.

[Wik] Wikipedia, Hypersphere.
URL http://en.wikipedia.org/wiki/Hypersphere

[Yan00] X. Yang and N. Chong, Enhanced approach to film flicker removal, *Proceedings of SPIE Applications of Digital Image Processing XXIII*, 4115:pp. 39–47, 2000.

[Yu99] W. Yu, K. Daniilidis and G. Sommer, Approximate orientation steerability based on angular gaussians, 1999.