PERSPECTIVES ON PANORAMIC PHOTOGRAPHY

THÈSE N^{O} 2419 (2001)

PRÉSENTÉE AU DÉPARTEMENT DE SYTÈMES DE COMMUNICATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES TECHNIQUES

PAR

David HASLER

Ingénieur en microtechnique diplômé EPF
de nationalité suisse et originaire de Wetzikon (ZH)

acceptée sur proposition du jury:

Prof. M. Vetterli, Prof. S. Süsstrunk, directeurs de thèse
Dr M. Näf, rapporteur
Dr L. Sbaiz, rapporteur
Dr I. Tastl, rapporteur
Prof. P. Vandergheynst, rapporteur

Lausanne, EPFL
2002

Perspectives on Panoramic Photography

David Hasler

Thesis nb 2419, July 2001

Résumé

L'imagerie digitale enrichit le champ des ressources de la photographie traditionelle, notamment grâce à la disponibilité immédiate des images, et aux possibilités de traitement de ces images. La présente thèse expose certaines questions du traitement de l'image. En particulier, elle couvre les principaux aspects de la création d'images panoramiques et énonce une technique de rendu de l'image se basant sur des propriétés du système visuel humain. Le premier sujet traité est l'alignement d'un panorama de 360 degrés, qui soulève la difficulté de l'alignement de la dernière image sur la première. Dans la suite, il s'agit de la correction de la parallaxe. La parallaxe apparaît lorsque les photographies ne sont pas toutes prises depuis le même endroit. Une méthode qui consiste à segmenter la scène en plusieurs plans de profondeur est proposée pour reconstruire la scène de manière cohérente. Tout un chapitre est ensuite consacré à la couleur, ou comment corriger les incohérences de couleurs qui surviennent entre plusieurs prises de vues d'un même paysage. La partie suivante aborde le problème des discordances entre deux images, généralement causées par l'apparition ou la disparition d'un objet d'une photographie à l'autre. Contrairement à l'approche classique qui suppose que ces parties de la scène génèrent, lorsque deux images sont comparées points-à-points, des valeurs de comparaison totalement arbitraires, la méthode proposée caractérise ces valeurs de comparaison par une distribution calculée sur la base des histogrammes des images. Ceci restreint le problème et permet la création d'estimateurs de mouvement plus robustes que ceux utilisés jusqu'à présent. Le dernier chapitre traite la question de l'affichage d'une image de grande dynamique. Ce type d'image contient à la fois des zones très claires et des zones très sombres qui ne peuvent être reproduites à l'écran - ou sur papier - sans pertes notables de qualité : soit les zones claires apparaissent brûlées, soit les zones sombres apparaissent en noir. La méthode présentée se base sur un modèle de perception de contraste et permet de comprimer la dynamique de l'image tout en maintenant les petits détails visibles dans l'image finale.

Abstract

Digital imaging brings a new set of possibilities to photography. For example, little pictures can be assembled to form a large panorama, and digital cameras are trying to mimic the human visual system to produce better pictures. This manuscript aims at developing the algorithms required to stitch a set of pictures together to obtain a bigger and better image.

This thesis explores three important topics of panoramic photography: The alignment of images, the matching of the colours, and the rendering of the resulting panorama. In addition, one chapter is devoted to 3D and constrained estimation.

Aligning pictures can be difficult when the scene changes while taking the photographs. A method is proposed to model these changes—or *outliers*—that appear in image pairs, by computing the outlier distribution from the image histograms and handling the image-to-image correspondence problem as a mixture of inliers versus outliers. Compared to the standard methods, this approach uses the information contained in the image in a better way, and leads to a more reliable result.

Digital cameras aim at reproducing the adaptation capabilities of the human eye in capturing the colours of a scene. As a consequence, there is often a large colour mismatch between two pictures. This work exposes a novel way of correcting for colour mismatches by modelling the transformation introduced by the camera, and reversing it to get consistent colours.

Finally, this manuscript proposes a method to render high dynamic range images that contain very bright as well as very dark regions. To reproduce this kind of pictures the contrast has to be reduced in order to match the maximum contrast displayable on a screen or on paper. This last method, which is based on a complex model of the human visual system, reduces the contrast of the image while maintaining the little details visible the scene.

Acknowledgements

During my time as a graduate student, I received help and support from many people. I would first Like to Thank my advisor Prof. Martin Vetterli for welcoming me in his laboratory, for the numerous discussions we had and for the liberty he left me for choosing the topics I wanted to work on. I would also Like to give a Big thank you to Luciano for the continuous help and advices he gave me throughout the thesis, and for the painful work of reading it through in the end. In a second time, I would also like to thank Prof. Sabine Süsstrunk, my second advisor, who joined the Laboratory during my thesis, and who helped me with everything related to colour and perception. I would also like to thank Serge Ayer, for giving me the ideas of the subject I ended up to work on. Also a special thank to Jocelyne, who made every administrative task very easy.

I would like to give a special Thank to Dr. Ingeborg Tastl, who welcomed me as an intern at Sony Reseach Labs in California, for proposing me a very nice and interesting subject (and helping me with it), and for sharing the sundays on the mountain bike tracks of the Bay Area. I'm also very thankful to her for accepting to be on my thesis comity, and doing the long trip from California.

I also like to thank the other members of the Jury, for accepting to read the thesis, and to come to comment my defense. Also thank you to Prof. Pierre Vandergheynst for helping me with contrast metrics.

On a more personal note, I would like to thank Jérôme and Claudio for the numerous discussion we had at coffee break as well as Pier-Luigi and Michael for the same reasons. Thanks also to Andrea (and to all the Lab) for the coffee machine I got at the end of the thesis (when you don't need coffee anymore, but when you can really appreciate it) and for the help on Appendix B. I would also like to thank my office-mate Pina, for sharing the sometimes-not-so-easy situation with me, for the good atmosphere she brought in the Lab, and also for the numerous cakes she brought at coffee time.

On a more remote note, I would like to thank my friends in California. A big thank to Laurent who welcomed me there, and prevented me to sleep under the bridges in this area where the housing was so difficult to find. Also thank you to Pascal and Didier, for the numerous rides to San Francisco, the photographic trips and the rides by bike. Finally, I like to thank the whole mountain biking group there.

Last but not least, I would like to thank my whole family for the nice sunday evenings we used to spend together. Also thanks to my parents for hosting me after my return from California and for the continuous support I got from them. Finally, I would like to give a great thank to Véronique, who was of great support during the - sometimes difficult - end of the thesis.

Contents

Chapter 1

Introduction

Digital photography is a domain in full expansion and is likely to replace the analog camera business at a consumer level. In the beginning, having a digital camera was a convenient way of putting images on the World Wide Web. For the user, a big advantage of the digital camera over the analog counterpart is the ability to view the image on the little screen in the back of the device. Until very recently, digital cameras were much more expensive than the analog ones, and to get the same quality, are still more expensive, although this is likely to change in a near future. The quality of the picture has never been an issue for the average user until he was able to print the pictures on paper. Right now, printing services are available on the net, and the digital revolution has clearly begun.

Digital imaging has many advantages over its analog counterpart: The most obvious is the immediate availability of the picture. Another advantage is that a digital camera can behave like a camera with a slow film or with a fast film, with a daylight or with a tungsten film, through its analog gain and white balancing capability. Finally, digital images offer a wide spectrum of post-processing capabilities. For the camera manufacturer, using a digital sensor allows any transformation between the data read by the sensor and the final image. Nowadays, the camera manufacturers are trying to mimic the human visual system in this way. For the user, the most appealing post-processing feature is the ability to combine several images together. For example, one can place a beautiful landscape around a picture of himself. The combination of images can be used to get a larger picture, to decompose the motion of someone into several appearance of him in a single image, or more generally to introduce a chronological aspect in the image. Combinations of images can also be used to construct an image of better quality.

The goal of this thesis is to cover the topic of panoramic photography. An example of panoramic photograph is given in Figure 1.1. Traditionally, this kind of picture was created using roundshot camera, depicted in Figure 1.2. This camera rotates around a vertical axis and takes the image through a vertical split. The film translates behind the split, and captures the full 360 degrees scene. It delivers the equivalent of taking a huge amount of pictures by rotating a camera on a tripod, keeping only the central column of each picture, and assembling them side by side to create the final scene. The camera delivers very high quality pictures, but is reserved to professionals because of the costs involved. One of our goal was to develop the algorithms that mimic the behaviour of the roundshot camera using a set of pictures taken with a hand held digital camera.

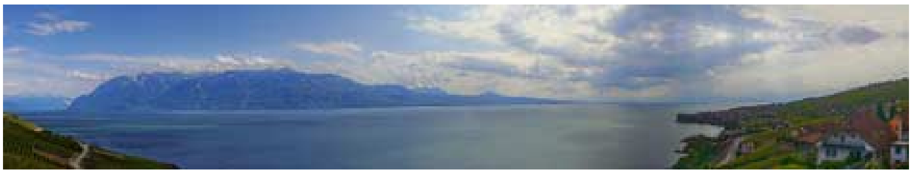


Figure 1.1: Example of panoramic picture. This pictures has been assembled using 25 pictures taken with a digital camera.

In order to assemble the pictures of a panorama, one has first to determine the relationship between a picture and its neighbour. Then, a global relationship among the pictures of the set is computed, such that the little errors in the picture-to-picture relationships do not alter the global panorama in a significant way. Then, the pictures are blended together: the transition from one picture to the next is done carefully to give the impression that the final panorama has not been taken with several pictures. Finally, the panorama is rendered as a single picture. The procedure is summarised in the chart of Figure 1.3. Most of our effort will be focused on computing the picture-to-picture relationships. Two kind of relationships are handled : geometrical relationships that define the alignment of a picture with the next, and colorimetric relationships that specify how the colour in an image relates to the colour in the next. Colour relationships are developed in Chapter 5. Two subtopics of the images alignment have been chosen: the alignment in presence of parallax in Chapter 3, and the alignment in presence of perturbations, or *outliers*, in Chapter 4. Blending considerations have been left out from the presentation.

The thesis explains how to reconstruct a panorama when the scene does not change during the shooting of the pictures. Chapter 2 introduces the geometrical relations of image pairs, several models of cameras and reviews the standard method to stitch two images together. Chapter 3 explains how to handle a set of picture as a whole by starting with an image-to-image comparison, and illustrates the problem by aligning a 360 deg panorama. In addition, it introduces a method to decompose the scene into several planes in space. This allows to reconstruct a panoramic image when the photographer moves between the captures instead of staying still and turning the camera, that is when parallax is present in the pictures. Chapter 4 presents a new way of handling outliers in the process of comparing images. An outlier is, for example, something that appears in an image and is absent in the others, for example a pedestrian that passes by in front of the camera. This allows to align pictures even if a large part of the images are covered by the outlier. Chapter 5 exposes how a camera handles colour and explains how to correct for colour mismatches between images. Finally, Chapter 6 exposes a technique to render high dynamic range images. High dynamic range images contain very bright and very dark regions. If these images are rendered on paper (or on the screen) using traditional techniques, there will be a loss of details either in the dark or in the bright region of the image. The technique uses some aspects of the human visual system in order to preserve the details in the whole image as best as possible.

If I had to read only one single chapter of the thesis, I would pick Chapter 4, which, I think, is the most original contribution to the field. It is simple and works well. If I had to read the whole thesis, I would leave the multi-planar model as well as the 360 deg panorama alignment of Chapter 3 for the end, in order to keep the motivation for the remaining chapters.

Figure 1.2: roundshot camera. This analog camera captures full panoramic images by rotating around a vertical axis.

The first is, in my opinion too complex to be practical, and the second is somehow counter-intuitive, but works in practice.

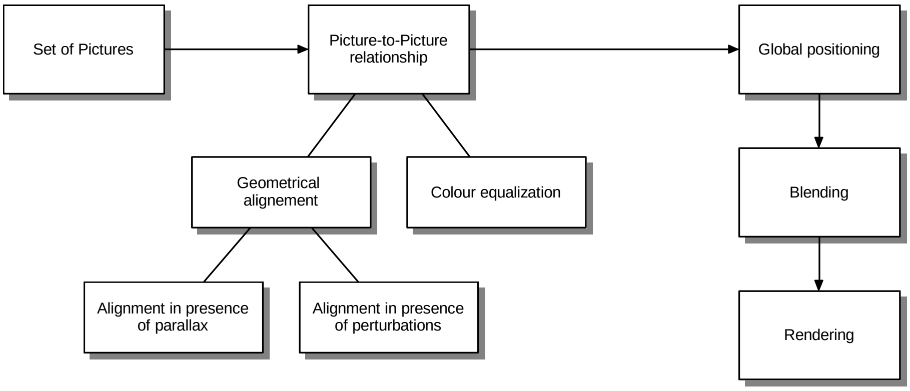

Set of Pictures
Picture-to-Picture relationship
Global positioning
Geometrical alignement
Colour equalization
Blending
Alignment in presence of parallax
Alignment in presence of perturbations
Rendering

Figure 1.3: Global procedure to create a panorama. Starting from a set of pictures, the picture-to-picture relationship is computed. Then, the picture positions are determined globally, the picture get blended and finally rendered.

Chapter 2

Geometric Transformations and Motion Estimation

The fundamental idea behind image mosaicking is to use the information that appears in several images simultaneously to reconstruct a better or bigger image. The following chapter exposes the basic theory of image formation from a geometric viewpoint, and reviews the standard techniques used to align these pictures. The geometrical relations are used to compute the motion that a camera undergoes between two captures and allow to put the images in a common reference frame. Once the images are in the common reference frame, they are compared pixel by pixel to see if the camera motion estimate is good enough. Figure 2.1 shows an example that takes image I_1, transforms it such that it resembles image I_0, does a comparison of the pixel values, and updates the estimate of the motion of the camera until the comparison of the pixel values reaches a satisfactory level.

Before doing any computation, it is necessary to choose a model for the camera, a model for the camera motion and a model for the scene. In the choice of these models, there is a trade-off between complexity and accuracy, and the final choice depends on the specific situation. We will not try to formulate any computable criteria to choose among these models. We will also make a basic assumption for the rest of the text, namely, that *the scene is static*, i.e. nothing moves in the scene during the whole capturing process. In more pragmatic terms, it means that moving objects will be considered as noise. Motion will refer to the *camera motion*, unless stated otherwise.

Among the estimation methods to align the pictures, we can distinguish two groups: Feature based method, and methods based on optical flow. In the former, the correspondence between some points in the images are supposed to be known. For example, in Figure 2.2, the position of the church is supposed to be known in the left *and* in the right image. In methods based on optical flow, an initial overlap position for each image pair suffices to solve the estimation problem. This work focuses on parametric method based on optical flow. The methods using correspondences are nonetheless presented here, because most of the problems related to them will appear in the parametric methods as well.

The chapter starts by presenting a camera model - the pinhole model - that is used to project a scene from three dimensional space onto an image. Then, given an image pair, it studies where a point in space can appear on the pair, i.e., it derives some rules that relate the position of the point in one image with respect to its location on the other image. Then, these

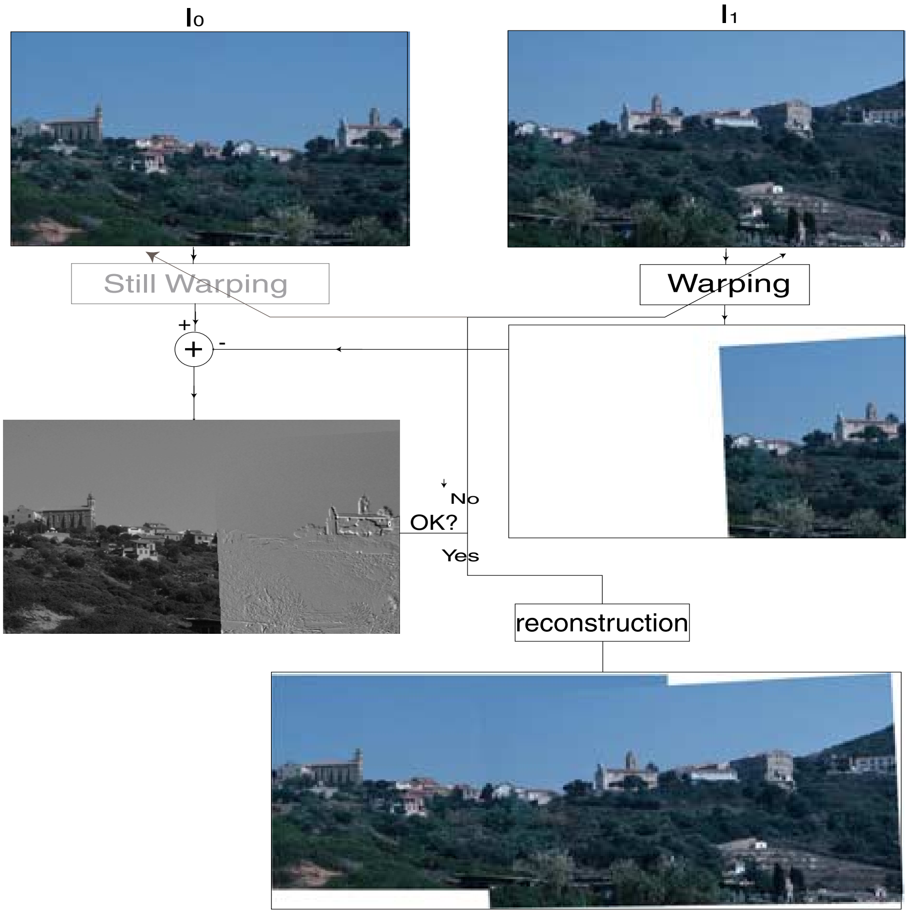


Figure 2.1: Principle of motion estimation: image I_1 gets transformed according to the motion parameters, and compared to image I_0. If the comparison is satisfactory, then a mosaic can be reconstructed, otherwise the motion parameters have to be corrected. In some situations, for example when correcting for lens distortion or for colour mismatch, I_0 also gets transformed, but this transformation does not involve the motion of the camera.

rules gets restricted to the case where all the points in space are confined on a plane. A couple of *camera motion models* are presented that relate the position of a point in an image directly to its location on the other image, without computing its position in the scene. These models make the assumption that the scene is confined to a plane, to several planes, or to the plane at infinity. These models are referred to as *camera motion models*, because some of them express in an explicit way the motion that the camera undergoes during the capture process. To conclude the camera model section, an extension to the pinhole model is presented. The feature-based motion estimation methods are reviewed in a following section, followed by the parametric methods based on optical flow. Compared to the standard ones, these parametric methods have been extended to be used in the context of Chapter 5, that deals with colour correction.

2.1 Notations and terminology

Let us first define the notation used in the text:

- Capital letters $(\mathbf{X}, \mathbf{T})$ are used to denote matrices and vectors in three dimensional ($3D$) space. Lower-case letters $(\mathbf{p}, \mathbf{q})$ are used to denote two dimensional ($2D$) vectors, or vectors in $2D$ projective coordinate space. Boldface $(\mathbf{p}, \mathbf{R})$ is used for vectors and matrices, and normal typeface (X_1, X_2, u, f) is used for numbers and vector components.
- $\mathbf{X}$ refers to the $3D$ coordinate of a point. See Figure 2.2
- (u, v) is used to denote the pixel coordinate in the image.
- f refers to the focal length of the image.
- $\mathbf{q}$ refers to the normalised $2D$ projective coordinates of a point in the image, hence $q_1 = u/f,\ q_2 = v/f,\ q_3 = 1$.
- $\mathbf{p}$ refers to the $2D$ projective coordinates of a point in the image: $\mathbf{p} = \mathbf{q} \cdot f$, thus $p_1 = u,\ p_2 = v,\ p_3 = f$. For convenience, we will sometimes use $\mathbf{p}$ instead of (u, v) to denote the $2D$ Euclidean coordinate in the image.

Most of the techniques presented here work with an image couple: image I_0 and image I_1. Image I_0 is called the *still image* or *target image*. Image I_1 gets transformed into the *warped image* that resembles - if the estimation is correct - the *target image* I_0. In other words, the warping process tries to compute how would the part of the scene in image I_1 look like, if it would have been taken by the camera in the position of image I_0. Since the scene did not change between the time image I_0 and I_1 were taken, the warped image should be identical to image I_0 in the part of the scene that is contained in both images.

In some situations, for example when correcting for lens distortion or for colour mismatch, the still image I_0 gets also transformed and becomes the *target image*. The reason we refer to it as *still image* is that this last transformation does not involve any motion parameters of the camera.

$\mathbf{U}_\theta : \mathbf{p} \to \mathbf{p}'$ denotes the image-to-image coordinate transformation, also referred to as the *motion model*. By definition, the pixel at coordinate $\mathbf{p}$ in the warped image is the same as the pixel at coordinate $\mathbf{p}'$ in image I_1 - up to a possible interpolation if $\mathbf{p}$ and $\mathbf{p}'$ have real valued coordinates.

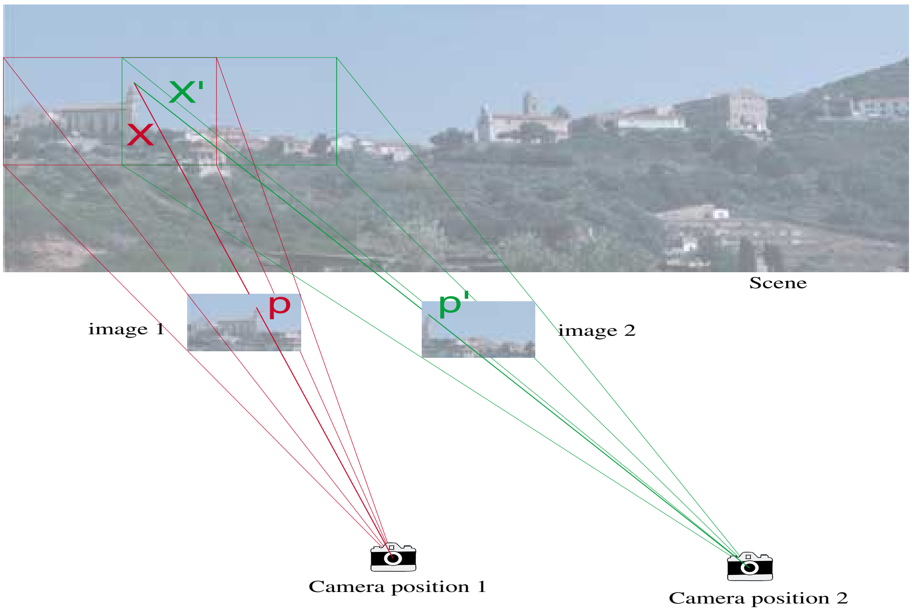


Figure 2.2: Illustration of the image capture process. The scene is depicted in the background and is pictured into two images using a camera at two different position. $\mathbf{X}$ denotes the $3D$ coordinate of the points in space, expressed in the coordinate system of the camera.

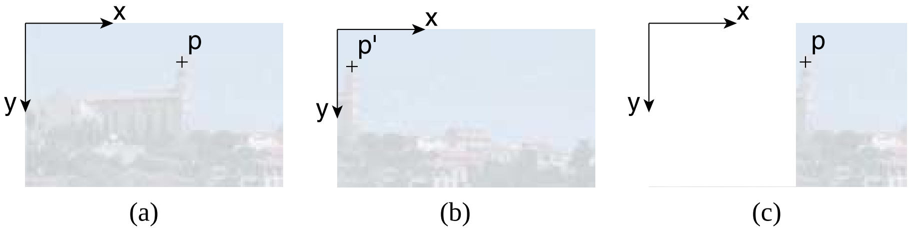


Figure 2.3: 2D coordinates transformation illustration. (a) target image I_0 (b) image to be warped I_1 (c) warped image. The object in position $\mathbf{p}$ in the *target image* is the same as the object in position $\mathbf{p}'$ in the image to be warped. The warping process puts both images in the same coordinate system, and allows a pixel-wise comparison.

2.2 The pinhole camera model

An ideal camera - or *pinhole camera* - creates a projective mapping of the scene onto a plane. In a well chosen coordinate system, the mapping can be described by

$$\mathbf{p} = f \cdot \begin{bmatrix} \frac{X_1}{X_3} \\ \frac{X_2}{X_3} \end{bmatrix}, \tag{2.1}$$

where the vector $\mathbf{X}$ contains the $3D$ coordinates of a point and $\mathbf{p}$ contains the corresponding coordinates in the image (see Figure 2.2). f is, by definition, the *focal length* of the camera and should be expressed in pixel units. For example, if the focal lens f of the camera is equal to 75 mm, in a 35 mm camera that outputs an image that is 1000 pixels wide, then $f = 1000 \cdot 75/35$ pixels.

The pinhole model applies to cameras with good quality and fixed focal lenses. If the focal length is too small (for example a 24mm lens), or if the camera uses a tele-objective lens (with a variable focal length), and also if the lens is cheap, like on most of the off-the-shelf digital cameras, then the pinhole model becomes approximate. In such a case, *distortion* should be taken into account, as presented in Section 2.4.6.

2.3 Coordinate constraints on the points in the scene

Using a pinhole camera model, it is possible to express a constraint on the position that a particular object of the scene has in an image pair. Section 2.3.1 expresses a constraint on a point with arbitrary location in the scene, and Section 2.3.2 focuses on the relations of co-planar points.

The image pair is taken by a camera that undergoes an arbitrary motion between the first and the second capture. This motion can be expressed by a rotation[1] followed by a translation. We can relate the coordinates of a single point $\mathbf{X}$ in $3D$ *space* expressed from 2 different viewpoints as[2] [Faugeras, 1993; Maybank, 1992]

$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{T} \tag{2.2}$$

where $\mathbf{T}, \mathbf{X}, \mathbf{X}' \in \mathbb{R}^3, \mathbf{R} \in \mathbb{R}^{3\times3}$. $\mathbf{R}$ and $\mathbf{T}$ describe the rotation and translation of the camera and $\mathbf{X}'$ represents the same point $\mathbf{X}$ in space expressed in the other coordinate frame of the image pair.

2.3.1 The epipolar constraint of points in image space

If the camera motion between two images is known, a point in one of the image can only be on a certain line in the other image: namely on the *epipolar line*. This rule is known as the *epipolar constraint* [Faugeras, 1993; Maybank, 1992]. The epipolar constraint can be explained as follow: Consider Figure 2.2: the optical centres $\mathbf{C}$ and $\mathbf{C}'$, the church positions

[1] The *rotation* denotes a $3D$ rotation defined by three angles: *pan*, *tilt* and *roll*.

[2] Half of the textbooks use equation $\mathbf{X}' = \mathbf{R}(\mathbf{X} + \mathbf{T})$, thus leading to slightly different equations.

$\mathbf{p}$ and $\mathbf{p}'$ in the images as well as the church position $\mathbf{X}$ or $\mathbf{X}'$ in space are co-planar. Thus, expressing every point in the coordinate system of image I_0,

$$\left\| \mathbf{R^T X}' \cdot (\mathbf{T} \times \mathbf{X}) \right\| = 0 \tag{2.3}$$

where $\times$ denotes the cross product (defined in equation 2.4) and $\cdot$ denotes the inner product. $\mathbf{R}$ is the rotation of the camera, and $\mathbf{T}$ is the translation of the camera. Rewriting equation (2.3) using only matrix multiplications leads to

$$\mathbf{X}'^T \mathbf{R} \mathbf{T}_s \mathbf{X} = 0,$$

where $\mathbf{T}_s$ is the 3 × 3 skew symmetric matrix such that

$$\mathbf{T}_s \mathbf{X} \triangleq \mathbf{T} \times \mathbf{X} \quad \forall \mathbf{X} \in \mathbb{R}^3 : \mathbf{T}_s = \begin{bmatrix} 0 & -T_3 & T_2 \\ T_3 & 0 & -T_1 \\ -T_2 & T_1 & 0 \end{bmatrix}. \tag{2.4}$$

By rescaling the equation by $X_3 X_3'$ we get

$$\mathbf{q}'^T \mathbf{E} \mathbf{q} = 0, \tag{2.5}$$

where $\mathbf{E} = \mathbf{R}\mathbf{T}_s$ is called the *essential matrix*. $\mathbf{E}$ is a 3 × 3 matrix of rank 2 and its span has dimension 8 (see [Maybank, 1992] for the details). The rank deficiency comes from a scale indetermination in the motion problem [Faugeras, 1993], as illustrated in Section 2.4.4.

2.3.2 Geometrical relations of planes in space

This section describes the geometry of a plane in space pictured from two locations and defined by a set of the two-dimensional correspondences $(\mathbf{p}, \mathbf{p}')$. It has been extensively studied in the literature, and we refer the reader to [Longuet-Higgins, 1986; Weng et al., 1991; Kanatani, 1995]. The difference with the previous Section 2.3.1 lies in the fact that all the points $\mathbf{X}$ are now confined on a plane. The goal is to retrieve the rotation $\mathbf{R}$ and translation $\mathbf{T}$ defined in equation (2.2). Rigid motion can be described using the *fundamental matrix*, also called *essential matrix* when the coordinates are normalised[3]. If the scene is composed of a single plane, the span of the elements of $\mathbf{E}$ in equation (2.5) has a lower dimension (see Section 2.5.1), so that the estimation techniques will differ from the general case. Let us define a plane with

$$\mathbf{N}^T \mathbf{X} = 1 \tag{2.6}$$

where $\mathbf{X}$ is the point coordinate in space and $\mathbf{N}$ a vector orthogonal to the plane. Thus, equation (2.2) becomes

$$\mathbf{X}' = (\mathbf{R} + \mathbf{T}\mathbf{N}^T)\mathbf{X} \tag{2.7}$$

or equivalently

$$\mathbf{X}' = \mathbf{H}\mathbf{X}.$$

It has been shown that the matrix $\mathbf{H}$ can be determined by 4 points correspondences, if no 3 points are co-linear to each other [Hu and Ahuja, 1995].

[3] The coordinates are normalised when the focal length is equal to one.

2.3.3 Ambiguity

If a dense set of correspondences is available, there is in general not a unique solution to the motion equation (2.7), namely:

- If the camera translation is perpendicular to the plane, the solution is unique, up to a scale factor [Maybank, 1992].
- If there is no translation in the motion ($\mathbf{T} = 0$ in equation 2.7), then the plane is undetermined but the motion is unique.
- Otherwise there are always two solutions to the motion. Knowing one solution, we can easily determine the other one [Maybank, 1992].
- There is still a very peculiar case where there is an infinite number of solutions. This situation arises when one of the pictures is taken through a mirror, leading to $\det(\mathbf{H}) < 0$ (in equation 2.7) [Hu and Ahuja, 1995]. We will not consider this situation.

2.4 Camera motion models

This section exposes the modelling used to transform or *warp* the images. As exposed on Figure 2.1, the warping process is the operation that allows to put two images into the same reference frame, and thus, allowing a pixel-wise comparison between these images. The motion model defines the set of all possible images that can be obtained from a single image through the warping process. This set of images is also sometimes referred to as the *orbit* of the reference image [Mann and Picard, 1995]. The motion model is chosen according to how the images have been taken.

2.4.1 The rotation model

The rotation model is the appropriate model to describe the transformation that an image undergoes when the camera is rotating around its optical centre (i.e. the camera is not translating). In practice this is the model used to stitch a $360°$ panoramic image. The relation between the points in two images is given by

$$\mathbf{X}' = \mathbf{R}\mathbf{X}, \tag{2.8}$$

where $\mathbf{X}, \mathbf{X}' \in \mathbb{R}^3, \mathbf{R} \in \mathbb{R}^{3\times3}$. $\mathbf{X}$ and $\mathbf{X}'$ are the coordinates of a single point in space expressed in two different coordinate systems, as shown on Figure 2.2. $\mathbf{R}$ is the rotation matrix of the camera and can be written as the product of three rotation angles: *Tilt*, *Pan* and *Roll* [Craig, 1998, chap. 2]:

$$\mathbf{R}' = \mathbf{R}_r \cdot \mathbf{R}_p \cdot \mathbf{R}_t$$

$$\mathbf{R}_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos t & -\sin t \\ 0 & \sin t & \cos t \end{bmatrix}, \quad \mathbf{R}_p = \begin{bmatrix} \cos p & 0 & \sin p \\ 0 & 1 & 0 \\ -\sin p & 0 & \cos p \end{bmatrix},$$

$$\mathbf{R}_r = \begin{bmatrix} \cos r & -\sin r & 0 \\ \sin r & \cos r & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $\mathbf{R}_t$ represents a *tilting*, $\mathbf{R}_p$ represents a *panning*, and $\mathbf{R}_r$ a *rolling* transformation. The image coordinates $\mathbf{p}$ of $\mathbf{X}$ are described by equation (2.1). More explicitly:

$$\begin{aligned} p'_1 &= f' \frac{R_{11} \cdot p_1 + R_{12} \cdot p_2 + R_{13} \cdot f}{R_{31} \cdot p_1 + R_{32} \cdot p_2 + R_{33} \cdot f}, \\ p'_2 &= f' \frac{R_{21} \cdot p_1 + R_{22} \cdot p_2 + R_{23} \cdot f}{R_{31} \cdot p_1 + R_{32} \cdot p_2 + R_{33} \cdot f}, \end{aligned}$$

where f and f' are the focal lengths of the camera while taking the images.

2.4.2 The roto-translation model

This model is used to describe an arbitrary motion of a pin-hole camera. The most general description of the motion can be modelled as a rotation followed by a translation. If the camera motion and the positions of the objects in the scene are known, their relative positions with respect to the camera as well as their positions in the images can be computed. We can relate these coordinates in $3D$ *space* expressed from 2 different viewpoints as

$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{T}$$

where $\mathbf{T}, \mathbf{X}, \mathbf{X}' \in \mathbb{R}^3, \mathbf{R} \in \mathbb{R}^{3\times 3}$. $\mathbf{R}$ and $\mathbf{T}$ describe the rotation and translation of the camera. The image coordinates $\mathbf{p}$ of $\mathbf{X}$ are described by equation (2.1). More explicitly:

$$\begin{aligned} p'_1 &= \frac{R_{11} \cdot X_1 + R_{12} \cdot X_2 + R_{13} \cdot X_3 + T_1}{R_{31} \cdot X_1 + R_{32} \cdot X_2 + R_{33} \cdot X_3 + T_3} \\ p'_2 &= \frac{R_{21} \cdot X_1 + R_{22} \cdot X_2 + R_{23} \cdot X_3 + T_2}{R_{31} \cdot X_1 + R_{32} \cdot X_2 + R_{33} \cdot X_3 + T_3}. \end{aligned} \tag{2.9}$$

Thus, $\mathbf{p}$ depends on the depth of each point (X_3) that is actually not contained in the images. To be able to put an image in the same reference frame as the reference image, some suppositions about the depth of each pixel are to be made, that is, a model of the scene is needed.

2.4.3 The roto-translation planar model

This model is equivalent to the roto-translation model restricted to planar surfaces. In opposition with the roto-translation model, it does not require explicitly to know the depth of each point (i.e. coordinate X_3 in equation 2.9) to warp an image. As described in Section 2.3.2, the relations of two planes in space can be expressed as

$$\mathbf{X}' = (\mathbf{R} + \mathbf{T}\mathbf{N}^T)\mathbf{X}$$

where $\mathbf{R}$ is the camera rotation matrix, $\mathbf{T}$ the camera translation and $\mathbf{N}$ the perpendicular to the plane. Using the roto-translation planar model, the coordinates of a point $\mathbf{p}'$ in image I_0 can be computed as

$$
\begin{aligned}
p_1' &= f' \frac{R_{11} \cdot p_1 + R_{12} \cdot p_2 + R_{13} \cdot f + T_1 \cdot (\mathbf{N}^T \cdot \mathbf{P})}{R_{31} \cdot p_1 + R_{32} \cdot p_2 + R_{33} \cdot f + T_3 \cdot (\mathbf{N}^T \cdot \mathbf{P})} \\
p_2' &= f' \frac{R_{21} \cdot p_1 + R_{22} \cdot p_2 + R_{23} \cdot f + T_2 \cdot (\mathbf{N}^T \cdot \mathbf{P})}{R_{31} \cdot p_1 + R_{32} \cdot p_2 + R_{33} \cdot f + T_3 \cdot (\mathbf{N}^T \cdot \mathbf{P})}
\end{aligned}
\tag{2.10}
$$

where f and f' are the focal lengths of both images, $\mathbf{P} \triangleq [p_1, p_2, f]^T$. Because of the scale indetermination of the problem, $\mathbf{T}$ is defined up to a scale factor, and its norm can be arbitrarily set to $||\mathbf{T}|| = 1$. Thus $\mathbf{T}$ can be rewritten as

$$
\mathbf{T} = \begin{pmatrix} \cos(\alpha)\cos(\beta) \\ \sin(\alpha)\cos(\beta) \\ \sin(\beta) \end{pmatrix}. \tag{2.11}
$$

The whole motion can be described with 3 rotation angles for matrix $\mathbf{R}$, 3 parameters for $\mathbf{N}$, 2 parameters for $\mathbf{T}$ and the 2 focal lengths, making a total of 10 parameters. This description is redundant since the same transformation can be described using only 8 parameters, as suggested by equation (2.12). This suggests that a straight forward implementation of this model will lead to poor results (see Section 2.6.2).

2.4.4 The homography model

An homography can be described by a 3×3 matrix:

$$
\mathbf{X}' = \mathbf{K}\mathbf{X}, \tag{2.12}
$$

where $\mathbf{X}, \mathbf{X}' \in \mathbb{R}^3, \mathbf{K} \in \mathbb{R}^{3\times3}$. $\mathbf{X}$ and $\mathbf{X}'$ are the coordinates of a single point in space expressed in two different coordinate systems, as shown on Figure 2.2. The image coordinates $\mathbf{p}$ of $\mathbf{X}$ are described by equation (2.1). More explicitly:

$$
\begin{aligned}
p_1' &= p_3' \frac{K_{11} \cdot X_1 + K_{12} \cdot X_2 + K_{13} \cdot X_3}{K_{31} \cdot X_1 + K_{32} \cdot X_2 + K_{33} \cdot X_3} \\
p_2' &= p_3' \frac{K_{21} \cdot X_1 + K_{22} \cdot X_2 + K_{23} \cdot X_3}{K_{31} \cdot X_1 + K_{32} \cdot X_2 + K_{33} \cdot X_3}.
\end{aligned}
\tag{2.13}
$$

$\mathbf{K}$ can be defined by 8 parameters ($k_1 ... k_8$), by setting arbitrarily $K_{33} = 1$. This last setting is allowed because of the scale indetermination property of the problem. Indeed, if everything is scaled by a factor S in the scene, i.e. $\tilde{X} = SX, \quad \tilde{X}' = SX'$, the resulting image does not change, or equivalently, the coordinated p_1' and p_2' of equation (2.13) remain the same. For the same reason, we can set - somehow arbitrarily - $X_3 = f$ and $p_3' = 1$. Finally, equation (2.13) becomes

$$
\begin{aligned}
p_1' &= \frac{K_{11} \cdot p_1 + K_{12} \cdot p_2 + K_{13}'}{K_{31} \cdot p_1 + K_{32} \cdot p_2 + 1} \\
p_2' &= \frac{K_{21} \cdot p_1 + K_{22} \cdot p_2 + K_{23}'}{K_{31} \cdot p_1 + K_{32} \cdot p_2 + 1}.
\end{aligned}
\tag{2.14}
$$

The homography model is able to exactly capture all the transformations of a planar surface pictured by a pin-hole camera in an arbitrary position. Indeed, by comparing equation (2.14) to equation (2.10) that describes explicitly the relation of a planar surface pictured from an arbitrary position, we can see that they can be equivalent.

The difference between this formulation and the roto-translation planar model rests in the way the parameters are expressed: the roto-translation planar model is composed of the physical motion parameters of the camera and allows to distinguish the camera motion from the plane position, whereas the homography is composed by the K_{ij} parameters of equation (2.12) that have no specific physical meaning. Having to deal with the physical parameters has one advantage: It allows to impose a constraint on the parameters related to a physical constraint - for example when one wants to express the motion of two planes in the scene, having the same camera motion associated to it. This argument is further developed in Section 3.3.2.

2.4.5 The affine model

The affine model is an approximation of the homography model, where some of the terms of the matrix $\mathbf{K}$ in equation (2.12) are set to zero. It is expressed as

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}, \tag{2.15}$$

where A_{ij} are the scale parameters of the model and T_i are the translation parameters of the model. The affine model is a pure $2D$ model, since no depth terms are involved in the equation. It is presented here because it is still widely used in practice, and has several advantages. The first is its ease of implementation. The second is that it is a linear model in terms of the image coordinates: The motion estimation algorithm of Section 2.6.2 does a first order approximation that does not affect the affine model. This gives the affine model excellent convergence properties. The affine model is an approximation of an homography for small motions, and has been extensively used in video applications. As for photographic application, is has only little impact, except for intermediate estimation steps where its good convergence properties are used.

2.4.6 Lens distortion modelling

Until now, it has been assumed that the camera behaves like a pinhole camera, i.e. following equation 2.1. In practice, there is a little distortion in the geometry of the image formation. Fundamentally, the image formation process is governed by Maxwell equations [Saleh and Teich, 1991]. From there, two interesting analytical results can be derived [Walther, 1995, p. 26]. One is that a perfect projective image is a solution to Maxwell equations (i.e. an image where each point of a planar scene is pictured in a single point of the image plane following the pin-hole model). The other is that if the imaging system is symmetric around a plane perpendicular to the optical axis, and provided that the system produces a perfect image, then the mapping of the points in the scene follow:

$$\mathbf{p} = f \cdot \begin{bmatrix} \frac{X_1}{\sqrt{X_1^2+X_3^2}} \\ \frac{X_2}{\sqrt{X_2^2+X_3^2}} \end{bmatrix} \tag{2.16}$$

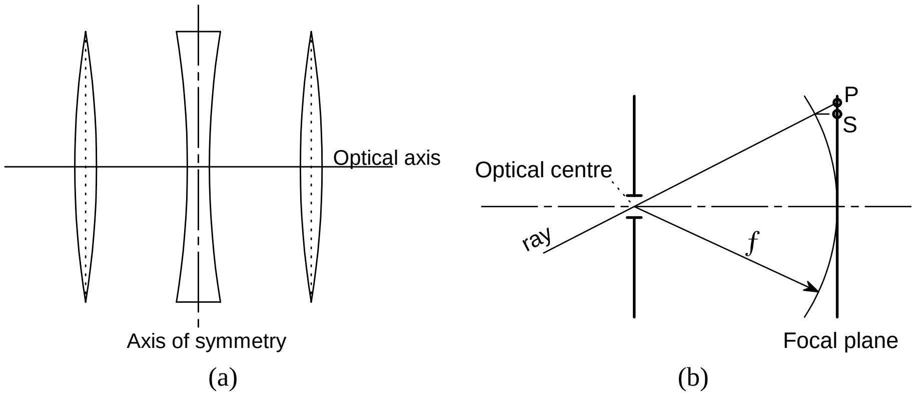


Figure 2.4: (a) Illustration of a symmetrical optical system. The axis of symmetry is perpendicular to the optical axis. (b) The two solutions of a perfect imaging system: If the camera follows the pin-hole model, the incoming ray generates a point at location **P**, whereas if the camera is a symmetrical optical system, the ray falls on point **S**.

In other words, the rays are mapped to the sinus of the incoming ray angle rather than to the tangent, as illustrated in Figure 2.4.

In practice, there is no such perfect imaging system, i.e., each point of the scene is mapped on a small surface of the image plane. Furthermore, the optical systems are not symmetric, because the distortion introduced by equation (2.16) would be unacceptable. Some distortion is unfortunately present in an image. The only thing we can say about it is that the distortion tends to be symmetric around the optical axes. The common way to model the distortion is through a Taylor series expansion around the origin of the unknown distortion function. Since the function is symmetric, the simplest distortion model leads to:

$$\begin{aligned}\mathbf{p} &= f \cdot \begin{bmatrix} \frac{X_1}{X_3} \\ \frac{X_2}{X_3} \end{bmatrix} \cdot (1 + \rho \cdot r^2) \\ r^2 &\triangleq \frac{X_1^2 + X_2^2}{X_3^2} \cdot f^2,\end{aligned}$$

where ρ is the unknown parameter. In two dimensions, the same distortion correction can be expressed as

$$\mathbf{V}_\rho : \begin{pmatrix} u \\ v \end{pmatrix} \longmapsto \begin{pmatrix} u \\ v \end{pmatrix} \cdot \left(1 + \rho \cdot \left[u^2 + v^2\right]\right).$$

This last equation is the one that is used to correct for lens distortion in Section 2.6.2. It has a single parameter: ρ.

2.5 Estimation methods based on correspondences

2.5.1 General linear methods

The first aim of the estimation method is to determine the various camera positions and orientation that have been used to take the pictures. Given the image correspondences $(\mathbf{p}, \mathbf{p}\,')$, Longuet-Higgins [Longuet-Higgins, 1981] proposed a linear method to find the essential matrix $\mathbf{E}$ of equation 2.5:

$$\mathbf{q}'^T \mathbf{E} \mathbf{q} = 0,$$

where $(\mathbf{q}, \mathbf{q}')$ are the normalised correspondences $(\mathbf{p}, \mathbf{p}')$. $\mathbf{E}$ is a 3×3 matrix, has rank 2 and is solved using the following (equivalent) system of equations

$$\mathbf{Q} \cdot [\mathbf{E}]_{9 \times 1} = \mathbf{0}, \tag{2.17}$$

where $\mathbf{Q}$ is a $9 \times n$ matrix composed by the elements $q_i q'_j$. $[\mathbf{E}]_{9 \times 1}$ has the elements of matrix $\mathbf{E}$ re-ordered on a single column. The rank of $\mathbf{Q}$ is equal to 8 in general, and a singular value decomposition (SVD) [Golup and van Loan, 1996] is used to solve the system. $[\mathbf{E}]_{\,9 \times 1}$ is given by the eigenvector of $\mathbf{Q}$ associated with the smallest eigenvalue. This algorithm is known as the *eight points algorithm*. Eight correspondences are required to solve the system, but in practice more of them are used in order to filter out the noise

From $\mathbf{E}$, it is then possible to retrieve the rotation and the translation up to a scale factor. Hartley [Hartley, 1992] extended the method for estimating the focal lengths using a similar approach.

Stability of the linear estimation

Among the critiques of the 8 points algorithm, the most relevant is its sensitivity to noise in the feature location. A multitude of algorithms have been proposed to be more robust to noise. Most of these algorithms perform a non-linear optimisation on the feature location using some robust metric; see [Luong et al., 1993] and [Zhang, 1996] for a comparison of these techniques. Then, Hartley [Hartley, 1997] presented a modified version of the eight-points algorithm, denoted as the *normalised eight points algorithm*. Without entering in the gory details, the normalised algorithm performs a change of coordinate system prior to the essential matrix computation. This change of coordinate system avoids that any given coordinate takes arbitrarilly more importance than the others in the approximations involved in the motion computation. The result of Hartley lead to a simple algorithm which is nearly as accurate as the best non-linear technique.

Retrieving the motion parameters from the essential matrix

The essential matrix should have one eigenvalue equal to zero and two other non-zero equal eigenvalues. The first step of the computation finds a matrix $\hat{\mathbf{E}}$ that match these criteria and is “as close as possible” to the matrix found by solving (2.17). Somehow arbitrarily, and for ease of computation, the distance criterion used to find $\hat{\mathbf{E}}$ is the Frobenuis norm, i.e. the sum of the squares of the elements of the matrix. In other words, $\hat{\mathbf{E}}$ is found by solving

$$\hat{\mathbf{E}} = \arg\min(\hat{E}_{ij} - E_{ij})^2$$

provided that $\hat{\mathbf{E}}$ has one eigenvalue equal to zero and two other non-zero equal eigenvalues. This choice is discussed in [Hartley, 1997; Kanatani, 1991].

Let $\mathbf{R}$ and $\mathbf{T}$ be, respectively, the rotation and translation of the camera between two pictures, as presented in Section 2.4. Since $\mathbf{E} = \mathbf{R}\mathbf{T}_s$ and $\mathbf{T}_s\mathbf{X} = \mathbf{T} \times \mathbf{X}$, then $\mathbf{E}\mathbf{T} = \mathbf{0}$. $\mathbf{T}$ can be retrieved by taking the cross-product of any two columns of $\mathbf{E}$, by keeping in mind that $\mathbf{T}$ can only be retrieved up to a scale factor because of the scale indetermination related to any two-pictures correspondence problem. $\mathbf{R}$ can be found using

$$\mathbf{R} = \mathbf{U}\mathbf{R}_s\mathbf{V}^T \textit{ or } \mathbf{R} = \mathbf{U}\mathbf{R}_s^T\mathbf{V}^T,\ \mathbf{R}_s = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \hat{\mathbf{E}} = \mathbf{U}\mathbf{D}\mathbf{V}^T,$$

where $\mathbf{U}\mathbf{D}\mathbf{V}^T$ is the Singular Value Decomposition (*SVD*) of the essential matrix $\hat{\mathbf{E}}$. This leads to 4 possible solutions to the motion of the camera (2 different rotations, and two possible signs for the translation). Sometimes, it is possible to discriminate between the solutions by imposing that the scene is in front of the camera (and not behind).

2.5.2 Linear method for planar motion

In the planar case, the space spanned by the elements of matrix $\mathbf{E}$ has dimension 6. Thus, the usual linear methods cannot be applied. Tsai [Tsai and Huang, 1981] and equivalently Longuet-Higgins [Longuet-Higgins, 1986] proposed an algorithm similar to the eight points algorithm adapted to the planar motion estimation problem, using a set of linear techniques. Weng [Weng et al., 1991] added an error estimation to a similar motion estimation technique, in order to measure the uncertainty on the motion parameters. Kanatani [Kanatani, 1995] proposed a normalisation technique that differs from the straight-forward MMSE criterion in parameter space used in [Tsai and Huang, 1981; Longuet-Higgins, 1986; Weng et al., 1991] so that the error criterion corresponds to the euclidian distance of the feature points to the computed plane. Lee [Lee, 1990] proposed a technique to recover the plane information when only 4 points are available.

In general, the techniques used in the planar case estimate the matrix $\mathbf{K}$ of equation (2.7). To perform the estimation, a coplanarity condition is used:

$$\mathbf{X}' \times \mathbf{K}\mathbf{X} = 0,$$

thus, by doing a rescaling by X_3 and X_3'

$$\mathbf{q}' \times \mathbf{K}\mathbf{q} = 0.$$

From there, the estimation of the camera motion and plane orientation is computed [Longuet-Higgins, 1986; Weng et al., 1991]. Unfortunately, the estimation of $\mathbf{K}$ is quite sensitive to measurement noise, as it was for the essential matrix.

2.5.3 Feature extraction methods

The feature extraction is a mean to find “interesting” points $\mathbf{p}_i$ in an image in order to be able to find the corresponding points $\mathbf{p}_i'$ in the other image (i.e. the location in the other image where the same object of the scene is pictured). There are fundamentally two strategies to extract some features from the images: the first extracts some edges with a corner detector and then retain the feature according to a criterion based on these edges (for example the points at intersections). The second group of methods work directly with image luminance. The criterion of selection is generally based on the image derivatives, see Deriche [Deriche and Giraudon, 1993] for a review.

Zhang [Zhang et al., 1994] proposed a technique to find some feature points matches between two images. It first finds the points with a big curvature using the operator $R(x, y) = \det(\mathbf{C}) - k \cdot trace^2(\mathbf{C})$ where $\mathbf{C}_{2\times2}$ is the second derivative matrix of the image luminance. Then, for each point of interest, it looks for the corresponding match in the other image, using a search window whose sizen is based on some a-priori knowledge about the motion. The score of the match is computed using the cross-correlation of a window around the feature, normalised by the intensity variance of the square patch. In order to see if a match is correct, there should be a couple of other matches close to the one found whose distance to it is equal in both images (i.e. consistent with the image motion). The algorithm makes a weighted sum of these matches around the main match to keep or reject the current match, by using a threshold value.

2.5.4 Non-linear methods

Non-linear methods estimate the motion parameters directly, without estimating the essential matrix $\mathbf{E}$ or matrix $\mathbf{K}$. They use a gradient descent algorithm - for example the Gauss-Newton or the Levenberg-Marquardt algorithm [Seber and Wild, 1989] - applied to the parameters of equations (2.7) and (2.6), which are non-linear with respect to the image coordinates. The rotation $\mathbf{R}$ can be described by 3 angular parameters; $\mathbf{T}$ is defined up to a scale and can be described with only 2 parameters, as shown on equation (2.11). The 2 focal lengths are necessary to get the image coordinates. If the method converges, it leads to a more accurate result than a linear method. To be able to apply the method, it requires an initial parameter set which, in general, is computed with a linear method.

2.6 Featureless estimation methods

In practice, the correspondences $(\mathbf{p}, \mathbf{p}')$ are not available. The correspondences can be computed as in Section 2.5.3, but the result is often noisy, and the final motion will be highly dependent on the image content. The advantage of a feature-based method is that it allows to compute the motion without any prior information but is less precise than the type of algorithm presented in this section. Furthermore, a feature-based method is not well suited for segmentation; a problem discussed in Section 3.3.2. Nevertheless, this kind of method is a very efficient tool to perform an initialisation for the algorithms presented in what follows, mainly because it allows the motion computation without any a-priori information about the motion.

2.6.1 Optical flow

Traditionally, optical flow techniques have been developed by Horn and Schunk [Horn, 1986] to solve the problem of translational motion between several frames in a video sequence. It assumes that each point $\mathbf{p}$ in image t is a translated version of the corresponding point in image $t + \Delta t$. The image luminance is described as

$$I(\mathbf{p}, t) = I(\mathbf{p} + \mathbf{\Delta p}, t + \Delta t), \quad \forall (\mathbf{p}, t), \tag{2.18}$$

where $\mathbf{p}$ is a two-dimensional position vector. By setting the flow velocity $\mathbf{u}_{\mathbf{p}} = \frac{\Delta \mathbf{p}}{\Delta t}$, and computing the Taylor expansion of equation (2.18) around point $(\mathbf{p}, t)$ by keeping only the first order terms leads to the well known optical-flow equation

$$\mathbf{u}_{\mathbf{p}}^T \frac{\partial I}{\partial \mathbf{p}} + \frac{\partial I}{\partial t} \simeq 0. \tag{2.19}$$

This equation has two unknowns (the two components of vector $\mathbf{u}_{\mathbf{p}}$), and has as many equations as there are pixels in the image overlap. Now, the flow $\mathbf{u}_{\mathbf{p}}$ might not be constant over the whole image. If the flow is allowed to have an arbitrary value for every pixel, the system would have twice as many unknowns as equations. There are two approaches to solve the problem: the first considers that the motion over the whole image can be described by a single motion model, which leads to the single parametric approaches of section 2.6.2. The second approach considers the motion to belong to one out of many motion models, which introduces the problem of mixed motion estimation and segmentation described in Section 3.3.1.

Remark. The first order approximation of the optical flow equation (2.19) assumes that the motion between two frame is small. This assumption is not valid in photography. Thus, photography applications always need an initial parameter estimate that can be set either by hand, or by other techniques like a feature point algorithm or an extensive initial search on a coarse version of the images.

2.6.2 Parametric methods

We will reformulate the problem introduced in Section 2.6.1 in a slightly different and more general, way: Let $I_0(\mathbf{p}), I_1(\mathbf{p}')$ be an image pair. By assuming that both images can be perfectly super-imposed by a warping process, we can write

$$I_0(\mathbf{p}) = I_1(\mathbf{p}'),$$

where $\mathbf{p}$ and $\mathbf{p}'$ denote the matching positions in the images. This supposes that the scene is static and that a particular object of the scene is mapped into the same luminance values on image I_0 and I_1. This last assumption cannot be made in photographic applications, so that the equation has to be corrected to

$$G_\theta \left[I_0(\mathbf{p}) \right] = S_\theta \left[I_1(\mathbf{p}') \right]. \tag{2.20}$$

S and G are the functions that relate the pixel value to the scene *radiance*, and θ is the parameter set of the system. These function will extensively be discussed in Chapter 5. Now, let us introduce the function

$$\mathbf{U}_\theta : \mathbf{p} \longmapsto \mathbf{p}'$$

to describe the rigid motion model that tells the system where an object of image I_0 appears in image I_1. The motion model is controlled by the set of parameters θ (that also accounts for the colour parameters of Chapter 5). Consequently, equation (2.20) becomes

$$G_\theta \left[I_0(\mathbf{p}) \right] = S_\theta \left[I_1(\mathbf{U}_\theta \left[\mathbf{p} \right]) \right].$$

Now, if lens distortion has to be taken into account, it is more convenient to describe the "motion" with two function: $\mathbf{U}$ and $\mathbf{V}$. $\mathbf{V}$ transforms the image into an undistorted image, but does not contain any of the motion parameters of the camera. Finally, the equality becomes:

$$G_\theta \left[I_0(\mathbf{V}_\theta \left[\mathbf{p} \right]) \right] = S_\theta \left[I_1(\mathbf{U}_\theta \left[\mathbf{p} \right]) \right].$$

In practice, this means that I_0 is transformed into an undistorted image by function $\mathbf{V}$ and I_1 is transformed into an undistorted image AND warped by function $\mathbf{U}$ on the undistorted version of image I_0. At any point of the two resulting image the *radiance* values should be equal.

If we suppose some smoothness in the images $I_{0,1}$ as well as in the functions $\mathbf{U}$ and $\mathbf{V}$, we can find an updating rule for θ with a standard descent algorithms [Seber and Wild, 1989] by minimizing an objective function h, which is, for example, the mean squared error function:

$$\begin{aligned} h(\theta) &= \frac{1}{n} \sum_{i=1}^{n} \rho[r_i(\theta)] = \frac{1}{n} \sum_{i=1}^{n} \|r_i[\theta]\|^2 \qquad (2.21) \\ r_i(\theta) &\triangleq \{ G_\theta \left[I_0(V_\theta \left[\mathbf{p}_i \right]) \right] - S_\theta \left[I_1(U_\theta \left[\mathbf{p}_i \right]) \right] \} \end{aligned}$$

where $\mathbf{p}_i$ are the pixel locations on the overlapping parts of the images, n the number of pixels, and r_i the error at each pixel location, also called *residual*. From a geometrical point of view, $I_1(\mathbf{U}_\theta \left[\mathbf{p}_i \right])$ is the image that has been transformed such as to resemble the other - undistorted - one; in figure 2.1 it is the image that comes out of the *warping* black box. We will come back to the choice of the function $\rho(\cdot)$ in Chapter 4. This last equation (2.21) gives the same importance to every pixel of the image. Now, because some pixels might get saturated, we add a factor W_i that allows to lower the influence of potentially saturated pixels. equation (2.21) becomes

$$\begin{aligned} h(\theta) &= \frac{1}{n} \sum_{i=1}^{n} W_i \cdot \rho[r_i(\theta)] = \frac{1}{n} \sum_{i=1}^{n} W_i \cdot \|r_i[\theta]\|^2 \qquad (2.22) \\ W_i &\triangleq W \left[I_0(V_\theta \left[\mathbf{p}_i \right]), I_1(U_\theta \left[\mathbf{p}_i \right]) \right] \\ r_i(\theta) &\triangleq \{ G_\theta \left[I_0(V_\theta \left[\mathbf{p}_i \right]) \right] - S_\theta \left[I_1(U_\theta \left[\mathbf{p}_i \right]) \right] \}. \end{aligned}$$

The choice of W_i is discussed in Chapter 5, but note that it depends only on the pixel values and not on the radiance value. The minimum of function $h(\theta)$ satisfies

$$\frac{\partial h(\theta)}{\partial \theta} = \mathbf{0}. \qquad (2.23)$$

Equation (2.23) is a non-linear system of equations, and is most of the time too complex to be solved analytically. Therefore, a recursive solution is applied: by assuming that an initial parameter set θ_0 is available, the goal is to find an updating rule such that $h(\theta + \delta\theta) < h(\theta)$, where $\delta\theta$ is the correction applied to the parameter set. By expanding function $h(\theta)$ by Taylor series, and keeping only the second order terms, we get:

$$\begin{aligned} h(\theta + \delta\theta) &\simeq h(\theta) + \frac{\partial h(\theta)}{\partial \theta^T} \delta\theta + \frac{1}{2} \delta\theta^T \frac{\partial^2 h(\theta)}{\partial \theta^T \partial \theta} \delta\theta \qquad (2.24) \\ &\triangleq h(\theta) + \mathbf{J}(\theta)^T \delta\theta + \frac{1}{2} \delta\theta^T \mathbf{H}(\theta) \delta\theta, \end{aligned}$$

where $\mathbf{J}(\theta)$ is the *jacobian* matrix and $\mathbf{H}(\theta)$ the *hessian* matrix. According to this approximation, the correction $\delta\theta$ is found by setting the derivative $\frac{\partial h(\theta+\delta\theta)}{\partial\delta\theta}$ to zero, hence

$$\mathbf{J}(\theta)^T + \mathbf{H}(\theta)\delta\theta = \mathbf{0}. \tag{2.25}$$

The correction to the parameter set θ leads to a minimum if the matrix $\mathbf{H}$ is positive definite. To rewrite equation (2.25) in terms of I_0, I_1 and θ, Let us first give a few definitions:

$$\begin{aligned}
\mathbf{U}_{\theta i} &\triangleq \mathbf{U}_\theta\left[\mathbf{p}_i\right] \\
\mathbf{V}_{\theta i} &\triangleq \mathbf{V}_\theta\left[\mathbf{p}_i\right] \\
\nabla G_\theta\left[I_0(\mathbf{V}_{\theta i})\right] &\triangleq \left.\frac{\partial G_{\mathbf{u}}\left[I_0(\mathbf{x})\right]}{\partial\mathbf{x}}\right|_{\mathbf{u}=\theta,\mathbf{x}=\mathbf{V}_\theta[\mathbf{p}_i]} \\
\nabla S_\theta\left[I_1(\mathbf{U}_{\theta i})\right] &\triangleq \left.\frac{\partial S_{\mathbf{u}}\left[I_1(\mathbf{x})\right]}{\partial\mathbf{x}}\right|_{\mathbf{u}=\theta,\mathbf{x}=\mathbf{U}_\theta[\mathbf{p}_i]} \\
\dot{G}_\theta\left[I_0(\mathbf{V}_{\theta i})\right] &\triangleq \left.\frac{\partial G_{\mathbf{u}}(v)}{\partial\mathbf{u}}\right|_{\mathbf{u}=\theta,v=I_0(\mathbf{V}_{\theta i})} \\
\dot{S}_\theta\left[I_1(\mathbf{U}_{\theta i})\right] &\triangleq \left.\frac{\partial S_{\mathbf{u}}(v)}{\partial\mathbf{u}}\right|_{\mathbf{u}=\theta,v=I_1(\mathbf{U}_{\theta i})}
\end{aligned}$$

In other words, $\nabla G_\theta\left[I_0(\mathbf{V}_{\theta i})\right]$ is the derivative along the lines and the columns of the radiance image computed from an undistorted version of image I_0. $\nabla S_\theta\left[I_1(\mathbf{U}_{\theta i})\right]$ is the derivative along the lines and the columns of the radiance image that has been warped according to the motion function $\mathbf{U}$. $\dot{G}_\theta\left[I_0(\mathbf{V}_{\theta i})\right]$ and $\dot{S}_\theta\left[I_1(\mathbf{U}_{\theta i})\right]$ are the derivative of the camera characteristics with respect to the parameters that control the characteristics (which might be different for image I_0 and I_1). Appendix A details the math involved to replace the elements in equation (2.25). The iteration to solve the motion estimation problem becomes

$$\theta_{i+1} = \theta_i + \alpha\delta\theta$$

$$W_i \cdot \frac{\dot{\rho}(r_i)}{r_i}\left\{\begin{array}{c} \nabla G_\theta\left[I_0(\mathbf{V}_{\theta i})\right]\frac{\partial\mathbf{V}_{\theta i}}{\partial\theta} + \dot{G}_\theta\left[I_0(\mathbf{V}_{\theta i})\right] - \\ -\nabla S_\theta\left[I_1(\mathbf{U}_{\theta i})\right]\frac{\partial\mathbf{U}_{\theta i}}{\partial\theta} - \dot{S}_\theta\left[I_1(\mathbf{U}_{\theta i})\right] \end{array}\right\}\delta\theta = -W_i \cdot \frac{\dot{\rho}(r_i)}{r_i} \cdot r_i \tag{2.26}$$

where only line i of the equation system is shown here. Thus, at each iteration step, a linear system of equations has to be solved. Then, an iteration, or *line search*, is performed in order to determine the factor α, by starting with $\alpha = 1$. Without being too specific about the details in An important fact to keep in mind is that equation 2.26 is more approximate than a second order approximation, and the more complex the motion model $\mathbf{U}$, the worse the convergence of the algorithm. For more details about the approximation involved, we refer the reader to Appendix A.

When reconstructing a mosaic composed of more than two pictures, the common practice consist in combining the two first images using the techniques presented so far, and registering the remaining pictures to the former image combination iteratively. This implies that from the third picture and on, the *target* picture is already distortion free and colour corrected, i.e. function $G(\cdot)$ and $V(\cdot)$ are set to identity.

2.7 Evaluation of featureless estimation methods

By applying the recursion described in the previous section by equation 2.26 on the four models presented in Section 2.4, namely the *affine*, *homography*, *rotation* and *roto-translation* *planar* model, we notice that the convergence properties differ drastically from one model to another. Although an extensive study of the convergence properties is out of the scope of this thesis, here are some hints: The most obvious cause for bad convergence are the set of approximations that are used to linearise the models. The importance of these approximations varies with the model chosen. Indeed, replacing the model by its first order approximation does not affect the *affine* model. Another problem can come from the conditioning of the equations involved in the problem, in particular equation (2.26). A particularly frequent problem is the confusion between the focal length and the lens distortion in rotational models.

2.8 The use of multi-resolution

The parametric motion estimation algorithm outlined in Section 2.6.2 is always applied using a multi-resolution approach. The multi-resolution approach consists of first estimating the motion on a coarse version of the images, and then refine the result by increasing the resolution step by step. The reason is twofold: first, using smaller images saves computation time. Second, the estimation is more likely to converge toward the global minimum. This last statement is crucial; in practice, most of the examples shown in this thesis would have ended in a local minimum if multi-resolution was ignored. To illustrate the point, Figure 2.5 shows a mosaic that is reconstructed using a translational model (i.e. the affine model of equation (2.15) with matrix A set to identity). The error is computed for a set of possible translations around the solution at three different resolutions, and is plotted on a graph. The graph shows that when the resolution decreases, the error surface becomes smoother and contains fewer local minima. It also shows that on the high resolution image, the location of the global minimum is defined with greater precision. The images in this example have been high-pass filtered to accentuate the regularisation phenomenon. If we start a motion estimation algorithm with the biggest translation error shown in the graph as initial estimate, the algorithm is likely to converge if it is started at the coarsest resolution, but would certainly fail if it is started at the highest resolution.

This regularisation phenomenon can be easily formalised by considering the correlation between the image, which is a similar measure to the mean squared error used so far. Let $I_0(z)$ and $I_1(z)$ be the z-transform of a two images (for simplicity, we are considering only one dimensional signals). Let $F(z)$ be a low-pass filter. The correlation between the two images is given by

$$C = I_0(z) \cdot I_1(z) .$$

Now, if we filter the image prior to computing the correlation, we get a new correlation surface

$$C_f = [F(z) \cdot I_0(z)] \cdot [F(z) \cdot I_1(z)] ,$$

which is equal to

$$C_f = F(z)^2 \cdot C,$$

thus, the correlation surface is equal to the initial correlation surface C filtered (twice) with the low-pass filter. In other words, filtering the images with a low-pass filter is equivalent to filtering the correlation surface with the low-pass filter. Thus, the surface becomes smoother. The same holds approximately for the mean squared error surface of Figure 2.5; the images at lower resolution being a low-pass version of the original image.

2.9 Conclusions

This chapter presented the geometry used to relate the position in an image pair of an object, pictured from arbitrary locations. It presented the pinhole model of a camera, as well as an extension that deals with the distortion introduced by the lens of the camera. The points in space are first considered in arbitrary locations, and then restricted to be on a planar surface. This restriction allows to build models that enable to compute the location of an object in an image, given its location on the other image as well as the camera motion, without explicitly computing the position of the object in space. Several models are presented, the most useful being the rotation model that allows to build full panoramic images, as will be seen in Chapter 3.

This chapter also reviewed the fundamental method to align images. It described the method based on correspondences, followed by the parametric methods based on optical flow. These last method have been extended to better suit the modelling of Chapter 4 and Chapter 5. The purpose of using a multi-resolution approach in the motion estimation has also been been discussed and illustrated by an example.

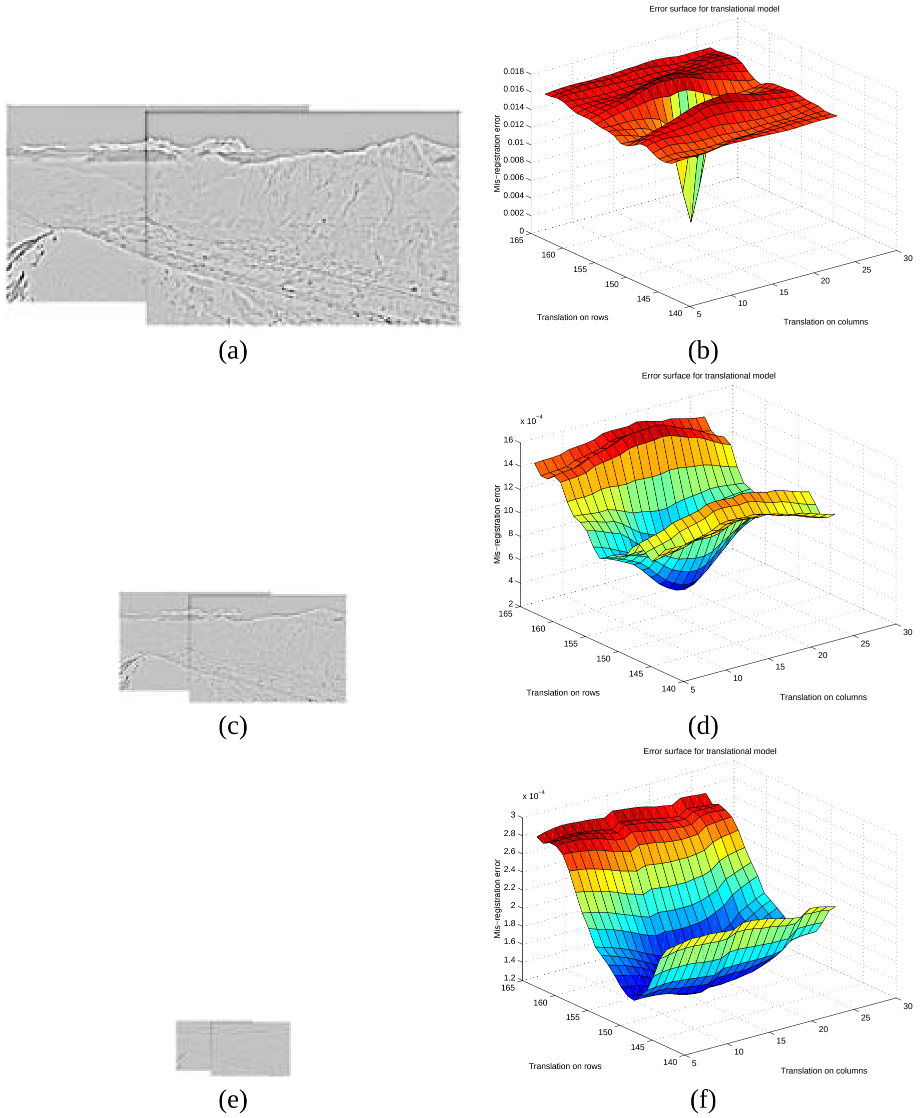


Figure 2.5: Using multi-resolution helps converging toward the global minimum. (a)(c)(e) images used to compute the error for a set of different translation parameters. (b)(d)(e) Corresponding error surfaces. The error surface is more regular if the resolution is coarse. Prior filtering on the images was used to emphasise the phenomenon.

Chapter 3

Constrained and Multiple Motion Estimation

Two major issues are addressed by this chapter: closing the gap in a 360 degrees panorama, and dealing with parallax.

To construct a 360 degrees panorama, or more generally, a mosaic where the last image overlaps with the first one, the standard methods compare an image to its neighbour, and compute the rotation of the camera. By doing it for every picture, the total estimate of the camera rotation should give 360 degrees. Nevertheless, because of small errors, this is not true in practice, and an algorithm is proposed to solve the problem. More generally, the chapter presents a way of going from an image-to-image estimation toward a global estimation method, which deals with more than two pictures simultaneously.

By taking pictures, a photographer often has to move between captures, and this results in the presence of parallax, i.e. the objects of the scene close to the camera, for example the floor, move more in the field of vision than the objects far away. The second part of the chapter addresses this problem by proposing an algorithm that sub-divides the scene into several planes.

3.1 Global estimation methods

Until now, we confined our analysis to the estimation of an image with respect to its neighbour. But what about the estimation of a whole mosaic? Any estimation technique will start with an image-to-image estimation, as exposed in the preceding sections. The purpose of this section is to address the issue of redundant estimations. For example, in a panoramic image, the camera rotation can be computed between all the image couples. If the image represents a 360° panoramic image, then the composition of all the camera rotation estimates should add-up to the identity (modulo 2π). In practice, it will not be the case, because of the cumulative estimation errors. The answer to these question is given by *constrained estimation methods*.

There are fundamentally three alternatives to apply a constraint: The first performs an optimisation by adding a cost which accounts for the the divergence from the constraint [Sawhney et al., 1998; Shum and Szeliski, 1997]. The second alternative uses the constraint to reduce the parameter set size, and the third adds some equations to the system by means of

Lagrange Multipliers [Fua and Brechtbuehler, 1996]. The first alternative does not guarantee that the constraint is satisfied and needs to carefully control the cost function in order to end-up “close enough” to the constraint. The second is better from a precision point of view, but increases the complexity and needs a starting parameter set to find iteratively a better solution. We did not consider the third alternative.

The following section presents a method that applies a constraint by reducing the parameter set size. The algorithm has two parts: the first part finds a set of parameters that meet the constraint using a linear approximation, and the second part refines the result iteratively. The key idea behind these techniques is to use a criterion that measures the change in the overlapping part of the image. The idea is to keep the overlap region of the pictures unchanged, if possible. This is done by taking a set of positions in the image overlaps, and measure how far they have to be moved in order to fulfil the constraint. Conversely, the constraint is enforced by moving this set of positions as little as possible.

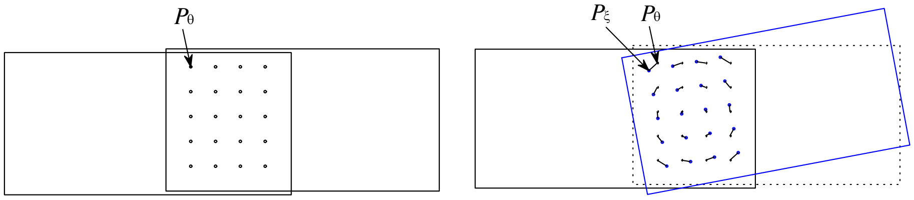
The first application improves a full panoramic image. The ill-posed parameter is the focal length that is estimated by imposing that the camera rotates by 360 degrees. The second example measures the position of two planes in space. The ill-posedness comes from an inherent ambiguity between translation and rotation [Daniilidis and Nagel, 1993]. The constraint imposes a single camera motion between the two images.

3.1.1 A simple analogy

We will explain our approach with a simple example: Let us suppose that a panorama is constructed using an image-to-image estimation, as illustrated in Figure 3.3(a). Now, let us consider an image pair from this panorama, schematised in Figure 3.1(a). On the image overlap, we choose a set of uniformly distributed points - also present in Figure 3.1(a). Now, we will try to fulfil a constraint - that tells us that the total rotation of the camera for shooting the panorama is $360 \deg$ - by keeping the overlap part of the image unchanged, if possible. We will assume that by fulfilling the constraint, the new image pair configuration is like in Figure 3.1(b) : the right image had to be tilted slightly. The error criterion to measure how much change occurred in the overlap area is given by the displacement of the set of points in the overlap area, i.e. the mean of the distances between points P_θ and P_ξ in Figure 3.1(b). The fulfilment of the constraint is performed by finding a new set of parameters that minimises this last distance for every overlap region of the panorama; this ensures that the overlap part of the images change as little as possible. In other words, this is like if we place some springs to attach the set of points to their original locations, and force the first and last image of the panorama to overlap, letting the springs distribute the displacement on every image.

3.1.2 Notations

Two frameworks are used in the context of constrained estimation. The first uses independent and unconstrained measurements - as used until now - and will be denoted by the letter θ. The parameter set under the constraint will be denoted by the letter ξ or $\mathbf{r}_i$. It is important to notice that ξ_i and θ_i represent the same parameters expressed in two different spaces.



(a) (b)

Figure 3.1: Illustration of the error criterion used to enforce a constraint on the overall picture locations. (a) Shows the original picture overlap; a set of points is chosen on the overlap. (b) Final picture location, after enforcing the constraint. The distance measure between configuration *(a)* and *(b)*, is given by mean of the distances between $\mathbf{p}_\theta$ and $\mathbf{p}_\xi$.

3.1.3 Linear optimisation on a simple constraint

Let us consider a motion model described by the two-dimensional function $\mathbf{U}_\theta$ which defines a mapping between two images (as presented in Section 2.6.2):

$$\mathbf{U}_\theta : \mathbf{p} \to \mathbf{p}'_\theta$$

where $\mathbf{p}$ and $\mathbf{p}'_\theta$ denote the matching pixel positions and θ stands for the motion parameters. The pixel value at position $\mathbf{p}$ in the warped image is the same as the one at position $\mathbf{p}'_\theta$ in the still image. In this context, the mismatch in luminance as well as lens distortion considerations are left out. Let us suppose there are 2 parameter sets θ_1 and θ_2 on which a constraint has to be imposed. The simplest constraint is the one that enforces equality of some parameters of θ, implying that they measure the same physical entity. For example, in a scene composed of two planes, one can measure the camera displacement between the two images and the plane position in space, using an algorithm designed to work with a single plane [Longuet-Higgins, 1986; Weng et al., 1991]. Applying the algorithm twice: once for plane 1, and once for plane 2, will result in parameter sets θ_1 and θ_2. The parameter sets can be rewritten in two parts: a common part θ^c and an autonomous part θ^a. In the planar motion case, the θ^c denotes the camera motion (rotation, translation and focal length), θ_1^a the position of plane 1 and θ_2^a the position of plane 2. Rewriting the expression in a new framework gives:

$$\begin{aligned}\xi_1 &= [\theta_1^c, \theta_1^a, \mathbf{0}]^T\\ \xi_2 &= [\theta_2^c, \mathbf{0}, \theta_2^a]^T,\\ \xi_i &= \xi + \mathbf{w}_i,\end{aligned}$$

where $\mathbf{w}_i$ represents the measurement - or estimation - noise and ξ the "true" parameter set. Note that both ξ_1 and ξ_2 are expressed in what is called the *constrained parameter space*, and are considered as being two noisy samples of the same physical entity ξ. In theory $\theta_1^c = \theta_2^c$. Where the parameter is independent of the measurement, the value is arbitrarily set to $\mathbf{0}$ - this value having no influence on the remaining computations. Now, given a set of overlapping positions $\mathbf{p}_i^1$ and $\mathbf{p}_i^2$, we propose to take the average position displacement in the image space as a measure of the distance between ξ_1 and ξ_2. The goal is to find the parameter set ξ that minimises this distance

$$D\left(\xi\right) = D(\xi_1, \xi) + D(\xi_2, \xi) \tag{3.1}$$

$$D(\xi_i, \xi) \triangleq \frac{1}{n_i} \sum_{j=1}^{n_i} ||\mathbf{U}_i(\mathbf{p}_j, \xi_i) - \mathbf{U}_i(\mathbf{p}_j, \xi)||^2. \tag{3.2}$$

$\mathbf{U}_i(\cdot, \xi_i)$ is the function that performs the warping according to parameter set i, and n_i is the number of pixels in the overlap area[1]. Using a first order approximation, (3.2) can be rewritten as

$$D(\xi_i, \xi) \approx \frac{1}{n} \sum_{j=1}^{n} || \frac{\partial \mathbf{U}_i}{\partial \xi}\Big|_{(\mathbf{p}_j, \xi_i)} (\xi_i - \xi)||^2. \tag{3.3}$$

Thus, the distance between the parameter sets is

$$D\left(\xi\right) = \sum_{i=1}^{2} \frac{1}{n_i} (\xi_i - \xi)^T \mathbf{E}_i^T \mathbf{E}_i (\xi_i - \xi) \tag{3.4}$$

$$\mathbf{E}_i \triangleq \frac{\partial \mathbf{U}_i}{\partial \xi}\Big|_{\xi = \xi_i},$$

where $\mathbf{E}_i$ has $2n_i$ rows, containing the derivatives along the x and y axis of the image for each pixel. To find the value of ξ that minimises the distance in (3.4), we set the derivative with respect to ξ to zero and we get

$$(\mathbf{E}_1^T \mathbf{E}_1 + \mathbf{E}_2^T \mathbf{E}_2)\xi = \mathbf{E}_1^T \mathbf{E}_1 \xi_1 + \mathbf{E}_2^T \mathbf{E}_2 \xi_2, \tag{3.5}$$

where ξ is obtained by solving the linear system of equation (3.5).

3.1.4 An example similar to a panoramic constraint

Let us start with a small example that is similar to the full panorama problem. Consider the triangle of Figure 3.2. Let us suppose that we are measuring the angles α, β, and γ with some measurement device. These values are accompanied by noise, and the result of the measurement gives $\alpha : \theta_1 = 40\,\mathrm{deg}$, $\beta : \theta_2 = 50\,\mathrm{deg}$, and $\gamma : \theta_3 = 80\,\mathrm{deg}$. We know that the sum of the angles in a triangle is equal to $180\,\mathrm{deg}$. We rewrite the three angle measurements in a global system where the constraint - the $180\,\mathrm{deg}$ angle sum constraint - is embedded, i.e. we specify the state of the triangle by giving the values of angle α and β, provided that the sum of the angle is equal to $180\,\mathrm{deg}$. In other words, we define the state of the triangle with variable ξ:

$$\xi = [\upsilon_1, \upsilon_2],$$

where υ_1 is the value of angle α and υ_2 is the value of angle β, provided that the value of angle γ is equal to $180 - \upsilon_1 - \upsilon_2$. Now, we rewrite the initial measurements in this new framework: The first measurement θ_1, which measures angle α, is rewritten as

[1] $\mathbf{U}_1(p_j, \theta)$ makes the correspondences between the two images of the pixels of plane 1 in the former example. This function extracts the parameters of plane 1 from θ.

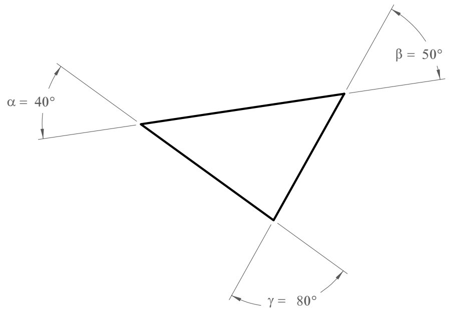


Figure 3.2: This triangle illustrates a similar problem to the alignment of a full panoramic image: The sum of the angles of the rectangle should give $180\,\mathrm{deg}$. Now what is the strategy to adopt if the measurements of the angles do not give $180\,\mathrm{deg}$?

$$\xi_1 = [\theta_1, ?] = [40, ?].$$

The measurement of angle α does not define the value of angle β, thus we left it unspecified. It turns out that the computation that follows will annihilate this value, so by convention we put a 0 instead, i.e.

$$\xi_1 = [\theta_1, 0] = [40, 0].$$

Similarly, we can rewrite the measurement of angle β as

$$\xi_2 = [0, \theta_2] = [0, 50].$$

Finally, the third measurement has to be taken into account. We have to express the measurement θ_3 of angle γ in this new framework. In practice, we have to find (υ_1, υ_2), such that $\theta_3 = 180 - \upsilon_1 - \upsilon_2$. Since there are two unknowns, and only one equation, there is an infinite number of solutions to this problem. But remember, the goal is to find the three angles of the triangle out of the noisy measurements. Our intuition tells us that the "real" values of the angles should be close to the measured ones. Since the remaining computation does a first order approximation at some point, we should express the third angle θ_3 using a representation (υ_1, υ_2) that is as close as possible to the final solution. As a guess, we choose to express it as

$$\begin{aligned}\xi_3^{(1)} &= [\upsilon_1^{(1)}, \theta_2] = [50, 50]\\ \xi_3^{(2)} &= [\theta_1, \upsilon_2^{(2)}] = [40, 60],\end{aligned}$$

by modifying only one of the angles, and leaving the other equal to a former measurement. We opt to retain two solutions from this last measurement θ_3. Note that both description $\xi_3^{(1)}$ and $\xi_3^{(2)}$ describe a triangle with the angle γ equal to $80\,\mathrm{deg}$. At this point the state ξ of the triangle is defined by four noisy measurement: $\left\{\xi_1, \xi_2, \xi_3^{(1)}, \xi_3^{(2)}\right\}$. Finally, the three angles of the triangle are found by minimizing an error criterium D between the "optimal" state $\widehat{\xi}$

and the set of measurements, i.e.

$$\widehat{\xi} = \arg\min_{\xi} \left\{ D(\xi,\xi_1) + D(\xi,\xi_2) + \frac{1}{2}D(\xi,\xi_3^{(1)}) + \frac{1}{2}D(\xi,\xi_3^{(2)}) \right\}.$$

Note that the last measurement θ_3, which is represented by two samples $(\xi_3^{(1)}, \xi_3^{(2)})$ has been weighted by a factor $1/2$ in order to influence the final result in the same way than θ_1 and θ_2.

The alignment of the panoramic image is very similar to the triangle case, except that there is one of the parameter that is common to every measurement, namely the focal length. The remaining parameters are the angles of rotation of the camera that can be measured with each image pair. The constraint of $360\,\mathrm{deg}$ is embedded by suppressing the last rotation from the description, as was done in the triangle example. Similarly, the last rotation is expressed as a combination of the former ones, in the same way it is done in the triangle example.

3.1.5 Linear optimisation on a panoramic constraint

For full panoramic images, the combination of all rotations along the panorama reduces to identity:

$$\mathbf{R}^{(1)}(\xi) \cdot \ldots \cdot \mathbf{R}^{(m-1)}(\xi) \cdot \mathbf{R}^{(m)}(\xi) = \mathbf{I}, \tag{3.6}$$

where $\mathbf{R}^{(i)}(\cdot)$ is the rotation matrix representing the motion between image i and image $i+1$. Note that equation (3.6) holds also for an arbitrary motion of the camera. As previously, the individual estimations θ_i can be written in the *constrained parameter space*, that is

$$\xi_i = [\theta_i^c, \mathbf{0}, .., \theta_i^a, \mathbf{0}, ...]^T,$$

where θ^c represents the focal length of the camera, and θ_i^a the rotation angles of each camera motion. ξ only contains the parameters of the $m-1$ first rotations. In order to meet the constraint, the rotation between the last and the first image ($\mathbf{R}^{(m)}(\xi_m)$ or $\mathbf{R}(\theta_m)$), has to be expressed as a combination of all the other rotations:

$$\xi_m = [\theta_m^c, \upsilon_1^a, \upsilon_2^a, ..., \upsilon_{m-1}^a]^T, \tag{3.7}$$

where υ_i^a are rotation parameters that are computed in Section 3.1.5 and that verify

$$\mathbf{R}(\theta_m) = \mathbf{R}^{T(m-1)}(\xi_m) \cdot \ldots \cdot \mathbf{R}^{T(1)}(\xi_m). \tag{3.8}$$

In other words, the rotation $\mathbf{R}^{(m)}(\xi_m)$ should be equal to the one found in the unconstrained estimation. There are several ways to find ξ_m that are discussed in Section 3.1.5 that computes ξ_m in the same way as in the example of Section 3.1.4 and finds $m-1$ different solutions for ξ_m: $\left\{\xi_m^{(1)}, ..., \xi_m^{(m-1)}\right\}$. Now the goal is to find ξ that minimises the distance sum of equation (3.3), corrected to handle the $m-1$ solutions for ξ_m :

$$D(\xi) = \sum_{i=1}^{m-1} D(\xi_i, \xi) + \frac{1}{m-1}\sum_{k=1}^{m-1} D(\xi_m^{(k)}, \xi) \tag{3.9}$$

where

$$D(\xi_i, \xi) \simeq \frac{1}{n_i}\sum_{j=1}^{n_i} \| \frac{\partial \mathbf{U}_i}{\partial \xi}\Big|_{(\mathbf{p}_j, \xi_i)} (\xi_i - \xi)\|^2.$$

Each of the solution of ξ_m has been weighted by a factor $\frac{1}{m-1}$ in order to give ξ_m the same importance in the computation of the error criterion. The solution is found like in (3.5) by solving the linear system

$$(\sum_{j=1}^{m} \mathbf{E}_j^T \mathbf{E}_j)\xi = \sum_{j=1}^{m} \mathbf{E}_j^T \mathbf{E}_j \xi_j, \tag{3.10}$$

$$\mathbf{E}_i \triangleq \frac{\partial \mathbf{U}_i}{\partial \xi}\Big|_{\xi=\xi_i}. \tag{3.11}$$

Expressing the last rotation

The computation of ξ_m in equation (3.7) is not unique. Indeed, we are trying to express a rotation (θ_m) as a combination of $m-1$ rotations (remember that ξ_m and θ_m represent the same rotation expressed in a different way). We will proceed exactly as in the small example of Section 3.1.4, where the goal was to express one angle as a combination of two angles. To compute ξ_m, we propose to keep the estimate of $m-2$ rotations unchanged, and recompute one of the rotations (υ_k^a) such that the combination of the $m-1$ rotations is equal to θ_m. In other words

$$\xi_m^{(k)} = [f, \theta_1^a, ..., \theta_{k-1}^a, \upsilon_k^a, \theta_{k+1}^a, .., \theta_{m-1}^a],$$

where $\xi_m^{(k)}$ verifies (3.8). There are $m-1$ different ways of doing the computation (depending on where υ_k^a is placed in the vector $\xi_m^{(k)}$) leading to $m-1$ solutions for ξ_m: $\left\{\xi_m^{(1)}, ..., \xi_m^{(m-1)}\right\}$.

The computation of the distance (3.9) requires the computation of $\frac{\partial \mathbf{U}_m}{\partial \xi}\Big|_{\xi=\xi_m}$. It can be computed using

$$\frac{\partial \mathbf{U}_m}{\partial \xi} = \frac{\partial \mathbf{U}_m}{\partial \theta_m}\frac{\partial \theta_m}{\partial \xi}.$$

To compute $\frac{\partial \theta_m}{\partial \xi}$, we use the inverse function theorem [Marsden, 1974, chap. 7]: let $\mathbf{M}(\cdot)$ be the function that transforms the rotation angles into a rotation matrix, which for convenience is represented by a vector of 9 elements. Let $\mathbf{R}_m = \mathbf{M}(\theta_m)$ be the rotation matrix associated to the rotation angles of θ_m (also organised as a vector of 9 components). We can write

$$\mathbf{R}_m = \mathbf{M}\left(\mathbf{M}^{-1}(\mathbf{R}_m)\right).$$

Now, by taking the derivative with respect to $\mathbf{R}_m$ at both sides gives

$$\mathbf{I} = \frac{\partial \mathbf{M}(\mathbf{u})}{\partial \mathbf{u}}\Big|_{\mathbf{u}=\mathbf{M}^{-1}(\mathbf{R}_m)} \frac{\partial \mathbf{M}^{-1}(\mathbf{R}_m)}{\partial \mathbf{R}_m},$$

where $\mathbf{I}$ is the identity matrix. Since $\theta_m = \mathbf{M}^{-1}(\mathbf{R}_m)$,

$$\frac{\partial \theta_m}{\partial \mathbf{R}_m} = \left[\frac{\partial \mathbf{R}_m}{\partial \theta_m}\right]^{-1}.$$

where $[\cdot]^{-1}$ denotes the generalised inversion. Finally $\frac{\partial \mathbf{U}_m}{\partial \xi}$ is computed using

$$\begin{aligned}\frac{\partial \mathbf{U}_m}{\partial \xi} &= \frac{\partial \mathbf{U}_m}{\partial \theta_m}\frac{\partial \theta_m}{\partial \mathbf{R}_m}\frac{\partial \mathbf{R}_m}{\partial \xi} \\ &= \frac{\partial \mathbf{U}_m}{\partial \theta_m}\left[\frac{\partial \mathbf{R}_m}{\partial \theta_m}\right]^{-1}\frac{\partial \mathbf{R}_m}{\partial \xi}.\end{aligned}$$

3.1.6 Non-linear optimisation

So far we have based our global optimisation task on the first order approximation expressed in equation 3.3. The purpose was to have a good parameter set that satisfies a constraint. Now the next step is to optimise globally the picture alignment. At this point, we have a set of point correspondences $(\mathbf{p}, \mathbf{p}'_\theta)$ resulting from the first individual and non-coherent motion estimation. From the new coherent parameter set ξ, one can recompute a new set of correspondences $(\mathbf{p}, \mathbf{p}'_\xi)$.

An unconstrained optimisation produces a good alignment of the image overlaps. Therefore, we want to minimize the distance between the point correspondences of the unconstrained method and the point correspondences of the coherent parameter set, by solving

$$\min \sum_{j}^{m} \rho(\mathbf{p}'_{\theta_j} - \mathbf{p}'_\xi), \tag{3.12}$$

where in general $\rho\left(\cdot\right)$ is the squared norm or any other function used in robust estimation [Ayer, 1995; Fua, 1999], defined in Chapter 4. The optimisation can be performed using a standard descent algorithm, like the Gauss-Newton algorithm [Seber and Wild, 1989], presented in Section 2.6.2.

3.2 Panoramic images

For constructing a panoramic view of several images, the camera motion can be estimated on each pair of overlapping images. We estimated the motion using the method described in Section 2.6.2. We can see in Figure 3.3(a) that an accumulation of little registration errors caused the last picture to be out of alignment with the first one. The images have been projected onto a cylinder for rendering . The gap represents the mis-alignment with respect to the 360 degrees constraint. By solving (3.10), we obtained the image in Figure 3.3(b). We can see that the error has been spread out over each rotation, and the focal length has been adjusted to keep the overlaps between the images. The next optimisation step using criterion (3.12) did not produce any significant change to the result. This suggests that the approximation used in equation (3.3) is accurate enough for this particular example.

It is worth pointing out that the constraint expressed in equation (3.6) is not exactly a 360 degrees constraint but rather a 0 modulo 360 degrees constraint. The side effect is that the initial estimate should be close enough to 360 in order to produce a good result. It may happen that the solution of equation (3.10) will lead to a 720 degrees panorama or to an identity panorama, which superimposes every image onto each other. The singularity of the method is easily detectable, but is not easy to correct in the case of an identity result.

3.3 The multi-planar system

Until now, we assumed that the camera undergoes only a rotation, so that the mosaic can be reconstructed using a single rotational model. In practice, however, the photographer often has to move between captures, and this results in the presence of parallax, i.e. the objects of the scene close to the camera, for example the floor, move more in the field of vision than the objects far away. Our goal is to be able to reconstruct a mosaic in presence of parallax. The method chosen uses a description of the scene by several planes in space, as was also done by Torr [Torr et al., 1999] and Baker [Baker et al., 1998]. The planes provide a simple way to handle parallax and can be extended to the parts of the image where no 3D information is available. Moreover, in everyday scenes, a lot of objects are nearly planar (walls, floors, etc.). The choice of not expressing a 3D meshed model, like Koch [Koch et al., 1998], is motivated by the fact that we want to be able to reconstruct the mosaic with few images only. But, in contrast to Shum [Shum and Szeliski, 1997] who handles parallax in a 2D manner, we introduce 3D parameters to attempt to achieve better precision, and thus better quality. The novelty of our approach lies in handling parallax without a complete overlap of the scene on several images. The segmentation of the scene into several planes is performed by means of a *mixture model*.

3.3.1 Mixture of models formulation of an image model

The mixture of models is a statistical framework that allows to handle data that has several distinct behaviours. As presented by Lindsay [Lindsay, 1995], the simplest mixture model arises when one samples from a population composed of several sub-populations that are called the *components* of the mixture. Each component is characterised by a probability model

$$\Pr[X = x \mid J = j] = P(x; \xi_j) \tag{3.13}$$

where j indicates from which model the random variable X is sampled. P represents a known density function called the *component* density and ξ_j are the unknown component parameters. The proportion of the total population that is in the j^{th} component is called the *component weight* and is denoted by ϕ_j. The variables (X_i, J_i) are random samples of the joint density function

$$\begin{aligned}\Pr[X = x, J = j] &= \Pr[X = x \mid J = j] \Pr[J = j] \\ &= P(x; \xi_j) \cdot \phi_j.\end{aligned} \tag{3.14}$$

The mixture model arises when the component label j is missing, that is when we do not know to which model a random variable belongs. Since ϕ_i are the proportion of each model, we have $\sum_i \phi_i = 1$ and $\phi_i \geqslant 0$.

In our framework, the models are the different planes of the scene, ξ_j are the unknown motion parameters and the plane positions. The component label j describes a segmentation of the image into the different motions. This formulation has been used in the past [Torr et al., 1999; Wang and Adelson, 1993; Jepson and Black, 1993; Weiss and Adelson, 1996; Ju et al., 1996; Sawhney and Ayer, 1996; McLachlan and Basford, 1988; Black and Anandan, 1996] to segment the image according to several motion models.

3.3.2 The multi-planar motion model

In this section, the planar motion model of Section 2.4.3 is adapted to handle several planes in the scene. The parameter set is composed of the seven parameters of the camera, as in Section 2.4.3, to which 3 parameters per plane are added, resulting in a motion parameter vector θ with $7 + 3M$ components, where M is the number of planes in the scene. In this formulation, we will segment the pixels into the different planes, regardless of the fact that they might represent an outlier (the outlier are treated in a sub-sequent step). We formulate the joint problem of a multi-planar motion estimation and image segmentation as the optimisation of a single objective function: Let $\hat{I}_j$ be the prediction for image I_0, obtained by warping I_1 using parameter set j. The conditional probability density function for I_0 can be expressed as [McLachlan and Basford, 1988]

$$f(I_0(\mathbf{p}) \mid I_1(\mathbf{p}), \mathbf{\Phi}) = \sum_{j=1}^{M} \phi_j \left\{(1-\phi_{\mathcal{O}j}) \cdot p\left[e_j(\mathbf{p},\theta) \mid \sigma_j\right] + \phi_{\mathcal{O}j} \cdot p_{\mathcal{O}}[e_j(\mathbf{p},\theta)]\right\} \tag{3.15}$$

$$e_j(\mathbf{p},\theta) \triangleq I_0(\mathbf{p}) - \hat{I}_j(\mathbf{p},\theta)$$

M is the number of planes of the scene, ϕ_j is the component weight associated to plane j (i.e. the proportion of the image covered by plane j), $p_{\mathcal{O}}$ is the outlier probability distribution, and $\phi_{\mathcal{O}j}$ is the outlier proportion in plane j. $\mathbf{\Phi}$ represents the complete parameter set (the motion parameters θ, the model variances σ_j and the model weights ϕ_j). Equation (3.15) is a mixture of models where each component is a mixture of inlier and outliers. We assumed that every model belongs to the same parametric family, in particular:

$$p\left[I_0(\mathbf{p}) \mid \hat{I}_j(\mathbf{p},\theta), \sigma_j\right] \sim \mathcal{N}(\hat{I}_j(\mathbf{p},\theta), \sigma_j)$$

where $\mathcal{N}(\hat{I}_j(\mathbf{p},\theta), \sigma_j)$ are the set of Gaussian random variables with spatially varying mean $\hat{I}_j$. This is equivalent to the following formulation, which uses the pixel-to-pixel error

$$p\left[I_0(\mathbf{p}) - \hat{I}_j(\mathbf{p},\theta) \mid \sigma_j\right] \sim \mathcal{N}(0, \sigma_j).$$

The outlier distribution $p_{\mathcal{O}}$ is computed as outlined in Chapter 4. We are looking for the *maximum likelihood* estimate of $\mathbf{\Phi}$, the complete parameter set. For each pixel location $\mathbf{p}_i$ the posterior probability τ_{ji} of the population membership can be computed as

$$\begin{aligned}\tau_{ji} = \Pr[\mathbf{p}_i \in \hat{I}_j \mid I_0, \Phi] &= \frac{\Pr[I_0 \mid \mathbf{p}_i \in \hat{I}_j, \Phi] \Pr[\mathbf{p}_i \in \hat{I}_j \mid \mathbf{\Phi}]}{\Pr[I_0 \mid \mathbf{\Phi}]} \\ &= \frac{\left\{(1-\phi_{\mathcal{O}j}) \cdot p\left[e_j(\mathbf{p},\theta) \mid \sigma_j\right] + \phi_{\mathcal{O}j} \cdot p_{\mathcal{O}}[e_j(\mathbf{p},\theta)]\right\} \cdot \phi_j}{\sum_{k=1}^{M} \phi_k \left\{(1-\phi_{\mathcal{O}k}) \cdot p\left[e_k(\mathbf{p},\theta) \mid \sigma_k\right] + \phi_{\mathcal{O}k} \cdot p_{\mathcal{O}}[e_k(\mathbf{p},\theta)]\right\}}\end{aligned} \tag{3.16}$$

In practice, for every model a warping is performed, and an error value is computed for each model at a particular location $\mathbf{p}$. Equation (3.16) is the probability of the error times the proportion of the image covered by the plane, normalised over all the models. To get a segmentation of the image into the different layers, we introduce a binary indicator variable z_{ji}:

$$z_{ji} = \begin{cases} 1, & \mathbf{p}_i \in \hat{I}_j \\ 0, & else \end{cases} \tag{3.17}$$

z_{ij} takes value 1 if $\tau_{ji} > \tau_{ki} \quad \forall k$ and 0 otherwise. In Section 3.4.2 z_{ij} will be modified to take into account the dependence between neighbouring pixels. The model proportions ϕ_j are computed using

$$\hat{\phi}_j = \sum_{i}^{N} \frac{z_{ji}}{N}. \tag{3.18}$$

3.4 Estimation algorithm

The estimation can be performed in a recursive way using the Expectation-Maximisation algorithm [Moon, 1996; Dempster et al., 1977] to solve the problem.

The algorithm needs an initial estimate of all the parameters, which is given by the initialisation technique described in Section 3.4.3. The algorithm starts by computing the error value at each pixel location, and then a sequence of expectation and maximisation steps is performed.

Expectation:

- The single memberships τ_{ji} are computed using (3.16).
- The memberships are specialised to binary values, using either (3.17) or the morphological version in (3.21).
- The component weights are computed by applying equation (3.18).
- For each model, the outlier proportion $\phi_{\mathcal{O}j}$ is computed as described in Chapter 4. The error histogram on which the estimation of $\phi_{\mathcal{O}j}$ is based is constructed using the error values in the region belonging to plane j.

Maximisation:

- In the Maximisation step, a robust motion estimation is performed, as described in Section 2.6.2 and Chapter 4, by considering for each model only the pixels that have been assigned to it.

After each E-M iteration, there is an attempt to merge or suppress some planes based on a criterion explained in the following section.

3.4.1 Estimating the number of planes

By trying to estimate the number of planes, there is a trade off between complexity and goodness of fit: the more planes are used, the smaller the registration error, but the more complex the model. Moreover, an over-parametrised model might fit the noise and would not explain the scene properly. Roberts [Roberts et al., 1998] addresses the question in a more general framework, he looked for the maximum likelihood [2] value for M, the number of models. By making some assumptions on the priors (the a-priori value of the parameters), he derives an expression that accounts for the goodness of fit and complexity at the same time. The method studies the uncertainty as a whole: The uncertainty of the parameters (e.g. the Cramer-Rao lower bound) increases with the number of parameter estimated, whereas the uncertainty of the data (the inverse of the goodness of fit) decreases with the parameter number. Minimizing the whole uncertainty is a trade off between the two. It is then shown that the Minimum Description Length (MDL) [Rissanen, 1978] criterion gives similar results than the global maximum likelihood estimator in estimating the number of models. These results also match those found in Barron [Barron and Cover, 1991] and Takeuchi [Takeuchi, 1997]. Although Torr [Torr et al., 1999] used the full evidence to determine the number

[2] He refers to it as “full evidence”.

of planes in his multi-planar description, we use the MDL criterion like in Liang [Liang et al., 1992] and Ayer [Sawhney and Ayer, 1996], mainly because of reduced computational complexity.

The Description Length counts the amount of information needed to code a particular state of the estimation model. We compute the encoding length of the mis-registration between $\{\hat{I}_j\}$, the image obtained by warping image I_1 according to the equation of plane j, and I_0. We have to code the model parameters θ, the model scales σ_j, the weights ϕ_j, the error values at each pixel location and the segmentation information. The expression for coding the error values for each pixel is given by [Cover and Thomas, 1991]

$$\begin{aligned} L_e &= -\sum_{i=1}^{N} \log P(e) \\ &= -\sum_{i=1}^{N} \log \left[\sum_{j=1}^{M} z_{ji}((1-\phi_{\mathcal{O}j}) \cdot P_{\mathcal{I}j}(e) + \phi_{\mathcal{O}j} P_{\mathcal{O}}(e)) \right] \end{aligned} \quad (3.19)$$

To code the segmentation information, a first order Markov Model is used, where the segmentation of each pixel is coded according to the state of its left neighbour. In practice, the method counts the number of transition from state k to state l[3] (and vice-versa) by scanning the image line-wise, and builds a transition matrix $T_k l$. The encoding cost is given by

$$L_T = -\sum_{k=1}^{M} \sum_{l=k+1}^{M} T_{kl} \log(T_{kl}) - \left(\sum_{k=1}^{M} T_{kk} \right) \log \left(\sum_{k=1}^{M} T_{kk} \right) \quad (3.20)$$

where $\mathbf{T}$ has been normalised such that $\sum_{k=1}^{M} \sum_{l=k}^{M} T_{kl} = 1$. The diagonal elements of $\mathbf{T}$ have been grouped in equation (3.20) because it is only necessary to know that a pixel has the same label than its neighbour, without specifying which one, to be able to reconstruct the segmentation image. The coding lengths of the models parameters θ as well as the model scales σ_j can be neglected.

On complexity

To evaluate equation (3.19), the algorithm constructs a matrix $\mathbf{E}_{M \times M}$ whose element E_{kl} count the encoding length of the error if model l is replaced by model k. This requires about $M \cdot N$ operations, where M is the number of models, and N the number of pixels ($M \ll N$). The construction of the matrix $\mathbf{T}$ requires about N operations. Then, to test if a model is a candidate for deletion, the Description Length can be computed using only matrix $\mathbf{E}$ and $\mathbf{T}$, and thus representing a negligible amount of computation.

On the order of deletion

In the beginning, the number of models is largely over-estimated. This has the effect of making the final result sensitive to the order in which the models get pruned, similar to motion estimation where the model can be trapped in a local minimum. To prevent this, the algorithm first deletes the models that can be replaced by a similar one. Formally, the similarity

[3] By state k we mean that the pixel belongs to the plane k.

between two models can be defined as the minimum average amount by which the correspondences $\{\mathbf{p}_i; \mathbf{p}'_i\}$ have to be moved in both models to be described by a single one of the same family, which then has correspondences $\{\mathbf{p}_i; \mathbf{p}''_i\}$, as described in Section 3.1.3 or in [Hasler et al., 1999]. The distance is expressed as $\sum_i \|\mathbf{p}'_i - \mathbf{p}''_i\|^2$. In more pragmatic terms, to find the similarity between two models, the two models are merged into a single one and the displacement of the correspondences caused by the merging is taken as a similarity measure.

The algorithm starts by ranking the models by similarity. Then, in the set composed by each model and its closest neighbour, it prunes the one that reduces the Description Length the most, and recomputes the similarity ranking. If no pruning in this set reduces the Description Length, the algorithm considers the set composed of each model and its second closest neighbour, until the minimum of the Description Length is reached.

3.4.2 The segmentation algorithm

This section presents a way to use the dependence between the pixels in an image with a morphological algorithm inspired by the *watershed* algorithm [Beucher and Meyer, 1993; Wang, 1998; Gatica-Perez et al., 1999].

The algorithm starts with a set of raw membership probability images τ_{ji} of equation (3.16) and an initial segmentation image, where some regions are assigned to the different models of the mixture and some other regions are left unspecified. The aim of the algorithm is to make these regions grow until every pixel of the image is assigned to a model. For a given model j, a point in its segmentation mask z_{ji} will be added if

$$\begin{aligned} (z_j \oplus k)\,(\mathbf{p}_i) &= 1, \\ z_j(\mathbf{p}_i) &= 0 \quad \forall j, \\ \tau_j(\mathbf{p}_i) &> \Upsilon, \end{aligned} \tag{3.21}$$

where k is a morphological filter kernel. Here we used a 3×3 matrix made out of ones. $\oplus$ denotes the binary morphological addition operation [Haralick and Shapiro, 1992]. $z_j(\mathbf{p}_i)$ and z_{ij} represent the same variable; the first notation is used to emphasise the fact that the measure takes place at position $\mathbf{p}_i$. Υ is a threshold that is iteratively updated as follow

$$\Upsilon^{(k+1)} = \min(\Upsilon^{(k)}, \alpha \cdot \mu_\tau)$$

where μ_τ is the mean of the $\tau_j(\mathbf{p}_i)$ values that have been added by the algorithm at the preceding iteration computed over all models, and α is a value that accounts for the speed of the flood in the watershed algorithm, or in other words, a parameter that represents the a priori assumption on the smoothness of the segmentation. Typically, α is set to 0.7 ($0 \leq \alpha \leq 1$), although a heuristic can be used on α to speed up the segmentation. In practice, α has some influence on the speed of convergence, but rather little influence on the final result.

To update the segmentation at each E-M iteration, the algorithm starts with an erosion operation to shrink the segmented region followed by the growing operation described in (3.21). This is similar to what was proposed by [Baker et al., 1998] but in reverse order. The amount of erosion allowed at each iteration depends on the minimum size an object in the scene can have.

3.4.3 The initialisation phase

The algorithm described so far is quite sensitive to initial conditions, hence a good initialisation is crucial to its success. Because it is aimed at the reconstruction of a mosaic composed out of a number of images, we want to avoid an initialisation performed by hand. Indeed, if the images were segmented by hand, then nothing would be left to the algorithm, since, in contrast to video applications, we cannot use the result of the segmentation to treat the next image.

The algorithm starts with

- A set correspondences $\{\mathbf{p}_i; \mathbf{p}'_i\}$, either set by hand or computed automatically using some well known techniques [Deriche and Giraudon, 1993].
- The a priori uncertainty D of the position of these correspondences (measured in pixels).
- The resolution at which the algorithm should start.
- A parameter α that accounts for the smoothness of the scene segmentation.
- An a priori minimum size of the object of the scene.

The algorithm begins with a 3D estimation using feature points, then goes down to 2D using a mixture of homographies, to finally come back to 3D with the multi-planar model.

Camera motion and initial planes in space

From the set of features $\{\mathbf{p}_i; \mathbf{p}'_i\}$, the algorithm computes the pose of the camera [Hartley, 1997] (in our case, we assumed a fixed and known focal length). Then, for each feature point $\mathbf{p}_i$ in image I_1, it finds the 2 closest features in the image, and builds a plane in space that crosses the 3 points $\{\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k\}$. The equations of the plane, as well as the camera motion define all the parameters of the planar motion model described in Section 2.4.2. At this stage of the computation there are as many planes (or parameter sets) as there are features in the initial correspondences set. Note that at least 2 planes are already redundant at this stage.

Merging the planes

Next, given a plane, the algorithm tests if a feature $\mathbf{p}_l$ is part of that plane or not. The algorithm runs 2 or 3 steps of the gradient descent method on the plane parameters (i.e. $\mathbf{N}$ in equation 2.10) by including feature $\mathbf{p}_l$ in the feature set of the plane. If that feature can be included in the plane by moving it less than a given distance D - and by also moving the other features of the plane less than D from their original location - the feature is accepted and the plane parameters are updated. Formally, the plane is updated if we can find $\mathbf{N} = [N_1 \; N_2 \; N_3]^T$ such that

$$\max_{\forall i : \mathbf{p}_i \in plane} \; \min_{N} (||\mathbf{p}'_i - \mathbf{U}_\mathbf{N}(\mathbf{p}_i)||) < D,$$

where $\mathbf{U}_\mathbf{N}(\cdot)$ is defined by equation (2.10).

Finally, the algorithm suppresses the redundant planes from the list.

Going to 2D

From the set of planes, a set of homographies are built, by determining the components of matrix $\mathbf{K}$ in equation (2.12). We chose to use this intermediate step with a mixture of homographies because the homography has better convergence properties than the planar model due to the reasons pointed out in Section 2.6.2.

Improving the initial motion estimates

With the features belonging to each homography, the algorithm builds the convex hull of the features in the image, and use the pixels inside this hull to perform a single motion estimation, as described in Section 2.6.2. This last step improves the alignment for each model.

Initial segmentation

Finally, it computes the binary membership weights for each pixel in the image, in itself creating an image z_j for each model, and performs a morphological erosion on each z_j image, to get rid of casual zero-crossings (the amount of erosion used is adjusted according to the minimum size of the object of the scene). At this point, each pixel belongs to one model of the list, unless it has been eroded by the filtering. The result of the erosion is used to trigger the segmentation algorithm of Section 3.4.2.

Back to 3D

Once the mixture of homographies has reached stability, the initialisation of the 3D multi-planar model is computed using artificially generated correspondences $\{\mathbf{p}_i; \mathbf{p}'_i\}$ by using the motion parameters of the homographies, and picking uniformly the feature points $\{\mathbf{p}_i\}$ in the regions assigned to each homography. One plane is built per homography, and the algorithm continues with the same segmentation than before. This last step into the $3D$ world enables to get rid of the models that are inconsistent with the single camera motion constraint, and improves slightly the alignment.

3.5 Experiments with two planes

This example illustrates how a constraint on the camera motion can improve the estimate of a plane location in space. This constraint enforcement is used in the multi-planar system when going from a $2D$ scene description toward a $3D$ description. The algorithm used to merge the planes was presented in Section 3.1.3.

We start with two photographs of two scribbled blackboards taken in a classroom. The blackboards are both parallel to the wall of the room, and can slide one over the other. The top blackboard is the one that is the furthest away from the camera. There is a distance of 3 metres from the lower blackboard to the camera, and the camera moved 2m to the left parallel to the blackboard. The planar motion model of Section 2.3.2 is used to model the warping of the blackboards. By making the correspondences of some points in the image, we calculated the motion parameters using the feature points techniques of Section 2.5.2. The segmentation is supposed to be known. Here we present the most relevant results from the computation, namely a measure of the distance between the 2 blackboards. This measure depends on the camera translation which is indeed one of the ill-posed parameter of the

type of computation	distance	error
Independent planes	19,7 cm	8.1 cm
Coherent motion	8.7 cm	2.9 cm
Joint optimisation	11.2 cm	0.4 cm
Measured	11.6 cm	-

Table 3.1: Improvement of the position estimate of two planes in space. The table compares the estimate of the distance between two planes computed with different methods to the real one

problem [Daniilidis and Nagel, 1993]. Another reason for choosing this parameter is that it can easily be measured with a precision of 1 mm. Three computations have been done: The first using two independent planes (two independent use of the planar motion model); The second by calculating one single camera motion using (3.5); and the last performing a joint optimisation for one motion and two planes by solving equation (3.12) and using the result of the second computation as a starting parameter set for an iterative algorithm.

The results are summarised in Table 3.1 and show that there is a step-wise improvement in precision. The distance computed using the 2 independent estimation (that gave 2 different camera motion) is measured along the optical axis from the first camera position. By imposing a single camera motion, using first the linear approximation of equation (3.3), and the non-linear optimisation, lead to an improved result. The distance between parameter set 1 and 2 (or ξ_1 and ξ_2) gave 4.6 “pixels” using (3.3) and 1.3 “pixels” using the more precise (3.2), e.g. the correspondences p'_θ have to be moved by 1.3 pixels in average in order to get a coherent parameter set. This makes the hypothesis of a single camera motion quite reasonable from the data point of view. Note that in order to get to the joint optimisation, it is necessary to compute the coherent motion first.

3.6 The Multi-planar reconstructions

3.6.1 Image of ’Le Louvre’ in Paris

In this section, we applied the multi-planar motion model to reconstruct a mosaic from an image pair. The images have been taken by walking around an obstacle, and therefore some parallax is present. Starting with 43 feature points, the system computed 13 planes to start with, ended up with 4 homographies, and then 3 planes in space. During the motion estimation process we considered that the focal lengths were known in order to keep the conditioning of the system at an acceptable value. The result is shown in Figure 3.5. Once the 3 warpings with the parameters of the 3 models have been done, the choice of the segmentation was left to the user. To illustrate the point, we first did an automatic segmentation, which takes the plane that is the closest to the camera. This generates the tilted biker and the distorted pedestrian. The pedestrian on the right has been corrected by hand, by choosing it to be part of the wall (instead of half in the wall and half on the floor). The apparent simplicity of the mosaic is somehow misleading: first, an illumination change occurred between the captures (due to a cloud). Then, the picture was taken with a 24 mm lens which has some severe vignetting in the borders. Finally, the homography (or the plane if we are in 3D) describing the floor does not follow the approximations of Section 2.6.2 as well as the wall or the tree in the Flower Garden sequence. Figure 3.7 shows the segmentation obtained using the mixture of homographies, and the mixture of planes in space. The images have been colour corrected according the algorithm outlined in Chapter 5 and the different results of the segmentation are shown for the original and corrected image pair.

3.6.2 Flower Garden Image

In order to compare our approach to previous ones [Torr et al., 1999; Baker et al., 1998], we applied our algorithm to 2 images of the Flower Garden sequence. The comparison is somewhat biased because we only use two images of the sequence and these images do not need any of the colour correction technique. Figure 3.8 shows the segmentation obtained, and the outlier mask. The model started with 55 features resulting in 6 planes, and ended up with 3 planes.

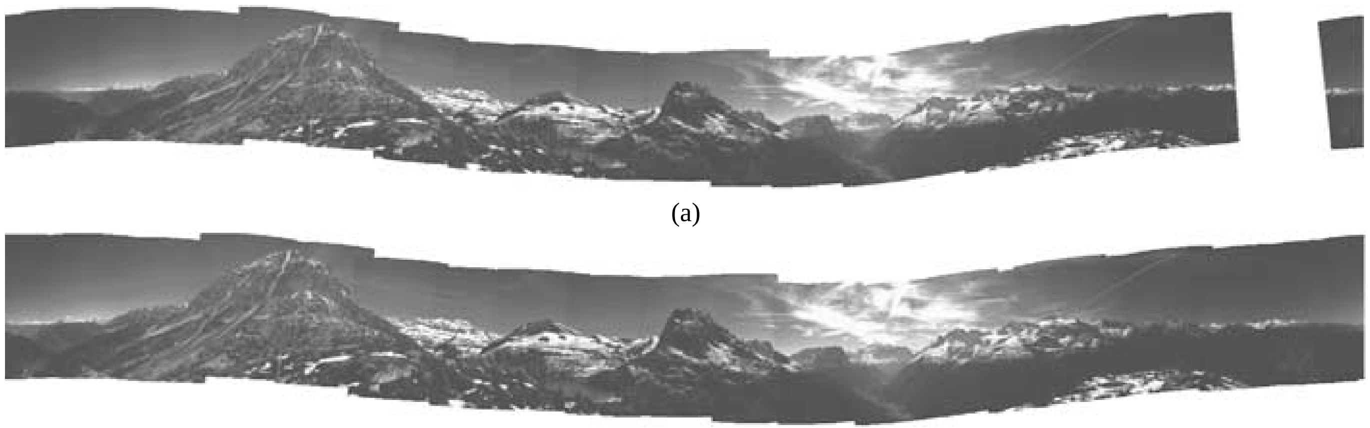
3.6.3 Images of Lausanne

A third mosaic is presented in Figure 3.9. In the background, the roofs present a lot of occlusions and de-occlusions and where just treated as outliers by the system.

3.7 Conclusions

This chapter presented a novel technique to go from an unconstrained toward a constrained estimation system. It allows to generalise the techniques that use image-to-image estimation toward a system that performs a global optimisation over several pictures. The method is successfully applied to close the gap of a 360 degrees panorama. An other example shows how to merge two estimates of a camera motion that are obtained by registering independently the images of two planes. By imposing that the two planes are taken with the camera at the same location, the example shows an improvement in the location of the planes in space. This last example is used in the second topic of the chapter where the scene was decomposed into several planes in space.

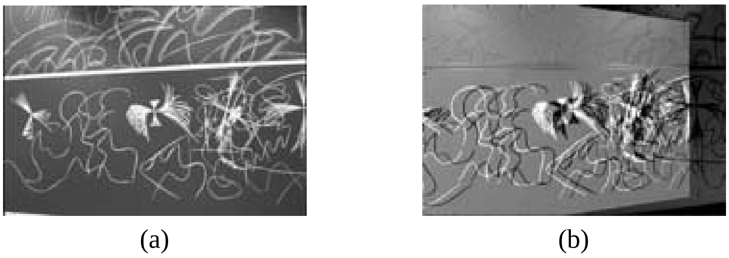
The second part of the chapter presents an algorithm to segment an image pair into several regions that correspond to planes in the scene. The algorithm is able to segment the image and to determine how many planes are in the scene. The final goal is to reconstruct scenes that contain parallax, and where the image do not necessarily overlap everywhere in the mosaic. Although the method showed to work on a few examples, with little initial information, the algorithm is computationally expensive and lacks robustness. The algorithm is based on a strategy using mixtures of models that suffers from some fundamental limitations that are outlined in Chapter 4. Intuitively, when there are many models in the scene, for a particular pixel, it is probable that at least one of the models generates a small registration error, even if this model has nothing to do with the pixel considered.



(a)

(b)

Figure 3.3: Full panoramic image. (a) The estimation has been performed using only image pairs. The little mis-registration on each image results in a large error when comparing both ends of the view. (b) The result has been found by solving a linear equation to meet the 360 degrees constraint. The gap in the panorama has disappeared. Note that the panorama goes up and down a little bit. This is not due to any estimation error, but rather to the way the images are projected onto the cylinder used to render the image.



(a) (b)

Figure 3.4: Image of the two blackboards (separated by the white line): The warped image has been subtracted from the original one. Perfect registration is represented by grey colour. (a) The original image that gets warped in image (b). (b) Warping using the parameters for the upper blackboard. The upper blackboard is almost cancelled (e.g. uniformly grey).

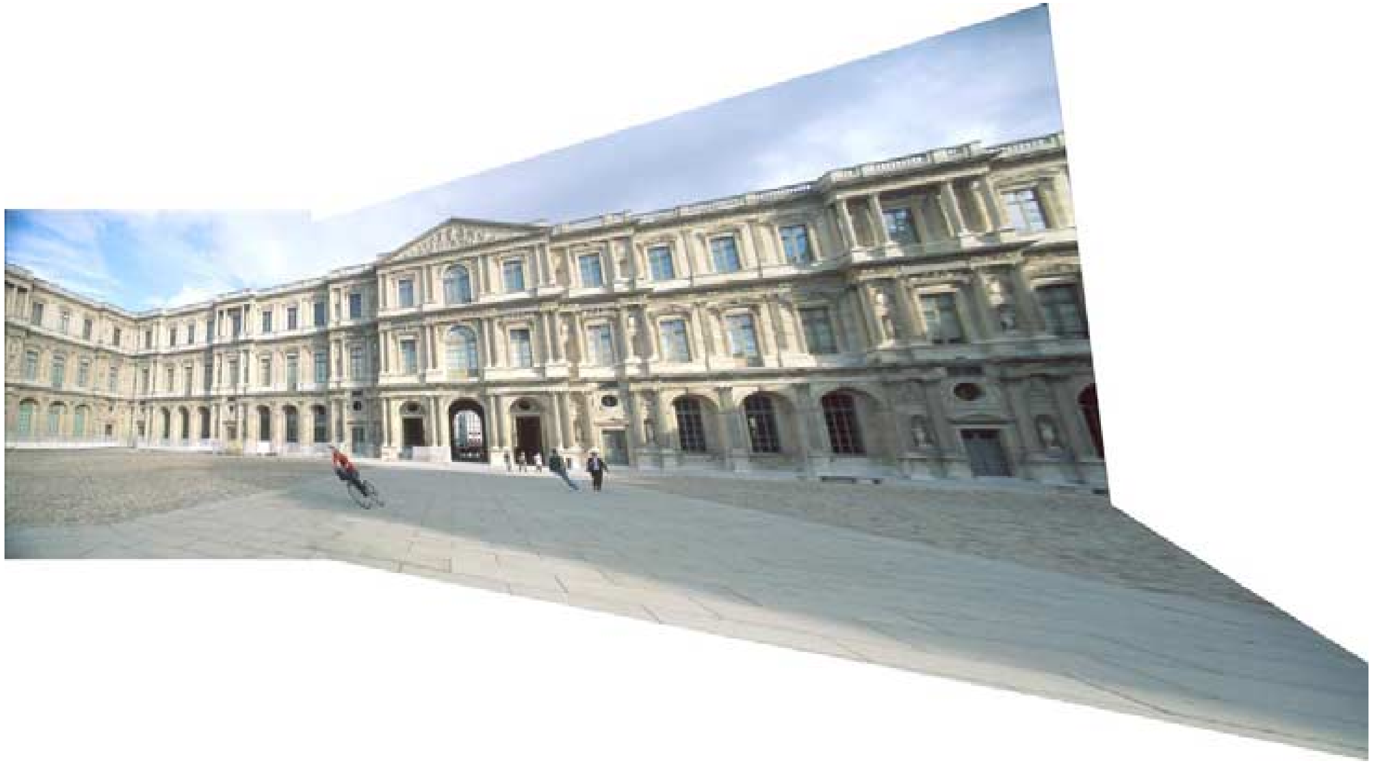


Figure 3.5: Mosaic formed of two pictures. The mosaic has been warped according to two planes, identified by the multi-planar model. The algorithm kept the texture of the plane closest to the camera. This allows to warp the part of the image that does not contain any stereo information. The apparent distortion of the figure is due to the large viewing angle. The original images were taken with a 24mm lens.

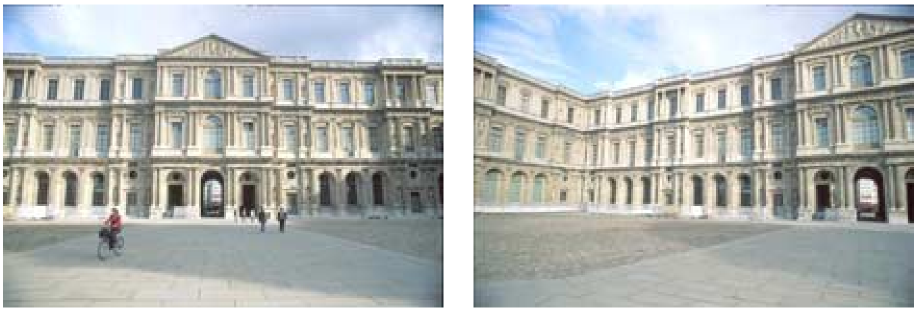
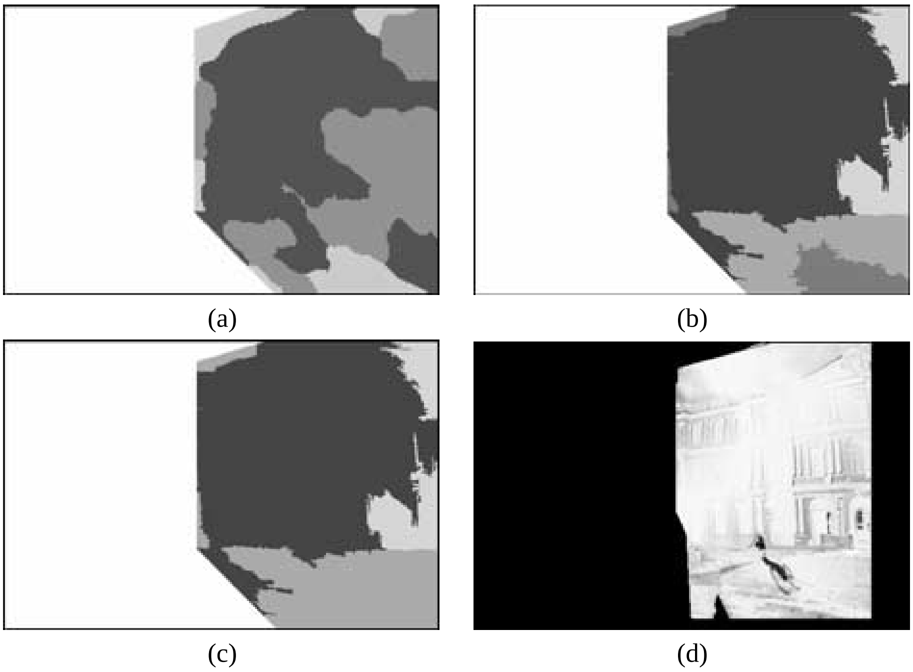


Figure 3.6: Images used to produce the mosaic of Figure 3.5.



(a)

(b)

(c)

(d)

Figure 3.7: Segmentation of the mosaic of *Le Louvre*. (a) using the algorithm without colour correction (b) Segmentation using the mixture of homographies (c) Final segmentation (d) Outlier mask: the probability of being an outlier is represented by the gray value. By using $3D$ estimation between images (b) and (c), one model which was not coherent with the camera motion got pruned. In (a) we can see that the segmentation without doing any colour correction is far from representing the scene in a satisfactory manner.

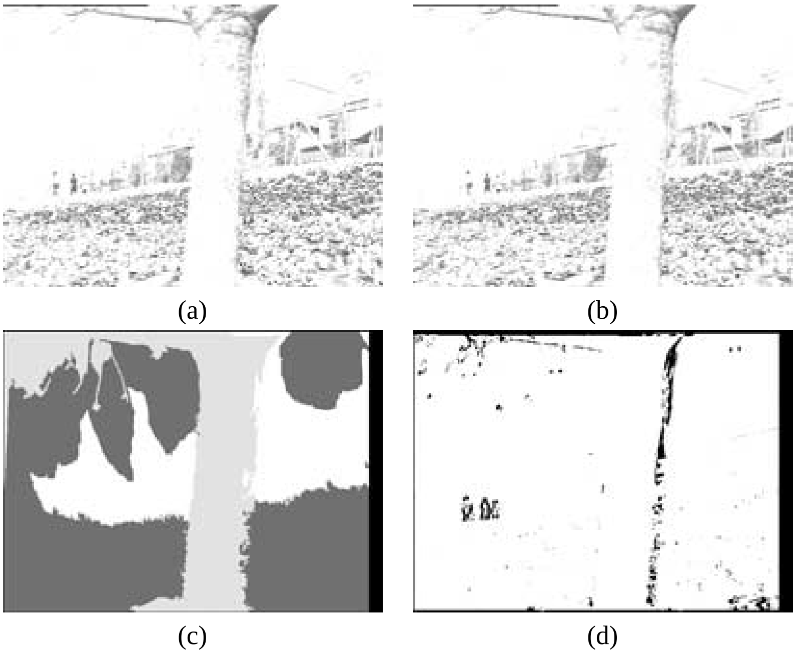


Figure 3.8: The segmentation algorithm applied to the Flower Garden sequence. (a)(b) Input images (c) segmentation image (d) outlier mask. The method found 3 planes in space: the floor, the tree and the background. The sky having no texture has been segmented randomly in one of the planes. The outlier mask is made of greyscale values where the greyscale defines the probability of a pixel to belong to the outliers.

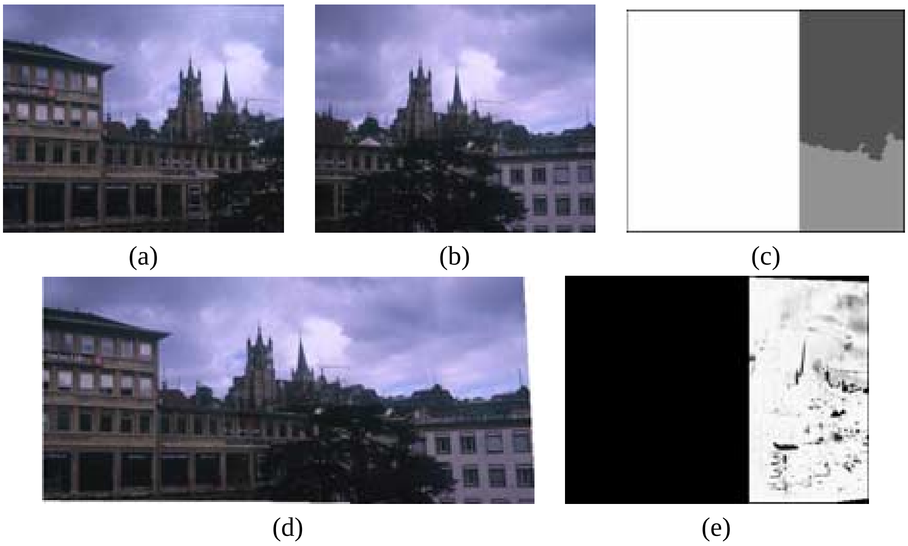


Figure 3.9: Example of mosaic with parallax. The church position with respect to the foreground is quite different in the two input images. (a) and (b) Input images. (c) Segmentation. (d) Reconstructed mosaic. (e) Outlier mask.

Chapter 4

Modelling Outliers

In the other chapters, various methods are presented to stitch images together, provided that the scene is *static*. This assumes that the camera captures the exact same scene, but from a different viewpoint. In practice, however, the estimation of the motion parameters can be perturbed if some objects appear in an image and are absent in an other. We address the question of how to characterise the outliers that may appear when matching two views of the same scene. The match is performed by comparing the difference of the two views at a pixel level, aiming at a better alignment of the images. When using digital photographs as input, we notice that an outlier is often a region that has been occluded, an object that suddenly appears in one of the images, or a region that undergoes an unexpected motion. By assuming that the error in pixel intensity levels generated by the outlier is similar to an error generated by comparing two randomly picked regions in the scene, we can build a model for the outliers based on the content of the two views. The matching problem is then expressed as a mixture of inliers versus outliers, and defines a function to minimise for improving the matching. Our model has two benefits: First, it delivers a probability for each pixel to belong to the outliers. Second, our tests show that the method is substantially more robust than traditional robust estimators (M-estimators) used in image stitching applications, with only a slightly higher computational complexity.

4.1 Introduction

Outlier rejection and robust motion estimation in computer vision applications is a subject that has been investigated for many years. The current techniques are based on optimisation or statistics and are applied to the imaging domain without taking proper advantage of the knowledge that can be extracted from the images. In order to use the information contained in an image, we first characterise an *outlier* in a (photographic) image pair as a region that has been occluded, an object that suddenly appears in one of the images, or a region that undergoes an unexpected motion. The key idea of this paper is to predict the outlier statistical characteristics as follows: We assume that by comparing two arbitrary parts of each image, we get an error pattern similar to the one generated by an outlier. From that, we can compute the expected error distribution generated by the outliers. Finally, we express the motion estimation problem as a *mixture of inlier versus outliers*, and handle outlier rejection like a standard mixture problem [Lindsay, 1995].

The performance of the model is demonstrated using two different types of experiments: The first experiment shows two aligned pictures, one of them containing an outlier. The proportion of the image covered by the outlier is varied by framing the images of the pair differently. The goal is to see when the model breaks down. The second experiment compares the motion estimator derived from our outlier model—which is an M-type estimator—to a standard robust M-estimator.

The chapter starts with a brief review of the principles of robust motion estimation[1], followed by a section defining the inliers, and finally a section describing the outlier model. Then, the two types of experiments and their results are shown. Finally, the model is extended to fit in a competitive mixture of models framework, like the multi-planar motion in Section 3.3.2. This extension points out some fundamental limitations associated to this family of motion estimation techniques.

4.2 State of the art

The state of the art can be approached from two different points of view: On the one hand, there is a general literature on robust estimation and outlier removal from the standpoint of statistics [Huber, 1981] that will not be further discussed here. On the other hand, there is the literature on robust *motion estimation*, for which [Ayer, 1995, chap. 3] and [Rousseeuw and Leroy, 1987] give an extensive review.

When handling images, outliers are detected by specifying a distribution for the inliers and using a threshold scheme if an observation diverges too much from the inlier data, as in [Sawhney and Ayer, 1996; Hager and Belhumeur, 1998; Park et al., 1994; Brailovsky, 1996]. More sophisticated methods [Netanyahu and Weiss, 1994; Schroeter et al., 1998; Torr et al., 1999] use a uniform distribution for the outliers and approach the estimation problem in the context of mixture modelling. In general, there is very little difference between the general statistical approach and the solutions that are applied to images.

4.2.1 A review of M-estimators

To discount the influence of outliers when matching two views of the same scene, M-estimators are applied along with a recursively reweighted least squares approach (See chapter 3 for the details). In this regression, the M-estimator weights the influence of each pixel by a different amount. Formally, this is justified by assuming that the error distribution has a different shape than the usual normal distribution, like for example a Lorentzian or a Geman-McClur shape [Ayer, 1995, chap. 3].

Let $\mathrm{P}_\sigma(\cdot)$ be the (discrete) assumed error distribution, which has one parameter: θ. The likelihood of parameter θ is defined by

$$L(\theta \mid \mathrm{P}_\sigma) = \prod_r \mathrm{P}_\sigma(r)^{n(r)} \tag{4.1}$$

where $n(r)$ is the number of occurrences of the error r. The maximum likelihood value of θ is found by solving

$$\hat{\theta} = \arg\max_\theta L(\theta \mid \mathrm{P}_\sigma).$$

[1] By *motion estimation*, we refer to the motion of the camera between two pictures.

Since the logarithm is a monotonic and increasing function, the maximum likelihood of parameter θ can be found by maximizing the log-likelihood

$$\hat{\theta} = \arg\max_{\theta} \sum_{r} n(r) \log\left[\mathrm{P}_{\sigma}(r)\right],$$

that allows to use a sum instead of a product. This is also equivalent to minimizing the negative log-likelihood

$$\hat{\theta} = \arg\min_{\theta} \sum_{r} n(r) \left(-\log\left[\mathrm{P}_{\sigma}(r)\right]\right).$$

We call the negative log-likelihood the *objective function* ρ_{σ}

$$\rho_{\sigma}(r) = -\log\left[\mathrm{P}_{\sigma}(r)\right] \triangleq \rho\left(\frac{r}{\sigma}\right). \tag{4.2}$$

To return to the motion estimation problem, finding the *Maximum Likelihood* estimate for the given error distribution is equivalent to finding the minimum of the objective function

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{M} \rho(\frac{r_i}{\sigma}), \tag{4.3}$$

where r_i is the error of a pixel-wise comparison of the images, θ is the unknown parameter of the system (often a vector containing the motion parameters of the camera), M is the number of pixels being compared, and σ is the standard deviation of the error.

The most frequently used approach consists in assuming that the error distribution is Gaussian, which leads to the usual (non-robust) minimum mean squared error estimate. In addition to the Gaussian distribution, we will use two distributions: the Lorentzian and the Geman-McClur distribution. The objective functions associated to these distributions (their negative log) is given, for the Gaussian[2] (ρ_N), Lorentzian (ρ_L) and the Geman-McClur (ρ_G) distributions by

$$\rho_N(\frac{r}{\sigma}) = \frac{r^2}{2\sigma^2} \tag{4.4}$$

$$\rho_L(\frac{r}{\sigma}) = \log(1 + \frac{r^2}{2\sigma^2}) \tag{4.5}$$

$$\rho_G(\frac{r}{\sigma}) = \frac{r^2/\sigma^2}{1 + r^2/\sigma^2}. \tag{4.6}$$

The weight W used by the M-estimators in the motion estimation iteration [Hasler, 2001, app. A] is given by $W(r) = \frac{\dot{\rho}(\frac{r}{\sigma})}{r}$, that is

$$W_N(r) = \text{const}, \tag{4.7}$$

$$W_L(r) = \frac{2}{2\sigma^2 + r^2}, \tag{4.8}$$

$$W_G(r) = \frac{2\sigma^2}{\left(\sigma^2 + r^2\right)^2}, \tag{4.9}$$

[2] The N subscript stands for *Normal* distribution.

for the Gaussian (W_N), Lorentzian (W_L), and Geman-McClur (W_G) distributions, respectively. The robustness of various estimators is illustrated by an example in Figure 4.9, where all estimators used in this paper are compared. For an extensive discussion on standard robust estimation, we refer the reader to [Ayer, 1995, chap. 3] and [Rousseeuw and Leroy, 1987].

Note that in order to use the Lorentzian or Geman-McClur estimator, we need to know the error scale σ, which is in general computed using the median of the error.

4.3 The outlier model

Considering a photographic image pair, we believe that *outliers* generate an error pattern similar to the error generated by comparing two random regions of the scene. Thus, we define the outliers as being the result of a random superposition of the images of a pair. The idea is then to characterise the outliers by computing their error distribution and handle the motion estimation problem as a mixture of inliers and outliers in a statistical framework.

Let I be the part of the scene that appears as I_0 in image 0 and as I_1 in image 1. We then compute the statistics of a random superposition of I_0 and I_1. Let us suppose that images 0 and 1 are the result of warping the scene I with two independent random variables $\mathbf{\Theta}_0$ and $\mathbf{\Theta}_1$ that are used as motion parameters: $I_0 = I(\mathbf{\Theta}_0)$ and $I_1 = I(\mathbf{\Theta}_1)$. We are interested in the error distribution $\mathrm{P}(r) = \Pr\{I_0(\mathbf{p}) - I_1(\mathbf{p}') = r\}$, where $\mathbf{p}$ and $\mathbf{p}'$ denote the matching positions in the images. Thus,

$$\mathrm{P}(r) = \sum_{\forall u} \Pr\{I_0(\mathbf{p}) = u, I_1(\mathbf{p}') = u - r\}, \tag{4.10}$$

where u denotes all possible values contained in the images, and r is the error value. Since $\mathbf{\Theta}_0$ and $\mathbf{\Theta}_1$ are independent, so are $I_0(\mathbf{p})$ and $I_1(\mathbf{p}')$, thus

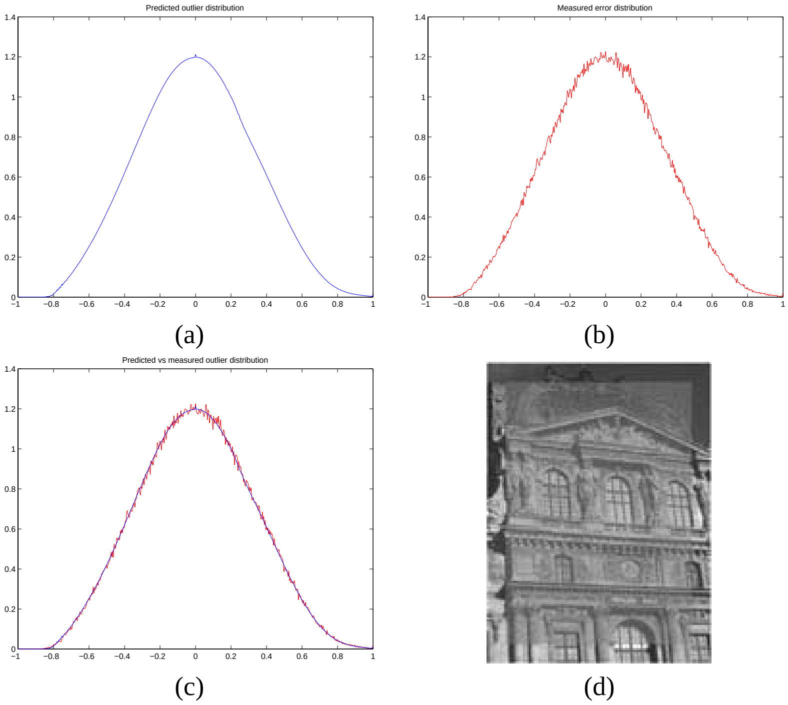
$$\mathrm{P}(r) = \sum_{\forall u} \Pr\{I_0(\mathbf{p}) = u\} \Pr\{I_1(\mathbf{p}') = u - r\}. \tag{4.11}$$

By making some assumptions about the image and the comparison process (detailed in Appendix B), the intensity probability distribution of a single pixel in the image is equal to the image histogram $\mathbf{H}$, normalised such that $\sum_u H(u) = 1$. Equation (4.11) becomes

$$\mathrm{P}(r) = \sum_{\forall u} H_0(u) H_1(u - r). \tag{4.12}$$

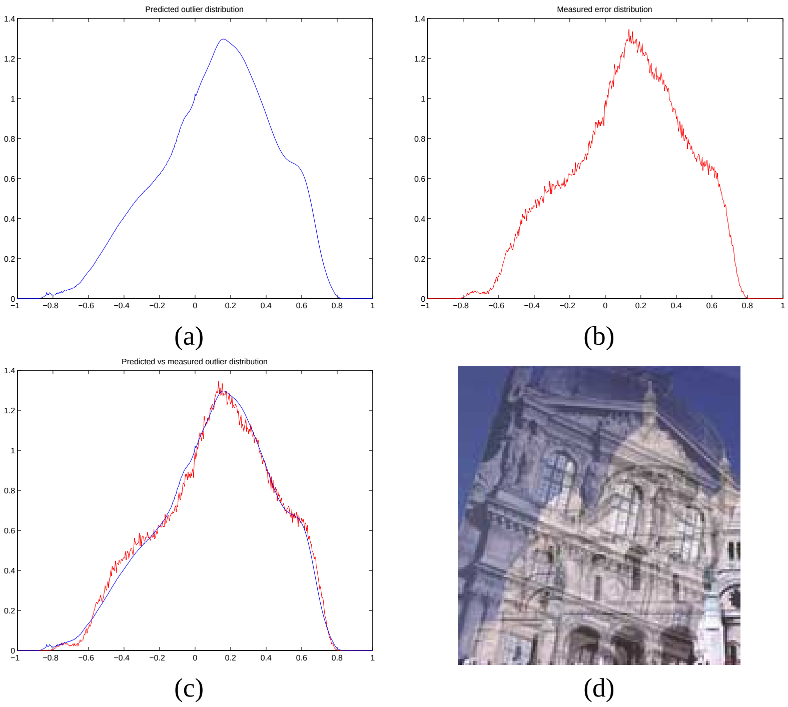
In other words, the outlier distribution is equal to the cross-correlation of the two image histograms, computed on the overlapping portion of the image pair. An experimental evaluation of (4.12) is presented in Figure 4.1 and 4.2, verifying the validity of this model. The main assumption behind this result is that the expectation of the error histogram is equal to the error histogram delivered by a single image match. Further details can be found in Appendix B.

Note that this formulation does not take advantage of the correlation between neighbouring pixels and assumes that the statistics of the part of the image that contains outliers is equal to the statistics of the whole image.



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

Figure 4.1: Evaluation of the outlier model. (a) The predicted outlier distribution. (b) The error histogram. (c) Comparison of *(a)* and *(b)*. The model and error histogram are super-imposed and can hardly be distinguished. Therefore, the model works well in this case. (d) The image superposition used to generate the data: a white noise image has been compared to a real image (the images are rendered using a transparent blending).



(a) (b)

(c) (d)

Figure 4.2: Evaluation of the outlier modelling. (a) The predicted outlier distribution (b) The error histogram. (c) Comparison of *(a)* and *(b)*. We can see that the model explains the data fairly well. The histograms are computed without taking the sky in one of the pictures into account. (d) The image superposition used to generate the data (the images are rendered using a transparent blending).

4.4 The inlier model

In an ideal situation, an inlier is an object that appears at the exact same location in the image pair, and has the exact same pixel value. In practice, the pixel values may differ because of acquisition noise, and the image might not be perfectly aligned. This slight misalignment is perfectly normal in the context of motion estimation and it is crucial to consider the slightly displaced objects as inliers; surely enough, these misaligned pixels are the ones that help the motion estimation algorithm achieve a better registration.

In order to characterise the inliers, we conduct the following experiment: we take two images of a scene and make sure that nothing changes neither in the scene, nor in the camera pose and settings, and compute the error histogram—the difference in the two pictures is due solely to the acquisition noise. Then, we slightly misalign one of the pictures, and recompute the error histogram. We repeat the experiment for different misalignment amplitudes and try to fit a model to the error histogram. The error histograms of an image pair are shown in Figure 4.3, for 0, 5 and 20 pixels of misalignment, respectively.

Figure 4.4 shows a comparison of the inlier histogram with a Laplacian distribution with zero mean[3]. The Laplacian distribution $(\mathcal{L})$ is defined by [Leon-Garcia, 1994]

$$\mathcal{L}_{(0,\sigma)}(r) = \frac{1}{2\sigma} \cdot e^{-\frac{|r|}{\sigma}}, \tag{4.13}$$

where r is the residual (the registration error) and σ is the standard deviation. Although a generalised Gaussian distribution [Aiazzi et al., 1999] would deliver a better fit, we disregarded this option because of the increased complexity that such a choice would bring: the generalised Gaussian distribution requires the estimation of two parameters instead of only one for the Laplacian. We chose to use the Laplacian distribution to model the inliers, which requires us to find its standard deviation σ that will differ with each estimation.

4.5 The *OutlierMix* model

By combining the results of the two previous sections, we can build a new model for matching two images; we call it the *OutlierMix* model. This model is a mixture of the outlier model of Section 4.3 with the inlier model of Section 4.4.

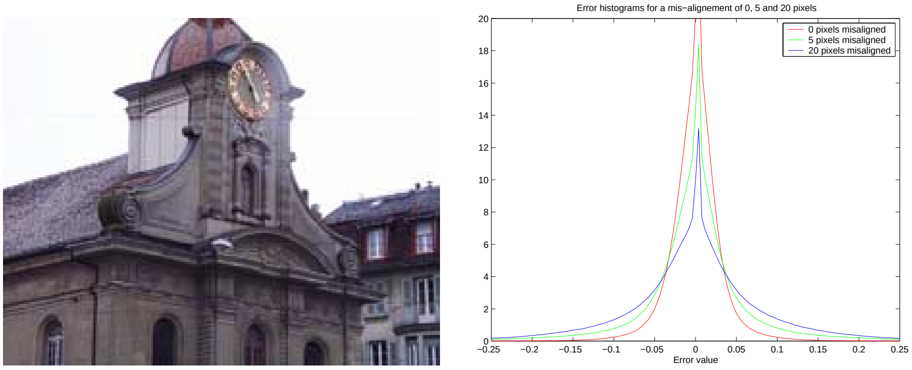
Given the outlier distribution $\mathrm{P}_{\mathcal{O}}$, we can describe the error generated by matching two images as a mixture of inliers and outliers:

$$\mathrm{P}(r) = \phi \mathrm{P}_{\mathcal{I}}(r) + (1-\phi)\mathrm{P}_{\mathcal{O}}(r) \tag{4.14}$$

$$\mathbf{H}_m = \phi \mathbf{H}_{\mathcal{I}}(\sigma) + (1-\phi)\mathbf{H}_{\mathcal{O}}, \tag{4.15}$$

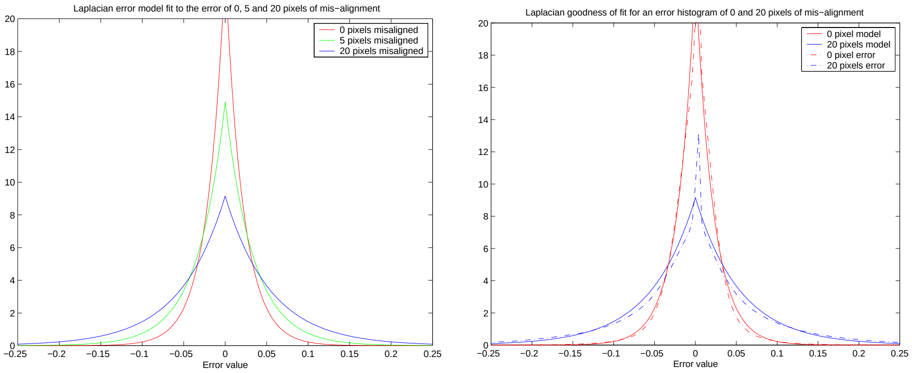
where $\mathrm{P}_{\mathcal{I}}$ stands for the inlier probability density, $\mathrm{P}_{\mathcal{O}}$ is the outlier probability density, ϕ is the proportion of inliers, and r the error value. Equation (4.15) is equivalent to (4.14) but using vector notation, where $\mathbf{H}_m$ denotes the model histogram, $\mathbf{H}_{\mathcal{I}}(\sigma)$ the inlier histogram and $\mathbf{H}_{\mathcal{O}}$ the outlier histogram. As discussed in Section 4.4, we assume that the inlier distribution is a Laplacian distribution with zero mean: $\mathrm{P}_{\mathcal{I}} \sim \mathcal{L}_{(0,\sigma)}$. The model parameters ϕ and σ are

[3] If the settings of the camera are varying (even slightly), then the registration model should compensate for these changes to ensure that the inliers are zero mean.



(a) (b)

Figure 4.3: Measuring inlier distributions. Two images looking like (a) are superimposed. (b) The error histogram for a perfect alignment, an alignment 5 pixels apart, and an alignment 20 pixels apart.



(a) (b)

Figure 4.4: Error distribution versus a Laplacian distribution. (a) 3 different Laplacian distribution computed to match the error distribution of Figure 4.3*(b)*. (b) Comparison of the Laplacian models in *(a)* with the error distribution in 4.3*(b)*. In *(b)*, the histogram showing a misalignment of 5 pixels has been suppressed for clarity reasons.

computed by fitting the model error distribution $\mathrm{P}(r)$ to the error histogram $\mathbf{H}_e$, according to Section 4.6.

Note that the outlier *distribution* depends only on the image histograms and not on the error. In other words, we can compute the outlier distribution without doing any image superpositions or any error computation. Nevertheless, to compute the outlier *proportion* (and the inlier standard deviation) we need to compute the error by superimposing the images.

4.5.1 The *UniformMix* model

A common practice is to model the outliers with a uniform distribution [Netanyahu and Weiss, 1994; Schroeter et al., 1998; Torr et al., 1999]. To compare this approach to the *OutlierMix* model, we introduce the *UniformMix* model, which is a mixture of the inlier model of Section 4.4 with a uniform distribution.

Given the outlier distribution $\mathrm{P}_{\mathcal{O}}$, we can describe the error generated by matching two images as a mixture of inliers and outliers:

$$\mathbf{H}_m = \phi\mathbf{H}_{\mathcal{I}}(\sigma) + (1-\phi)\mathbf{H}_{\mathcal{U}}, \tag{4.16}$$

$$\mathbf{H}_{\mathcal{U}}(r) \triangleq \frac{1}{N}, \quad \forall r \tag{4.17}$$

where $\mathbf{H}_m$ denotes the model histogram, $\mathbf{H}_{\mathcal{I}}(\sigma)$ is the inlier histogram, $\mathbf{H}_{\mathcal{U}}$ is the uniform histogram, ϕ is the proportion of inliers, and N is the number of possible error values r that an image matching can produce. The model parameters ϕ and σ are computed using the same procedure than for the OutlierMix model.

Note that this formulation differs from the one proposed in [Torr et al., 1999] by setting the inlier proportion ϕ according to the image content.

4.6 Fitting the model to the error distribution

Given the error distribution, the inlier model and the outlier distribution, we can find the *inlier scale* (σ) and the *inlier proportion* (ϕ) by fitting the model mixture to the measurements. The inlier scale is the standard deviation of the error generated by the inliers, and is also called the *error scale*. Three methods are proposed: the first method computes the maximum likely parameters, the second method minimises the Kullback-Leibler distance between the OutlierMix model and the error distribution[4], and the third method minimises the L_2-norm between the OutlierMix model and the error distribution. This last choice is motivated by computational complexity reasons. The experiments are then repeated using the UniformMix model.

To compare the OutlierMix model to the standard M-estimators presented in Section 4.2.1, we compute the inlier scale using the median of the error value. The inlier scale can be derived from the median m by solving $\int_0^m \mathcal{L}_{(0,\sigma)}(r)dr = 1/4$, i.e. $\sigma = m/\ln(2)$.

[4] The Kullback Leibler distance is chosen because of its relation to the *Minimum Description Length* (MDL) criterium that is often used in the context of mixture modelling [Rissanen, 1978].

4.6.1 Maximum likelihood computation

The likelihood L of the inlier scale parameter σ (defined in (4.1)) and the inlier proportion ϕ is given by [Lindsay, 1995, chap. 1]

$$L(\phi, \sigma \mid \mathrm{P}_{\mathcal{O}}) = \prod_{r} \mathrm{P}(r)^{n(r)},$$

where $\mathrm{P}(r)$ is the OutlierMix model distribution given by (4.14), $\mathrm{P}_{\mathcal{O}}$ is the outlier distribution, and $n(r)$ is the number of occurrence of the error r. Using the histogram notation of Section 4.5, the likelihood can be rewritten as

$$L(\phi, \sigma \mid \mathrm{P}_{\mathcal{O}}) = \left[\prod_{r} H_m(r)^{H_e(r)}\right]^{\alpha},$$

where $\mathbf{H}_m$ denotes the model histogram, $\mathbf{H}_e$ the error histogram and α a constant factor due to the normalisation of $\mathbf{H}_e$. To find the maximum likelihood estimate of the inlier scale σ and of the inlier proportion ϕ, we maximise the log-likelihood function $l(\cdot)$:

$$\begin{aligned} l(\phi, \sigma \mid \mathrm{P}_{\mathcal{O}}) &\triangleq \alpha \cdot \log \left[\prod_{r} H_m(r)^{H_e(r)}\right] \\ &= \alpha \cdot \sum_{r} H_e(r) \log \left[H_m(r)\right], \\ (\sigma, \phi) &= \arg\max \sum_{r} H_e(r) \log \left[H_m(r)\right], \end{aligned} \tag{4.18}$$

which is equivalent to maximizing the likelihood function. Equation (4.18) is solved numerically.

4.6.2 Kullback-Leibler distance fit

By using the Kullback-Leibler distance D [Cover and Thomas, 1991, chap. 9], the inlier scale and inlier proportion are found using

$$\begin{aligned} (\sigma, \phi) &= \arg\min D\left(\mathbf{H}_e \parallel \mathbf{H}_m\right) \qquad (4.19) \\ &= \arg\min \sum H_e(r) \log\left(\frac{H_e(r)}{H_m(r)}\right). \end{aligned}$$

The Kullback-Leibler distance expresses the error in coding length obtained by coding a random variable distributed like $\mathbf{H}_e$ by assuming it has distribution $\mathbf{H}_m$. Consequently, it is closely related to the MDL criterion and is thus a “natural” metric for distributions comparison. Equation (4.19) can be solved numerically.

4.6.3 L_2-norm fit

We used the L_2-norm because of its low computational complexity. The optimal parameter in the L_2 sense is computed using a simple line search on the inlier scale parameter σ, while

Method	Median	L_2-norm		Max-likelihood		Kullback-Leibler	
Parameter	σ	$1-\phi$	σ	$1-\phi$	σ	$1-\phi$	σ
Whole image (c)	0.028	22%	0.024	27%	0.020	27%	0.020
sub-image (d)	0.039	37%	0.024	42%	0.019	42%	0.019
sub-image (e)	0.061	52%	0.023	56%	0.018	56%	0.018
sub-image (f)	1.192	97%	0.021	98%	0.014	98%	0.014

Table 4.1: Inlier scale (σ) and outlier percentage ($1-\phi$) estimation. The table compares three methods that use the *OutlierMix* model to determine the inlier scale and the outlier percentage of the images in Figure 4.5. The inlier scale, which should be approximately constant, is also compared to a standard method that uses the median estimator. The three methods deliver similar results and are superior to the median based estimator.

the Kullback-Leibler distance and the maximum likelihood estimator require the computation of a logarithm for each error value, and require a two-variable regression to find the optimal parameters. Unfortunately, the L_2-norm approach does not guarantee that the inlier proportion ϕ is contained in the $[0;1]$ interval, and is not, in general, a "natural" measurement for distributions.

The optimal parameters in the L_2 sense are found by solving the following equation:

$$(\sigma,\phi) = \arg\min \|\mathbf{H}_e - \mathbf{H}_m\|_2$$

under the constraint

$$\phi \in [0;1].$$

If the inlier scale parameter σ is known, the inlier proportion ϕ can be found by taking the minimum mean squared solution of the following equation system:

$$\mathbf{H}_e = \phi \cdot \mathbf{H}_{\mathcal{I}} + (1-\phi)\cdot \mathbf{H}_{\mathcal{O}},$$

i.e.

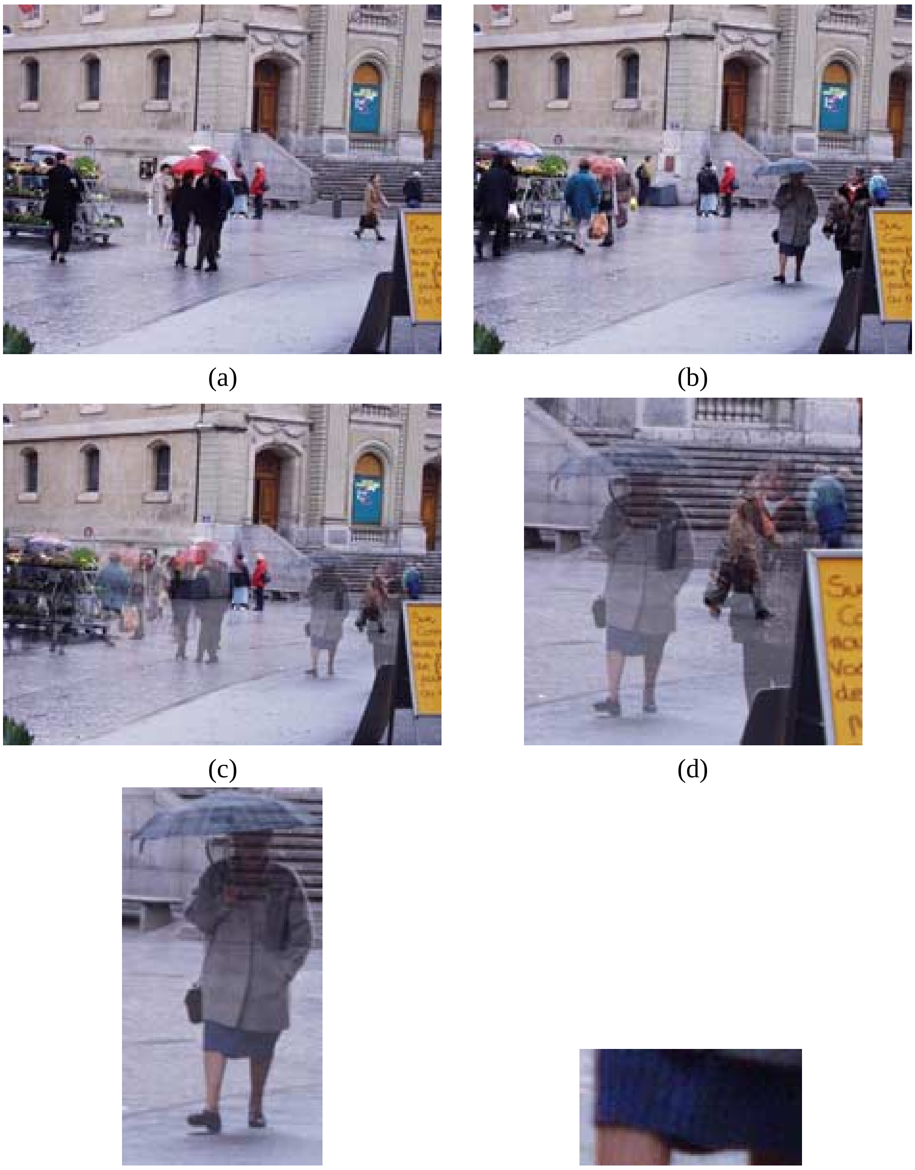
$$\phi = \frac{(\mathbf{H}_{\mathcal{I}} - \mathbf{H}_{\mathcal{O}})^T (\mathbf{H}_e - \mathbf{H}_{\mathcal{O}})}{\|\mathbf{H}_{\mathcal{I}} - \mathbf{H}_{\mathcal{O}}\|^2}.$$

In practice, σ is found using an exhaustive search given a maximum a-priori value, which in our case is set to 20% of the image's dynamic range[5].

4.6.4 Results

To illustrate the behaviour of each fitting method, two pictures of a pedestrian street are shown. The outliers are the pedestrians along with their shades, and the inliers are the background regions of the scene. The outlier proportion is varied by extracting from the image in Figure 4.5(a) and 4.5(b) several sub-images, which contain less and less background information, as illustrated in Figure 4.5(c) to 4.5(f). Table 4.1 shows the result of fitting the

[5] Special care has to be taken in finding σ because of possible local minima.



(a) (b)

(c) (d)

(e) (f)

Figure 4.5: (a)(b) Original pictures used to perform outlier detection tests. (c) to (f) Ghost pictures used to perform the outlier tests. The images are super-imposed using transparency, hence the outliers appear as a ghost image. (c) Whole image (d) Region of picture *(c)*, (e) Region of picture *(d)*,(f) Region of picture *(e)*. The sub-images are chosen such that the part of the image covered by the outliers is increasing.

Method	L_2-norm		Max-likelihood	
Parameter	$1-\phi$	σ	$1-\phi$	σ
Whole image (c)	19%	0.025	20%	0.025
sub-image (d)	37%	0.024	34%	0.023
sub-image (e)	42%	0.026	45%	0.024
sub-image (f)	98%	0.013	98%	0.012

Table 4.2: Inlier scale (σ) and outlier percentage ($1-\phi$) estimation. The table compares 2 methods that use the UniformMix model to determine the inlier scale and the outlier percentage of the images in Figure 4.5. The inlier scale estimate, which should have approximately a constant value, is not as robust as in the experiment in Table 4.1.

OutlierMix model to the error histogram. The sensitive parameter is the inlier scale; since the background is similar for every image pair, the inlier scale value should not change within the table[6]. The estimator based on the Kullback-Leibler distance gives the same result than the maximum likelihood estimator (less than 1% in difference). The L_2 estimator differs slightly from the other two, but gave also very satisfactory results. Although the relative differences between the estimates of the inlier scale appears to be large, its absolute value is very small compared to the dynamic range of the error ($[-1;1]$).

The outlier modelling is also compared to the UniformMix model, which assumes that the outliers can be represented by a uniform distribution. The results shown in Table 4.2 are not as good as the ones using the OutlierMix modelling, but are still better than the traditional median-based approach.

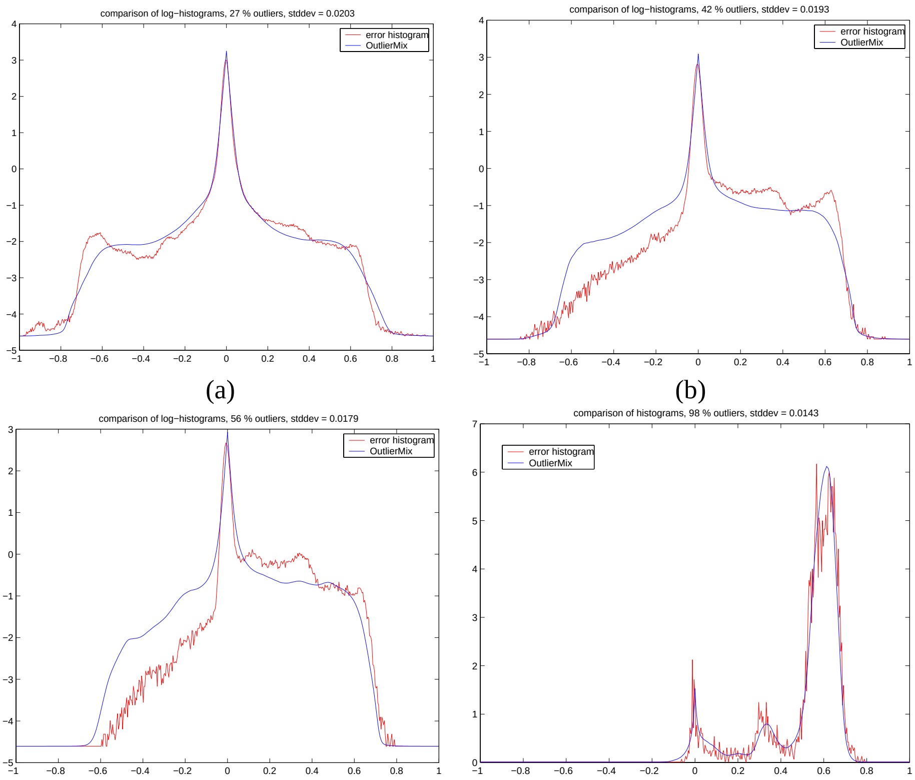
The modelling fit is shown in Figure 4.6 for the OutlierMix model using the maximum likelihood estimator. The fit is only shown for the maximum likelihood estimator because, graphically, there is hardly a difference with the L_2 estimator. Figure 4.7 shows the *outlier mask*. The outlier mask represents, for each pixel, the probability to belong to the outliers—defined by (4.14)—where the greyscale is linearly related to the probability to be an outlier (black is used for high probability of outliers). We can see that the mask is able to discriminate the inliers from the outliers even in the presence of more than 95% outliers.

4.7 Motion model based on the OutlierMix model

From the OutlierMix model, we can derive a new M-estimator by associating a weight $W(r)$ to the error value r of each pixel, according to whether the pixel is an outlier or not. Specifically, we set $W(r)$ equal to its probability to belong to the inliers:

$$\begin{aligned} W(r) = \mathrm{P}(\text{inlier} \mid r) &= \frac{\mathrm{P}(\text{inlier}) \cdot \mathrm{P}(r \mid \text{inlier})}{\mathrm{P}(r)} \\ &= \frac{\phi H_{\mathcal{I}}(\sigma, r)}{\phi H_{\mathcal{I}}(\sigma, r) + (1-\phi) H_{\mathcal{O}}(r)}. \end{aligned} \tag{4.20}$$

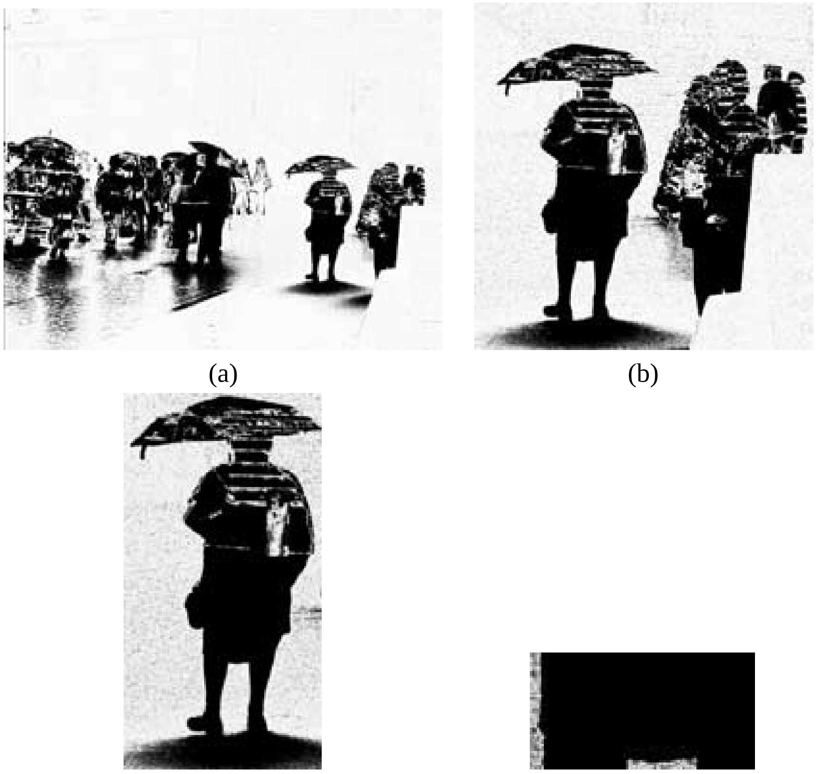
[6] Strictly speaking, some little changes could occur since it is not exactly the same background due to the different framing.



(a) (b)

(c) (d)

Figure 4.6: Fitting the OutlierMix model to the error histogram with a maximum likelihood approach. (a) 27% outliers. (b) 42% outliers. (c) 56% outliers (d) 98% outliers. The graphs correspond to the images of Figure 4.5, and are shown on a logarithmic scale, except for *(d)*. The smooth (blue) curve shows the OutlierMix model.



(a) (b)

(c) (d)

Figure 4.7: Outlier masks associated to the images in Figure 4.5. The grey scale is proportional to the probability to be an outlier; black is used for high probability of outliers (a 2.2 *gamma correction* is used to render the images).

In the context of motion estimation, the weight factor is introduced by the derivative $\frac{\dot{\rho}(r_i)}{r_i}$ of the objective function defined in Section 4.2.1. We set this derivative equal to the weight of Equation (4.20):

$$\frac{\dot{\rho}(r_i)}{r_i} = W(r),$$

and compute an objective function from it:

$$\begin{aligned} \rho(r) &= \int_0^r \dot{\rho}(e)de \\ &= \int_0^r \frac{\dot{\rho}(e)}{e} \cdot e \cdot de \\ &= \int_0^r W(e) \cdot e \cdot de. \end{aligned} \tag{4.21}$$

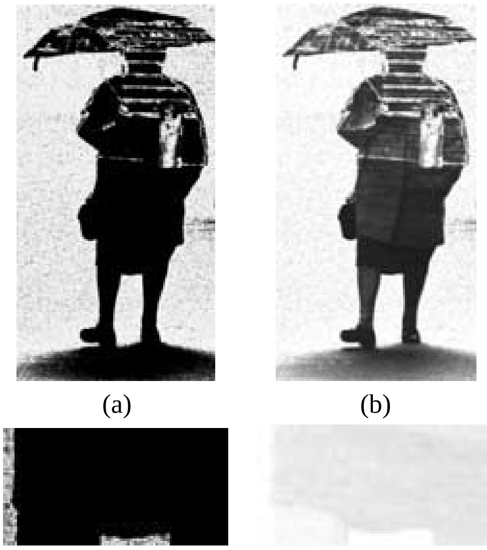
The objective function is set to 0 at the origin: $\rho(0) = 0$, and is computed numerically. Finally, the motion estimator using the OutlierMix model minimises the objective function in Equation (4.21). Note that this function is often asymmetric.

4.8 Robustness of the OutlierMix model

The traditional robustness computation approach evaluates the robustness of an estimator provided that a certain amount of samples are replaced by arbitrary values [Huber, 1981]. Unfortunately, it is not possible to characterise the performance of the OutlierMix model using this approach, because our model knows a priori all the possible values that the error can take. Consequently, it does not fit in the traditional framework of robustness characterisation. Nevertheless, in our context, we can argue that the model can handle an arbitrary percentage of outlying data, since the outlier detection is just a matter of choosing the proportion among two distributions. Hence, we propose two experiments to measure the performance of the OutlierMix model.

The first experiment evaluates the robustness of the scale estimate (σ) to outliers. Table 4.1 shows the estimate of the error scale for the pictures of Figure 4.5. The error scale should remain approximately constant. We can see that the estimate derived from the error median—the method generally used by the standard estimators—is stable for the cases of less than 50% outliers. Above 50%, it becomes unstable and totally fails in presence of 98% of outliers. The same phenomenon is illustrated in Figure 4.8, where the weighting factor of the Lorentzian model, defined in (4.5) is shown for every pixel. The figure shows the result obtained considering images 4.5(e) and 4.5(f). The first column of Figure 4.8 shows the weight based on the scale found with the OutlierMix model fit with the L_2-norm and the second column shows the weight based on the scale computed with the error median. More details are visible in the outlier of Figure 4.8(b) than in Figure 4.8(a), thus, the M-estimator gives a variable weighting to the pixels of the outlier, instead of rejecting them uniformly, as in Figure 4.8(a). More striking is the difference between images 4.8(c) and 4.8(d), where the traditional estimator is unable to distinguish the inliers from the outliers. This shows that the scale estimate using the OutlierMix model is more robust.

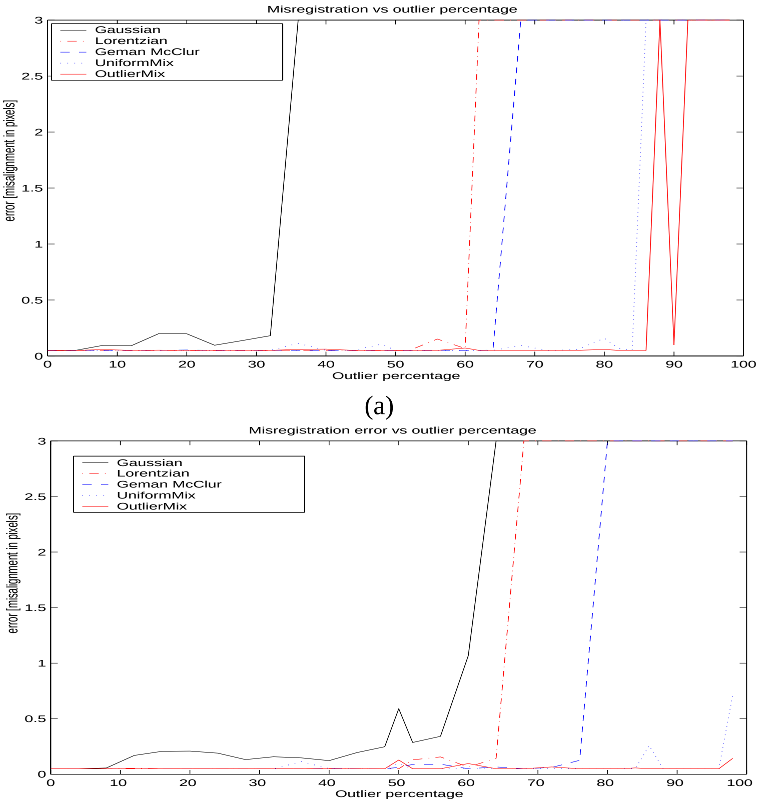
The second experiment compares the performance of a translation estimation with an M-estimator using a Lorentzian and Geman-McClur distribution (see Section 4.2.1) to the



(a) (b)

(c) (d)

Figure 4.8: Comparison of outlier masks for the OutlierMix model in (a) and (c) and a traditional M-estimator using the median to compute the error scale in (b) and (d). We can see in (d) that the traditional estimator fails to distinguish the outliers from the inliers. These masks are computed from the images in Figure 4.5, and are rendered using a 2.2 gamma correction.



(a)

(b)

Figure 4.9: Comparison of different estimators measured for two initial conditions. (a) 15 pixels of initial misalignment—median result for 5 tests. (b) 1.5 pixels of initial misalignment. A translation is estimated between 2 pictures, and the final translation error is shown. The OutlierMix model is the most robust of the set, handling up to 90% outliers in the first case and 98% in the second. There is some randomness associated with the test: for example the outlier-based model is able to handle 90% of outliers but fails with 88%.

motion estimator using the OutlierMix and the UniformMix models. Two pictures of the same scene are taken in the same conditions with a fixed camera. One of the pictures has been corrupted, by taking the left part of the picture and copying it to the right, as illustrated in Figure 4.10(b). By extracting a sub-image pair from these images, we can control the amount of outliers present in the pair. Then, for a whole set of image pairs, a motion estimation is performed[7], starting from a misalignment of 15 pixels in amplitude. The results—shown in Figure 4.9(a)—are binary: either the algorithm converges and the error is insignificant, or it does not converge and the resulting error is large. The misregistration is measured as the distance between the real position of the image (known a priory) and the one delivered by the motion estimation. Five tests are conducted[8] and the graph shows the median of the results. The model using the OutlierMix model clearly outperforms the traditional ones, and is able to handle 86% to 90% outliers in this particular example[9]. For the same example, the Gaussian M-estimator handles up to 32% outliers, the Lorentzian M-estimator handles up to 60% outliers and the Geman-McClur M-estimator handles up to 64% outliers. The UniformMix model handles up to 86% outliers and is therefore better than the traditional M-estimators but worse than the OutlierMix model. We want to emphasize that this model is sometimes used [Torr et al., 1999] without estimating the outlier proportion $(1 - \phi)$, which is one of the key elements in the success of this implementation. The experiment is reproduced for an initial misalignment of 1.5 pixels in amplitude, showing similar results.

In general, the performance of the OutlierMix model depends on the image content: the more the outlier data differs from the inlier data, the better the OutlierMix model performs compared to a traditional approach.

4.9 Computational complexity and estimation efficiency

The OutlierMix model is approximately as complex, in terms of computation cost, as the usual M-estimators. In addition to the computation required by the M-estimators, the OutlierMix model has to compute two image histograms (once), and requires an extensive search on the scale parameter of the inlier model. Nevertheless, the extensive search is based on histograms that contain in practice around 500 samples, which is very small in comparison to the number of pixels in an image. Additionally, a better estimate of the scale parameter leads to faster convergence, hence, the algorithm needs fewer steps to converge to the solution. The UniformMix model has the same complexity than the OutlierMix model, if we do not take the two initial image histograms into account.

A nice property of the OutlierMix estimator is its relative efficiency: The relative efficiency is defined as the ratio between the lowest achievable variance for the estimated parameters (i.e. the Cramer-Rao lower Bound) and the actual variance provided by the given method [Ayer, 1995]. In the absence of outliers, the outlier-based model reduces to a standard least square estimation which reaches the Cramer-Rao lower bound. In other words, the OutlierMix motion model achieves a relative efficiency of 1 in absence of outliers. Indeed,

[7] The estimation is performed using a two parameter translation model and a Gaussian multi-resolution pyramid with 4 levels.

[8] 5 different initial conditions are used, each of them with the same misalignment amplitude.

[9] The randomness of the experiment made the estimator fail at 88% 4 times out of 5, but succeeded at 90% 3 times out of 5.

the probability to be an inlier becomes 1 in this case, leading to a weight $W(r) = 1$ in (4.21) that generates an objective function $\rho(r) = r^2$.

4.10 Limitations

In the examples shown so far, the outlier model of Section 4.3 has proven to explain the data quite well. There is, however, a particular case where the model does not fit the data very well, which occurs when the image contains a large uniform area. For example, in Figure 4.2(d), more than $1/3$ of the image is covered by a uniform blue sky. To get the result of Figure 4.2(a), (b) and (c), we discounted the influence of the sky. If we take the sky into account, the error histogram exhibits a peak value that is not as accentuated in the outlier model. In this case, the expectation of the error distribution is not equal to the error distribution of a single realisation of the comparison process. Indeed, when the areas of the pictures are too uniform, the error distribution tends to be sensitive to the way the pictures are compared. Nevertheless, except for the peak values, the model still fits the data quiet well, as shown in Figure 4.11. This also explains why the fit in Figure 4.1 is better than the fit in Figure 4.2. Now, if we conduct several experiments with the images of Figure 4.2(d) and take the average of the resulting error histograms, then the error histogram tends to approach the outlier model, showing that the divergence observed in Figure 4.11(a) is indeed due to an expectation problem. Figure 4.11(b) shows the average of 10 error histograms obtained using 10 independent random superpositions.

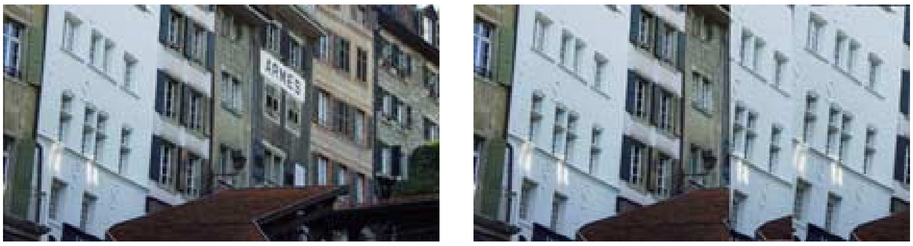
4.11 An outlier model for a model mixture

In Section 4.3, we proposed to compute the outlier distribution based on the assumption that an outlier comes from a random superposition of an image pair. Now, if we consider the multi-planar algorithm of Section 3.3.2, and more generally the family of motion models based on mixtures [Torr et al., 1999; Wang and Adelson, 1993; Jepson and Black, 1993; Weiss and Adelson, 1996; Ju et al., 1996; Sawhney and Ayer, 1996; Black and Anandan, 1996], we notice that there is not one, but several image superpositions that come into play. Out of these superpositions, for each pixel value, the algorithm chooses the one that best fits one of the models. Now, let us suppose there are 10 models in the mixture. The pixel value is compared to 10 pixel values in the other image of the pair, whose locations are defined by the 10 motion models of the mixture. In the presence of an outlier, these 10 locations are not consistent with the image and can as well be considered as being random. So, in our example, the outlier distribution can be computed by assuming that a pixel of the image is compared to 10 pixel values whose positions are chosen at random in the image, and that for each comparison, we pick the best one (the one whose error is minimum).

More specifically, the “outlier situation” occurs by supposing that every motion estimate is completely wrong. The obtained error value is the minimum absolute error value out of every random superposition. For two models, it can be written as follow:

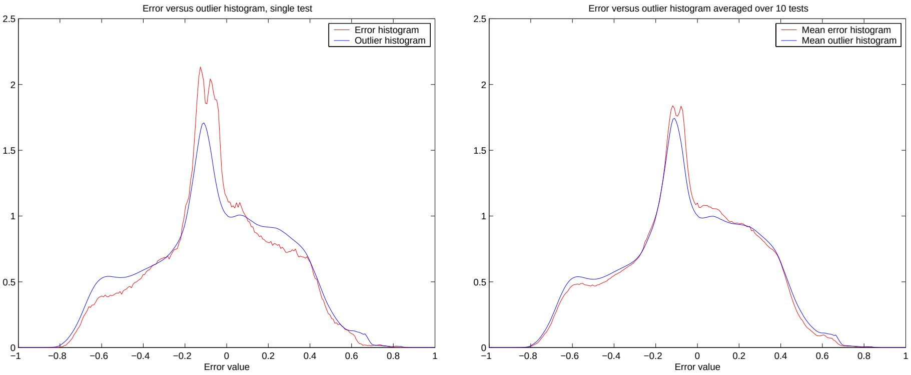
$$p_{\mathcal{O}}^{(2)}(\varepsilon) = \Pr\left(\min\left\{\left|I_0(\mathbf{p}) - I_1(\mathbf{p}^{(1)})\right|, \left|I_0(\mathbf{p}) - I_1(\mathbf{p}^{(2)})\right|\right\} = \varepsilon\right),$$

where $I_1(\mathbf{p}^{(j)})$ is the part of the scene I that is projected on the image using the random motion parameters $\mathbf{\Theta}_j$, like in Section 4.3 (i.e., $I_1(\mathbf{p}^{(j)}) = I(\mathbf{\Theta}_j)$), which, in practice, can



(a) (b)

Figure 4.10: Images used to measure the robustness to outliers of different estimators, shown in Figure 4.9. The right part of (b) is corrupted, and the outlier percentage is chosen by extracting a sub-image pair and carefully controlling the framing of the extraction.



(a) (b)

Figure 4.11: Comparison between the predicted outlier histogram—smooth (blue) curve—and the measured error distribution of the superposition of the pictures depicted in Figure 4.2(d). The whole figure is used to compute the histograms. (a) In some locations, the data exhibits some peak values that are not present in the modelling. These peak values are generated by the uniform areas contained in the images. Because of the uniformness of these areas, the matching does not, in general, lead to the expectation of the matching, thus explaining the divergence of the model from the data. (b) Shows the average over 10 experiments, showing that the error histograms tends toward the outlier model.

be computed from image I_1. Now let $r_j = \left|I_0(\mathbf{p}) - I_1(\mathbf{p}^{(j)})\right|$. Since the parameters $\mathbf{\Theta}_j$ are considered random and independent, the location $\mathbf{p}$ and $\mathbf{p}^{(j)}$ are independent and finally, the error values r_j are independent. The cumulative distribution of the error for the two plane situation can be expressed as

$$\begin{aligned} p_{\mathcal{O}}^{(2)}(\varepsilon) &= \Pr\left[\min(r_1, r_2) = \varepsilon\right] \\ &= \Pr\left[r_1 = \varepsilon, r_2 > \varepsilon\right] + \Pr\left[r_2 = \varepsilon, r_1 > \varepsilon\right] + \Pr\left[r_1 = \varepsilon, r_2 = \varepsilon\right]. \end{aligned}$$

Since the outliers have all the same distribution, and the models are independent, we get

$$\begin{aligned} p_{\mathcal{O}}^{(2)}(\varepsilon) &= 2 \cdot \Pr\left[r = \varepsilon\right] \cdot \Pr\left[r > \varepsilon\right] + \Pr\left[r = \varepsilon\right]^2 \\ &= 2 \cdot \Pr\left[r = \varepsilon\right] \cdot \left(1 - \Pr\left[r \leq \varepsilon\right]\right) + \Pr\left[r = \varepsilon\right]^2. \end{aligned}$$

By introducing the cumulative absolute error distribution $F_{\mathcal{O}}$, we get

$$\begin{aligned} F_{\mathcal{O}}(\varepsilon) &\triangleq \sum_{u=0}^{\varepsilon} p_{\mathcal{O}}(u) = \Pr\left[r \leq \varepsilon\right] \\ p_{\mathcal{O}}^{(2)}(\varepsilon) &= 2 \cdot p_{\mathcal{O}}(\varepsilon) \cdot \left[1 - F_{\mathcal{O}}(\varepsilon)\right] + p_{\mathcal{O}}(\varepsilon)^2. \end{aligned} \tag{4.22}$$

Now considering M models by applying (4.22) in a recursive manner, gives

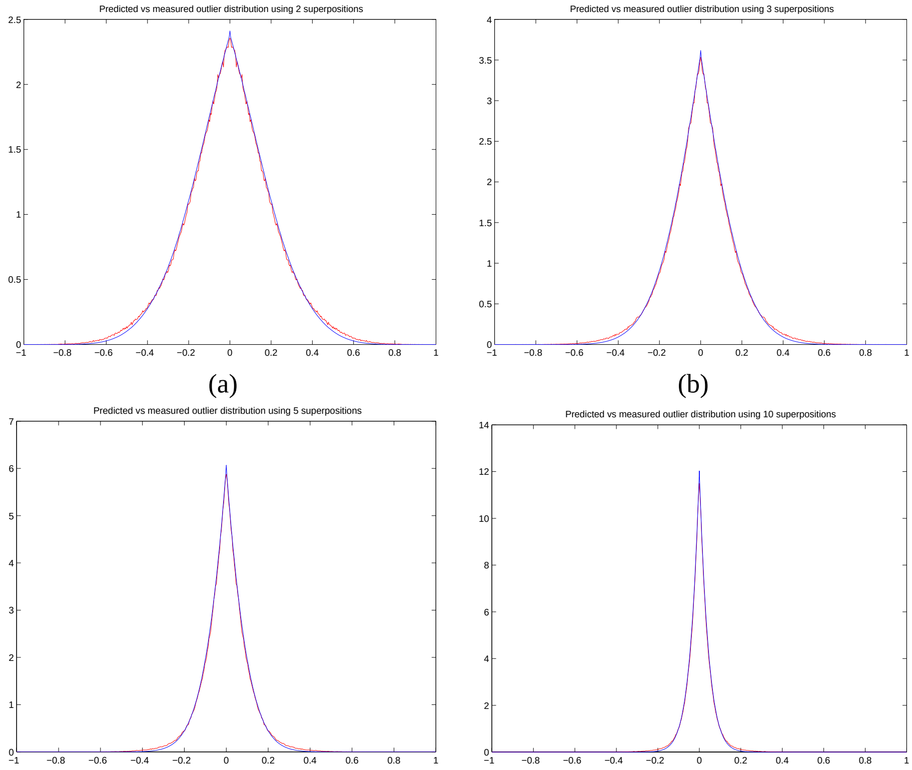
$$\begin{aligned} p_{\mathcal{O}}^{(M)}(\varepsilon) &= \sum_{j=1}^{M} \begin{pmatrix} M \\ j \end{pmatrix} p_{\mathcal{O}}(\varepsilon)^j \left[1 - F_{\mathcal{O}}(\varepsilon)\right]^{M-j} \\ &= -\left[1 - F(\varepsilon)\right]^M + \sum_{j=0}^{M} \begin{pmatrix} M \\ j \end{pmatrix} p_{\mathcal{O}}(\varepsilon)^j \left[1 - F_{\mathcal{O}}(\varepsilon)\right]^{M-j} \\ &= \left(p_{\mathcal{O}}(\varepsilon) + \left[1 - F_{\mathcal{O}}(\varepsilon)\right]\right)^M - \left[1 - F_{\mathcal{O}}(\varepsilon)\right]^M. \end{aligned} \tag{4.23}$$

This formula can be shown by induction. It shows that the outlier distribution tends to get more and more energy around 0 by increasing the number of models. In other words, it shows that if there are too many models present in the scene, it will be very difficult to distinguish the inliers from the outliers and consequently it will be hard to segment the image using a competitive mixture of models approach. The evolution of the outlier distribution is shown in figures 4.13 and 4.12.

4.11.1 Experimental evaluation

To verify experimentally the multi-outlier model of Equation (4.23) we repeat the experiment of Section 4.3 by super-imposing randomly two images. This time, the second image is super-imposed n times, and for each pixel the best value is kept to construct the error histogram. The two image pairs of Figure 4.1 and 4.2 are used. The superposition of the model histogram and the error histograms are shown in figures 4.13 and 4.12. The result shows a close match between the model and the measurements. The histograms are symmetric around the origin because the model considers only the absolute error value.

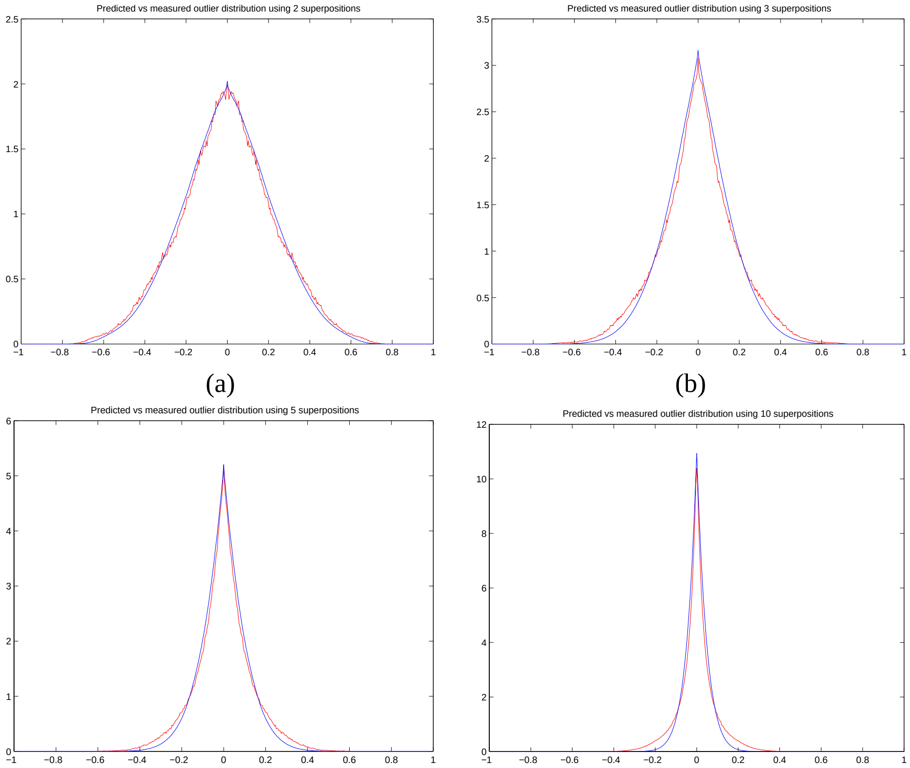
The most important fact to retain from this experiment is the evolution of the variance of the outliers with the number of models. The first observation is that the outlier distribution



(a) (b)

(c) (d)

Figure 4.12: Evolution of the outlier distribution with the number of models in a mixture, by comparing an image made of noise with a real image. (a) 2 models (b) 3 models, (c) 5 models, (d) 10 models. The plots show that the probability that an outlier generates a low registration error increases with the number of models. It also shows a very close match between the theory and the measurements, since each of the plot contains two curves - the measured and the modelled one - that can hardly be distinguished.



(a) (b)

(c) (d)

Figure 4.13: Evolution of the outlier distribution with the number of models in a mixture, by comparing two real images. (a) 2 models (b) 3 models, (c) 5 models, (d) 10 models. The plots show that the probability that an outlier generates a low registration error increases with the number of models. It also shows a very close match between the theory and the measurements, since each of the plot contains two curves - the measured and the modelled one - that can hardly be distinguished.

tends to have the same Laplacian shape as the inlier distributions of Section 4.4. Additionally, the variance of the outlier distribution for a mixture of 10 models is about the same than the variance of the inlier distribution for a mis-alignment of 20 pixels in Figure 4.4(a). In this case, it would be very difficult - if not impossible - to distinguish the inliers from the outliers. This points to a fundamental limitation of the motion models based on mixtures. This is why, in general, the segmentation algorithms [Torr et al., 1999; Weiss and Adelson, 1996; Ju et al., 1996; Sawhney and Ayer, 1996; Black and Anandan, 1996] consider in some way the relationship between neighbouring pixels. In the latter situation the multi-outlier model restriction does not apply anymore. Although the need of taking the neighbourhood relationships into account to segment an image seems obvious, the multi-outlier description is able, is some sense, to give a bound on the necessary conditions for a mixture model algorithm, which considers each pixel individually, to converge.

4.12 Conclusions

In this chapter, we present a new way of calculating outliers in image pairs. The method differs from the traditional approach by characterising the outliers with a distribution computed from the initial images. This restriction in the outlier characterisation allows the description of a new motion estimator that treats the motion problem as a mixture of inliers versus outliers, and is able to handle outlier percentages that exceed 50% of the image. The model has been tested using two kinds of experiments: The first experiment tests the ability of the model to discriminate the outliers from the inliers using two pictures taken with a fixed camera. The second experiment tests the performance of the motion estimator derived from the OutlierMix model. Both experiments show a substantial improvement compared to the standard techniques in use.

The outlier modelling is also extended to the class of motion estimators based on competitive mixture models. This extension allows to point out some limitations associated to the family of problems based on competitive mixtures, by giving some (experimental) conditions under which the mixture-based algorithms is likely to fail.

These results can also serve different purposes: For segmentation applications, or coding applications, the OutlierMix model can be used to separate the moving objects from the background without using arbitrary thresholds on the error values. The overall outlier proportion in an image pair can also be used to perform change detection.

Further research will investigate the extension of this modelling to colour images. An interesting extension would also be to consider the relationships of neighbouring pixels in this model.

Chapter 5

Equalising Colour across Pictures

5.1 Introduction

Until now, we considered the geometric aspects of image mosaicking, or how to properly align the images of a mosaic. To get a good panoramic image, it is also necessary that the colours in the images are matched. Indeed, if the colours do not match, then the boundary between the pictures will be clearly visible. To get consistent colours in the mosaic, an object of the scene should be represented by the same colour, regardless in which pictures it appears. This would be the case if the colour in the image represents the physical properties of the objects in the scene. In reality, this is not the case, because the mapping of the colours in an image has been developed by the photographic community with the only goal of getting the “best looking picture”[1]. Consequently, the colours in the different pictures do not match in most of the cases. There are two approaches to deal with the problem : the one is to take the pictures in such a way that the images will match in colours, the other is to find a way to correct for the colour mismatch.

To get a series of pictures that are consistent from a colorimetric point of view, it is necessary to keep the exposure settings of the camera constant during the whole capturing process. With an analog camera loaded with a transparency film, it should be sufficient. When using a negative film, the printed pictures go through an image dependent process, and a colour mismatch will occur. With a digital still camera, in addition to the exposure, the white point should also be kept constant, and the camera itself should not do any image dependent processing. All these conditions for matching the colours motivated the development of algorithms to perform the colour matching. Being able to choose the settings for every picture has the advantage of getting the best possible picture for every part of the scene. Indeed, if the exposure has to be kept constant over the whole mosaic, then there will be some under-exposed and some over-exposed pictures in the range, affecting the final image quality.

To match the colours in a panoramic image, our approach retrieves the physical properties of the objects in the scene and then renders the scene as single picture. Since the physical properties of an object do not depend on the picture it appears in, this approach gets rid of the mismatch in colours. The physical properties that are needed are the Red, Green and Blue

[1] Why a linear picture - the one that best represent the physical entity of the scene - looks unsatisfactory in practice is related to the way our visual system works, and will not be discussed here.

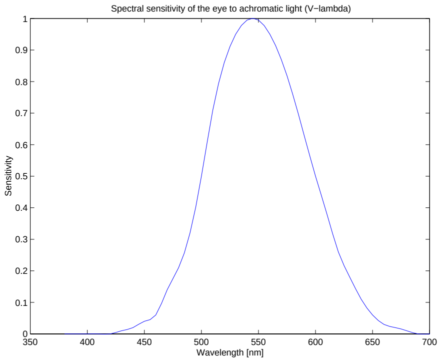


Figure 5.1: Overall spectral sensitivity of the human eye : $V(\lambda)$.

reflectance components of each entity in the scene. The emphasis is put on the estimation of these physical properties.

This chapter exposes, in Section 5.2, the general model that describes how the camera maps the scene to the pixel values of an image. Section 5.3 exposes several models for the camera tone mapping transfer functions, for digital or analog cameras, which are part of the more general modelling of Section 5.2. Then follows a description of the *vignetting* phenomenon that is inherent to any camera, and that can be corrected in some cases. Section 5.5, explains how the modelling can be applied for estimation, in the context of mosaicking. Finally, some results are shown for each of the proposed method.

5.1.1 Definitions and units

There are two kinds of terms and units related to the light: *radiometric* and *photometric* terms. Radiometric units consider each wavelength with the same importance. Photometric units use a weighting function $V(\lambda)$ for evaluating the amount of light in a mixture of radiations of different wavelength. The function $V(\lambda)$ represents the spectral sensitivity of the human eye to achromatic light under photopic light conditions, and is depicted in Figure 5.1. Below follow some photometric quantities, taken from [Hunt, 1998]:

Photopic vision : Vision by the normal eye when it is adapted to levels of luminance of at least several candelas per square metre. (The cones are the principal photoreceptors that are active in photopic vision).

Scotopic vision : Vision by the normal eye when it is adapted to levels of luminance less than some hundreds of candela per square metre. (The rods are the principal photoreceptors that are active in scotopic vision)

Luminous flux Power emitted, transferred or received in the form of radiation, weighted by the $V(\lambda)$ function.

Lumen, $[\mathrm{lm}]$: unit of luminous flux. The luminous flux of a beam of monochromatic radiation whose frequency is 540×10^{12} hertz, and whose radiant flux is $1/683$ watt. The frequency of 540×10^{12} is closely equal to a wavelength of 555 nm, the wavelength for which the $V(\lambda)$ function has its maximum value of 1.0.

Intensity I : Luminous flux per unit solid angle.

Candela, $[\mathrm{cd}]$: Unit of luminous intensity. The luminous intensity, in a given direction, of a point source emitting 1 lumen per steradian.

Illuminance E : luminous flux per unit area incident on a surface.

Lux, $[\mathrm{lx}]$: Unit of illuminance. The illuminance produced by a luminous flux of 1 lumen uniformly distributed over a surface area of 1 square metre.

Luminance L: In a given direction, at a point in the path of a beam, the luminous intensity per unit projected area (the projected area being at right angles to the given direction).

Candela per square metre, $[\mathrm{cd}/\mathrm{m}^2]$: Unit of luminance. The luminance produced by a luminous intensity of 1 candela uniformly distributed over a surface of 1 square metre.

Troland, $[\mathrm{Td}]$: Unit used to express a quantity proportional to retinal illuminance produced by a light stimulus. When the eye views a surface of uniform luminance, the number of trolands is equal to the product of the area in square millimetres of the limiting pupil, natural or artificial, times the luminance of the surface in candelas per square metre.

Remark. A CCD measures *illuminance*. The difference between *luminance* and *illuminance* is of two kinds: First *illuminance* is light falling on an object, whereas *luminance* is an entity to characterise light sources, which have a certain (non-zero) surface. Now, if we place a sensor right next to a perfect diffusing material, the illuminance of the sensor is equal (up to a constant factor) to the luminance of the surface.

Here are some basic formulas to handle these photometric quantities: The inverse square law

$$E = \frac{I}{d^2}$$

relates the illuminance E to the intensity I and to the distance d from the light source. The luminance L is related to the illuminance through

$$L = \frac{E \cdot R}{\pi},$$

where R is the reflection or the transmission factor of the illuminated surface, depending on the position of the light source. The luminous W_L flux is related to the radiant flux W (i.e. the spectral power) through

$$W_L = \frac{1}{683} \int_\lambda W(\lambda) \, V(\lambda) \, d\lambda.$$

Figure 5.2 illustrates the photometric units. A point source of 1 lm illuminates a sphere portion seen with an angle of 1 steradian. The sphere portion has a surface of 1 m ^{2} and has a luminance of 1 cd/m^2, by assuming it is made out of perfect diffusing material (i.e. there is no energy loss). The illuminance on the sphere is equal to 1 lx. By redrawing the portion of the sphere with twice its radius, but with the same angular size, one would get an illuminance of $1/4$ lx but still a luminance of 1 cd/m^2.

5.2 From the scene to the pixels

Technically, photography is all about light. The light is generated by a light source (the sun or a lamp), hits a particular object of the scene, is reflected and transformed by the object, and reaches the sensor of the camera after passing through the lens. The way the light is transformed by an object depends on the *spectral reflectance properties* of the object. The

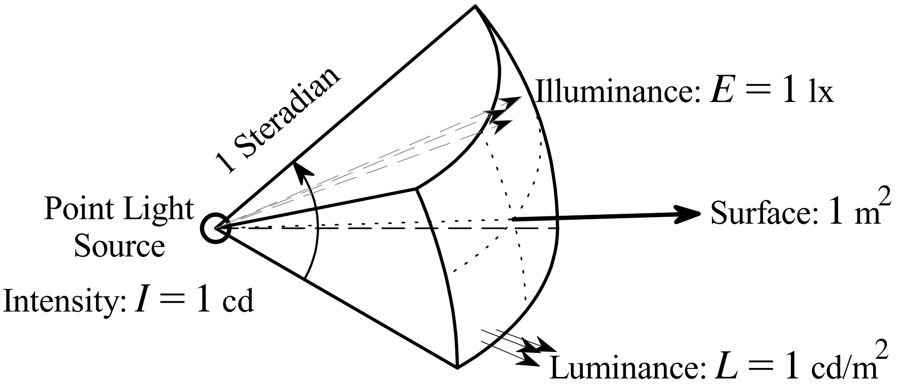


Figure 5.2: Illustration of the photometric units : A source of 1 lm illuminates the sphere portion, which has a surface of 1 m^2. The illuminance falling on the sphere is 1 lx, and the sphere has a luminance of 1 cd/m^2, by assuming it is a perfect diffuser.

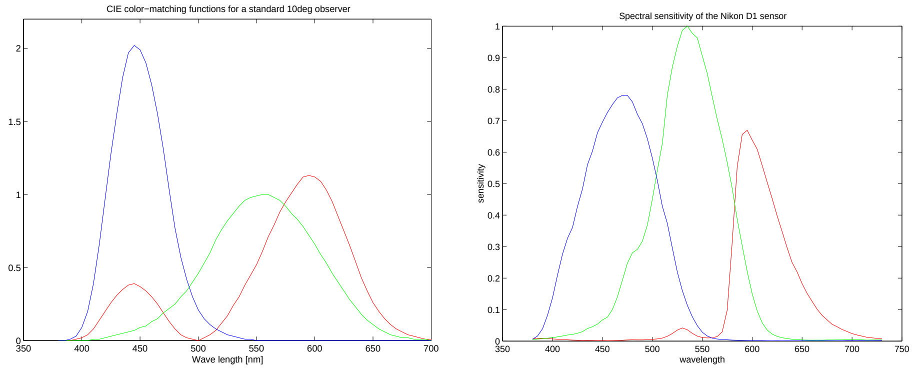
light is then recorded by the camera sensor (CCD) that measures for each pixel three light components : the Red, Green and Blue component. For a given point in the scene, the response of the camera is given by the scalar product of the spectrum of the light and the sensitivity of the camera, i.e.

$$[R, G, B]^T = \int_\lambda E(\lambda) \cdot R(\lambda) \cdot [S_R(\lambda), S_G(\lambda), S_B(\lambda)]^T \; d\lambda \tag{5.1}$$

where $E(\lambda) \cdot R(\lambda)$ is the spectrum of the incoming light, $R(\lambda)$ is the spectral reflectance of the point in the scene, $E(\lambda)$ is the spectrum of the light source, $S_R(\lambda)$, $S_G(\lambda)$ and $S_B(\lambda)$ are the sensitivities of the three types of receptors in the camera and λ is the wavelength. For any point in space, $E(\lambda) \cdot R(\lambda)$ is assumed to be constant. A CCD has a linear behaviour and the value for each pixel is proportional to the illuminance $E(\lambda) \cdot R(\lambda)$ at the sensor location. At this level, the data is called *raw data*. Then, the camera usually performs a white balancing followed by a non-linear mapping, described by the *Opto-Electronic Conversion Function* (OECF), also sometimes referred to as *gamma correction*. It also performs some transformations to get a “better looking image” - for example a gain in the colour saturation - that are ignored here. We will assume, for the rest of the chapter, that the final pixel value only depends on the value read by the CCD (and not on the surrounding pixels like in Chapter 6) and that the OECF does not change from picture to picture. This assumption is not correct for high end digital cameras.

5.2.1 Generalities on colour spaces and transformations

The three types of cones that are present in the retina enable the human eye to perceive colour. Each cone type has a particular spectral sensitivity, thus generating three different stimuli when exposed to a light source. The response of the eye to a light stimulus can be characterised with the CIE colour matching functions [Hunt, 1998; Stockman et al., 1993] depicted in Figure 5.3(a) - which are a linear combination of the actual response of the cones



(a) (b)

Figure 5.3: (a) CIE colour-matching functions for a 10 degrees standard observer. These functions are used to compute the response of the eye to an incident light spectrum. (b) Sensitivity of the Nikon D1 digital still camera, measured with a spectro-radiometer. The two sets of curves differ, and span a different spectral sub-space.

in the eye. A digital camera is composed of light sensitive cells, covered with three types of light filters, each having a specific spectral sensitivity. For practical reasons, the spectral sensitivity of the three types of cells in the camera differs from the sensitivities of the cones, mainly because of noise considerations. An example of spectral sensitivity of a digital camera sensor is depicted in Figure 5.3(b). As a consequence, the camera Red, Green and Blue (RGB) values are not the same as the tri-stimuli of the eye if the eye would have been exposed to the same scene. Once the scene has been described in terms of RGB values, some information is lost and it is not possible to exactly know what the cone tri-stimuli for the same scene would be, because, technically, the RGB values of the camera live in another spectral sub-space than the tri-stimuli of the eye. To get from RGB space to a colour space defined in a spectral sense some approximations have to be done, thus leading to a multitude of techniques that are out of the scope of this thesis [Hunt, 1998; Fairchild, 1998; Wandell, 1995]. In the simplest case, a 3×3 matrix is applied to the RGB raw data of the camera. The component of this 3×3 matrix are found by minimizing an error measure between the spectral sensitivities of the target colour space and a linear combination of the spectral sensitivity of the camera. Again, depending on the error criterion chosen, the results will differ. But, since multiplying the RGB values of the camera by a matrix is equivalent to having a camera with a different spectral sensitivity, we will ignore this transformation for the rest of the chapter. If the transformation is more complex, for example if the camera applies a 3×3 matrix after having applied a curve to the image, then the transformation will have a certain influence on the colour mismatch between several pictures, but we will assume that it does not affect the mismatch in a significant way.

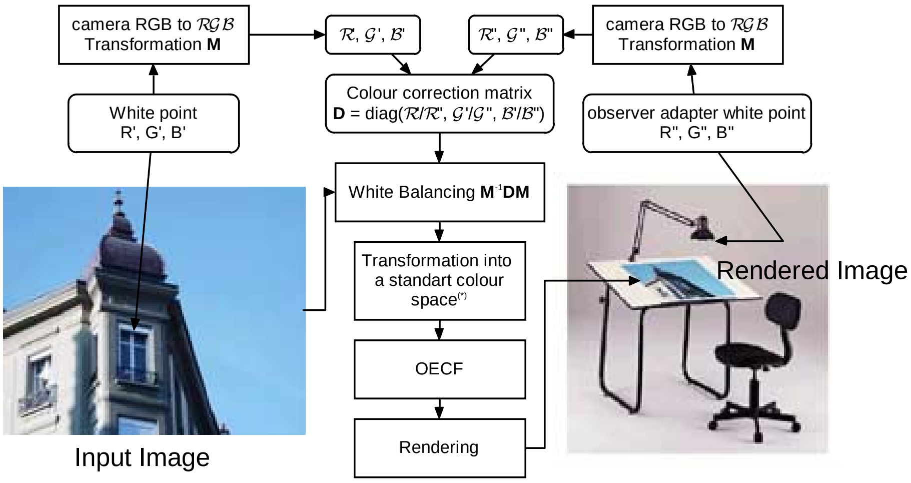

camera RGB to $\mathcal{RGB}$ Transformation **M**
$\mathcal{R}'$, $\mathcal{G}'$, $\mathcal{B}'$
$\mathcal{R}''$, $\mathcal{G}''$, $\mathcal{B}''$
camera RGB to $\mathcal{RGB}$ Transformation **M**
White point R', G', B'
Colour correction matrix **D** = diag($\mathcal{R}'/\mathcal{R}''$, $\mathcal{G}'/\mathcal{G}''$, $\mathcal{B}'/\mathcal{B}''$)
observer adapter white point R'', G'', B''
White Balancing **M**$^{-1}$**DM**
Transformation into a standart colour space(*)
OECF
Rendering
Input Image
Rendered Image

Figure 5.4: Illustration of the white balancing using the von Kries adaptation model. The white balancing is performed by multiplying the colours of the image by a diagonal matrix, which depends on the colour of the light that illuminates the scene and on the colour of the light used to view the image (the observer adapted white point). (*) Note that we assumed that the white point of the standard colour space is the same as the output white point.

5.2.2 White balancing

The colour of an object depends on its spectral reflectance properties, on the lighting conditions and on the local state of adaptation of the human observer. The adaptation of the human observer is a complex mechanism, but for our purpose, the only transformation that we are considering is the *white balancing* [Fairchild, 1998].

The purpose of white balancing can be explained by an example: a white piece of paper appears to be white under sunlight as well as under incandescent light. From a physical point of view, the power spectrum coming from the white piece of paper is equal to the spectrum of the illuminant[2], and differs in the two situations. Nevertheless, it will appear white to an observer, because of *chromatic adaptation*. Chromatic adaptation is the ability of the human visual system to discount the colour of the illuminant, and to approximately preserve the appearance of an object. If the white balancing is done properly in the camera, the colour of the white piece of paper (i.e. the RGB values) will be the same when it is taken under sunlight or under incandescent light[3]. Most of the current cameras apply the Von Kries adaptation model [Von Kries, 1902][Hunt, 1998, p. 125], schematised in Figure 5.4. Given the $\mathcal{R}'_w\mathcal{G}'_w\mathcal{B}'_w$ values of the light source, and $\mathcal{R}''_w\mathcal{G}''_w\mathcal{B}''_w$ the observer adapted white point, expressed in a given colour space, a gain factor is applied to the three colour channels independently, while the image is still in linear space. In other words, $\mathcal{R}''_w\mathcal{G}''_w\mathcal{B}''_w$ is

[2] We assume that the white piece of paper is a perfect reflector

[3] Here, we are not considering incomplete adaptation.

the ”colour” of the light used to view the image and the transformation is applied to the *raw data*:

$$[R, G, B]^T \mapsto \mathbf{MDM}^{-1} \cdot [R, G, B]^T$$
$$\mathbf{D} \triangleq \operatorname{diag}\left(\frac{\mathcal{R}''_w}{\mathcal{R}'_w}, \frac{\mathcal{G}''_w}{\mathcal{G}'_w}, \frac{\mathcal{B}''_w}{\mathcal{B}'_w}\right).$$

In practice the green channel is often left unchanged ($\frac{\mathcal{G}''_w}{\mathcal{G}'_w} = 1$). Several chromatic adaptation of this kind are used in practice and differ in the choice of the colour space where the adaptation is performed. In other words, the chromatic adaptation transforms differ in the choice of the matrix $\mathbf{M}$. In current digital cameras, we found out that the white balancing is performed directly in RGB space, i.e. the matrix $\mathbf{M}$ is the identity matrix. For the rest of the chapter, we will assume that the chromatic adaptation is performed in camera RGB space, thus

$$\mathbf{M} = \mathbf{I}.$$

For a comparison of different chromatic adaptation models, see [Süsstrunk et al., 2001].

5.2.3 Black point estimation

The camera is able to output values of 0 for some very dark points. Nevertheless, none of the sensor cells output a value of 0. This is first due to the presence of dark current in a CCD that generates a non-zero value even if there is no light at all. In addition to that, the darkest point in an image might generate a substantial amount of light that in turn generates a positive output of the CCD cell. The dark current value has to be subtracted from any CCD output. Additionally, we will assume that the camera is able to adapt the value it subtracts from every pixel depending on the image content. This ability allows the camera to capture the whole dynamic range of the image, and also allows the system to handle a certain amount of flare.

Flare is generated by the scattering of light in the image optics [Kingslake, 1983]. This will, in a first approximation, add a constant value to every pixel of the image. To get rid of the flare, it would be sufficient to subtract this value from every pixel, but in practice it is difficult to know the exact amount of flare, and there is no guarantee that this flare is uniform.

5.2.4 Mapping the pixels

The model used to get a pixel value from the value read by the sensor is as follow: for each colour channel, let E be the illuminance read by the sensor. The camera chooses a black point (E_{kR}, E_{kG}, E_{kB}) and a white point (E_{wR}, E_{wG}, E_{wB}) and normalises the output in-between these two values. Finally, it applies a curve $G^{-1}(\cdot)$ to the data:

$$\begin{bmatrix} E_R \\ E_G \\ E_B \end{bmatrix} \mapsto \begin{bmatrix} G^{-1}\left(\frac{E_R - E_{kR}}{E_{wR} - E_{kR}}\right) \\ G^{-1}\left(\frac{E_G - E_{kG}}{E_{wG} - E_{kG}}\right) \\ G^{-1}\left(\frac{E_B - E_{kB}}{E_{wB} - E_{kB}}\right) \end{bmatrix}. \tag{5.2}$$

$G^{-1}(\cdot)$ is the Opto-Electronic Conversion Function (OECF). The models of the OECFs are given in the next section (5.3).

5.3 Opto-Electronic Conversion Function models

As a convention, we will assume that the camera outputs values ranging from 0 to 1. The Opto-Electronic Conversion Function inverse is designated by the letter $G(\cdot)$ and is also supposed to output values in the range $[0, 1]$. In some cases, if an exposure difference has to be taken into account, the camera system transfer function will be designated by letter $S(\cdot)$, which is equal to $G(\cdot)$ up to a linear transformation.

This section is about Opto-Electronic Conversion Functions, the functions that relate the digital camera sensor illuminance to the pixel value of the output image. The term *Opto-Electronic Conversion Function* is reserved for digital cameras. The more general term *tone mapping transfer function* is used whenever there is an analog camera involved in the picturing process. This last function also relates the camera sensor illuminance to the final pixel value, but may involve more processes to get the final image, like for example a development and scanning process.

The following sub-sections describe three models of a tone mapping transfer function. These models aim at the same goal, i.e. how to get back the sensor illuminance from the pixel value, but differently. One is for an analog system, and two are for digital cameras. The performance of each model is illustrated in Section 5.6.

5.3.1 The transfer function of a slide and scanner system

Because film photography is still the best way to get high quality pictures at a decent price, we also use scanned diapositives as our input pictures. The film characteristics are given by a function called the *D-logH* curve that relates film density to the log Exposure, as illustrated in Figure 5.5. The scanner illuminates the diapositive and integrates the ratio of light that goes through the diapositive at a given photosite. The scanner is characterised by its Opto-Electronic Conversion Function (*OECF*). The OECF can be characterised using a calibration slide containing patches of known density [ISO16067 WD : 1999, 1999]. The use of a conventional scanner introduces all the problems associated to a digital camera, described in Section 5.2, so that ideally we should use a professional scanner that can output raw data. Nevertheless, in our case, it turns out that a good approximation of the scanner is given by the following formula: Let I be the pixel value normalised such that $I \subset [0, 1]$. We assume that I is related to the lightness L^* of the pixel through the relation $I = \frac{1}{100}(L^*)^{\frac{1}{\gamma}}$, where gamma γ is a parameter that can be set on the scanner. Setting $\gamma = 1$, gives

$$I = \frac{1}{100} \cdot L^*$$

L^* is related to the relative luminance $\frac{Y}{Y_n}$ of the original illuminated by the light source of the scanner through [Hunt, 1998, p. 63]

$$\begin{aligned} L^* &= 116 \cdot \sqrt[3]{\frac{Y}{Y_n}} - 16, \quad \frac{Y}{Y_n} > 0.008856 \\ L^* &= 903.3 \cdot \left(\frac{Y}{Y_n}\right), \quad otherwise. \end{aligned}$$

Then, $\frac{Y}{Y_n}$ is related to the film density D by

$$D = \log_{10}\left(\frac{Y}{Y_n}\right),$$

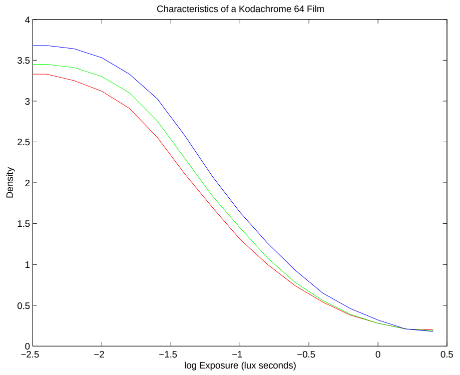


Figure 5.5: *D-logH* curve of the Kodachrome 64 slide film. This curve relates the film density to the illuminance hitting the film surface. This curve can be obtained on the manufacturer Web site.

and finally, the Exposure (i.e. light power) is given by the *D-logH* curve of the film, which enables to build the Slide and Scanner *tone mapping transfer function inverse*

$$R = G(I).$$

R is what we call *raw data*. The function $G(I)$ relates the illuminance entering the camera to the pixel values read in the image I.

5.3.2 An exponential model for the OECF of the camera

Here we consider pictures that are taken with a digital still camera. The digital still camera converts the light falling onto its sensor (CCD) directly into digital values, and is characterised by its Opto-Electronic Conversion Function (OECF) [ISO 14524:1999, 1999]. Nowadays, an OECF is considered as highly confidential data by the camera manufacturers and is not available to the common user. We will assume that the OECF of a camera is similar to the *gamma correction* curve applied in the video standards, such as the ITU-R BT.709 [ITU-R Recommendation BT.709-3, 1998], that has the following form:

$$\begin{array}{ll} I = 1.055 \cdot R^{\frac{1}{2.4}} - 0.055 & \text{if } R > 0.0031308 \\ I = 12.92 \cdot R & \text{else} \end{array}$$

where R is the raw data delivered by the CCD ($R \subset [0, 1]$) and I is the output pixel value. It is an exponential function extended by its tangent (when $R \leq 0.0031308$) to make it pass through zero. In our model, we make the assumption that the OECF of the camera can be described by

$$I = (1 + a) \cdot R^{\frac{1}{b}} - a, \tag{5.3}$$

where a and b are the two unknown parameters of the model. This function is also extended by its tangent to make it pass through the origin, which is a straight forward task to accomplish numerically. We did not express it explicitly in Equation (5.3) because the intersection point between the exponential and its tangent that crosses the origin has no closed form. Since the algorithms needs to compute the raw data pixel value R out of the pixel value I, the function it tries to estimate has the form

$$G(I) = R = \left(\frac{I+a}{1+a}\right)^{b}. \tag{5.4}$$

Even if it is not true for a high end digital camera, we will assume that this function is the same for every picture.

5.3.3 A polynomial model for the OECF of the camera

In order to give more flexibility to the shape of the Opto-Electronic Conversion Function, we adapt a model proposed by Mitsunaga and Nayar [Mitsunaga and Nayar, 1999] to describe the OECF. We will assume that the curve applied by the camera has the following shape

$$G\left(I\right) = c_1 I + c_2 I^2 + ... + c_N I^N \tag{5.5}$$

and the goal is to find the parameters c_i, under the constraint

$$\sum_{d=1}^{N} c_d = 1,$$

that restricts the output in the $[0, 1]$ range ($G(1) = 1$).

The advantage of this model compared to the exponential model of Section 5.3.2 is discussed in the result section 5.6.

5.4 Vignetting

Vignetting is responsible for the image being darker in the borders. On a high quality camera, this phenomenon is mainly due to a geometric effect: the $\cos^4$ effect [Kingslake, 1983, p. 121]. Indeed, if the camera pictures a scene of constant luminance, the illuminance hitting the sensor varies according to the cosine to the fourth power of the viewing angle. Thus,

$$\begin{aligned} E_m &= E_o \cdot \cos^4 \alpha \\ \tan(\alpha) &= \frac{r_{\mathbf{p}}}{f}, \end{aligned} \tag{5.6}$$

where E_m is the illuminance measured by the sensor, $r_{\mathbf{p}}$ the distance of the pixel from the optical centre of the image, f is the focal length of the camera and E_o is the illuminance that would have been measured if the light would have hit the sensor in the optical centre. By computing E_o for each pixel, we get an image without vignetting. Other phenomena related to vignetting, like its dependence on the aperture have been ignored here.

The $\cos^4$ vignetting can be derived using geometrical relations: let a ray of light go through a pin-hole camera, illustrated by the black rectangle in position 1 in Figure 5.6[4].

[4] All the variables used in this section are depicted in Figure 5.6.

Then, the camera is rotated by an angle A and is illustrated by the blue rectangle in position 2. Let D_1 be the (infinitely small) diameter of the pin-hole. Let us consider the camera in position i and let P_i be the energy of the light ray entering the camera. The proportion of the ray that reaches the camera sensor is proportional to the surface of the hole "seen" by the ray, i.e.

$$\frac{P_1}{\frac{\pi}{4} \cdot D_1^2} = \frac{P_2}{\frac{\pi}{4} \cdot D_1 \cdot D_2}.$$

Since $\frac{D_2}{D_1} = \cos(A)$, we get

$$P_2 = P_1 \cdot \cos(A).$$

The illuminance E_i caused by the ray on the sensor in position i is equal to the energy of the ray divided by the surface S_i of the light spot on the sensor:

$$E_i = \frac{P_i}{S_i}.$$

To get the relationship between the pixel values read by the camera in position 1 and 2 (i.e. E_1 versus E_2), we introduce the surface S_3:

$$S_1 = S_3 \cdot \cos(A),$$
$$\frac{S_3}{S_2} = \left(\frac{L_3}{L_2}\right)^2 = \cos^2(A).$$

Finally,

$$\frac{E_2}{E_1} = \frac{P_2}{P_1} \cdot \frac{S_1}{S_2} = \cos(A) \cdot \left[\cos^2(A) \cdot \cos(A)\right] = \cos^4(A).$$

5.5 Estimation methods

5.5.1 State of the art

There has been little work on colour registration in image mosaics, probably because the mosaicking techniques have been developed by the computer vision community who traditionally worked on greyscale values by assuming that the pictures are linear with respect to illuminance. The colour has been used to improve the registration algorithm, for example in [Johnson and Kang, 1997; Guestrin et al., 1998], but without trying to modify it. Colour is also used to identify regions of interest in image sequences to perform replacements, for example in weather news broadcast (chroma-keying), as explained in [Hanna et al., 1999]. We should also mention the work of Majumder et al. [Majumder et al., 2000] who addressed the opposite problem of achieving colour uniformity across multi-projector displays. Closer to our problem is the work of Rushmeier and Bernardini [Rushmeier and Bernardini, 1999] who equalised the colours in a 3D model of a statue, in a controlled environment, by compensating for illumination changes and white point changes, using light spectra techniques. Nevertheless, the most relevant papers for the colour correction problem are the one that compute *radiance maps* from several differently exposed photographs, mostly to get high dynamic range images. Mann and Picard [Mann and Piccard, 1995] (followed by [Mann,

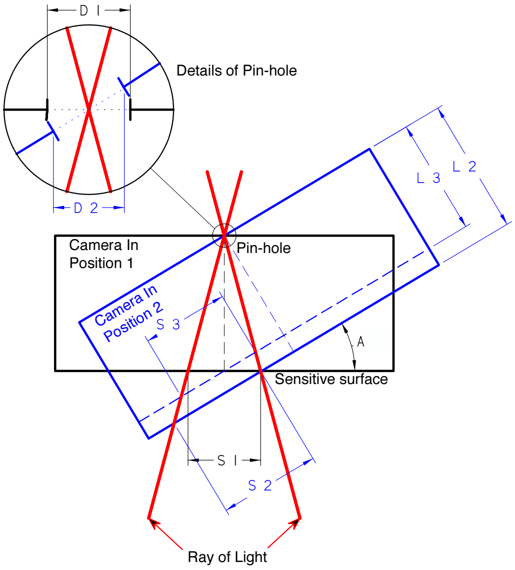


Figure 5.6: Computation of the vignetting phenomenon. The same ray of light is successively captured by a camera in two different positions. The spot of light in the back of the camera in position 2 is bigger than the spot for position 1, thus the value read by the sensor is smaller - i.e. darker - in position 2.

2000]) used an exponential model to enhance the dynamic range of an image. Debevec and Malik [Debevec and Malik, 1997] proposed a method to estimate an OECF of arbitrary shape, from pictures with known exposures. The only constraint needed is smoothness. Robertson et al. [Robertson et al., 1999] modified slightly Devebec's method to give more weight to more reliable pixels. Finally, Mitsunaga and Nayar [Mitsunaga and Nayar, 1999] proposed a polynomial model - similar to the one of Section 5.3.3 - to describe the OECF that can be deduced from a set of differently exposed pictures with only an approximate knowledge of the exposure time.

5.5.2 Avoiding bad pixels

Chapter 4 exposes the various methods used to discount the influence of *outliers*, that is, the elements in an image that might confuse an algorithm. These elements are, for example, the moving objects in the scene. Here, we are interested in knowing how much information is contained in a pixel pair about the lighting conditions, about the camera settings, etc. For example, if a pixel appears saturated in one of the images of the pair (i.e., it has a value of 0 or 255 with an 8 bits per channel camera), this pixel will only give an upper- or lower-bound information about the exposure difference of the image pair. If this pixel is used in the same way than the "good" ones, that is, the pixels that have values in the middle range of the camera, then the estimation will fail. By writing the motion estimation equation (2.26), the reliability of a pixel pair is embodied by the factor W_i, who multiplies each line of the equation system by a different amount. This factor depends on the two values delivered by the camera: for each image j of the pair, in each pixel location $\mathbf{p}_i$, we can compute a weight value $w_{j,i}$ that tends toward 0 in the border of the dynamic range of the camera, and take values of 1 in the middle range of the camera, thus giving less weight to potentially noisy pixels. Since we need one single weight value for each pixel of the image pair, the two weights $w_{0,i}$ and $w_{1,i}$ have to be combined into a single one. The resulting weight for the pixel pair is computed as

$$\frac{1}{W_i} = \frac{1}{w_{0,i}} + \frac{1}{w_{1,i}}. \tag{5.7}$$

The underlying assumption is that $w_{\cdot,i}$ is the inverse of variance of a Gaussian random variable. Indeed, if the pixel value is a random variable distributed as $\mathcal{N}\left(\mu_{\cdot,i}, 1/w_{\cdot,i}\right)$, - where $1/w_{\cdot,i}$ is the *variance* - then the residual is also a Gaussian random variable distributed as $\mathcal{N}\left(\mu_{0,i} - \mu_{1,i}, \frac{1}{w_{0,i}} + \frac{1}{w_{1,i}}\right)$, leading to Equation (5.7).

An example of the function we used is depicted in Figure 5.7. This weight is computed independently for each colour channel. In practice, the weight never reaches 0, but a value of 10^{-5} is used instead in order to avoid rejecting every pixel of the image pair if the pair has very different exposure parameters.

Remark

Although the function of Figure 5.7 seems somewhat arbitrary, an attempt to precisely modelling the noise as the variance of a gaussian random variable has been tried, but did not succeed. The modelling involves the use of the exposure parameters of each picture, and makes the approximation $\frac{\partial W_i}{\partial \theta} \simeq 0$ of Section 2.6.2 questionable. The added complexity of the motion estimation makes the approach computationally more expensive and, unless precise estimates of the noise variance are available, is not worth the effort.

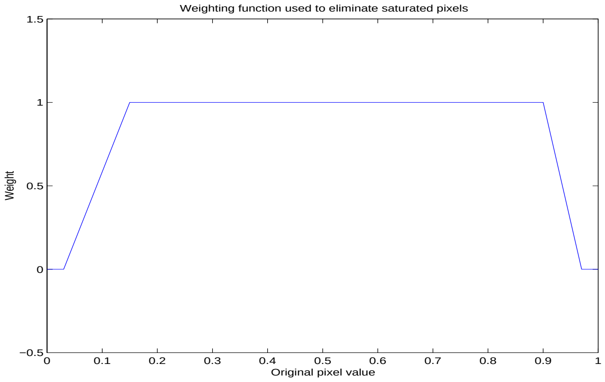


Figure 5.7: Weight values used to get rid of saturated pixels. The influence of each pixel in the registration process is decreased in the border of the image range. Possible pixel value range from 0 to 1.

5.5.3 Detecting mis-alignments

The next section (5.5.4) presents a method to compute the camera OECF from two aligned pictures. In practice the pictures are never perfectly aligned, and to make an OECF estimation robust, the method should be able to discount the mis-aligned pixels in the image. The camera OECF and the camera settings, do not affect in a significant way the relationship between a pixel and his neighbours, except for the difference in exposure. In other terms, the difference in colours between the two images is an information of the low-pass kind and can be annihilated by passing a high-pass filter on the images[5]. The two filtered images should be equal where they are well aligned. If there is a slight mis-alignment, then the images differ only if there is high frequency information at that location (for example an edge). On the other hand, the higher the local frequency of the image, the better the alignment has to be to compare the colour information. The comparison of the high-pass images provide a good way to determine what part of the image is a good candidate for a pixel-wise comparison.

In practice the images are filtered with the derivative of a gaussian filter (with $\sigma = 1.25$ pixels) and normalised such that the variance of the images in the overlap area are equal. The alignment weight is computed as

$$W_i^{(a)} = \frac{1}{|f(I_0(\mathbf{p})) - f'(I_1(\mathbf{p}'))| + Q},$$

where $f(I_0(\mathbf{p}))$ is the filtered image I_0 and $f'(I_1(\mathbf{p}'))$ is the filtered image I_1 with normalised variance. Q is a small positive number equal to the quantisation step of the image ($Q = 1/255$ for an 8 bit image). The quantisation step is used so that the weights used for two arbitrarily close error values - which might be separated by an amount Q - are not too dissimilar. The mis-alignment weight is combined to the one used in Section 5.5.2, using the same approach,

[5] Strictly speaking, we should not talk about low- and high-pass information since there are some non-linearities involved.

that is, (by borrowing the notations of Section 5.5.2)

$$\frac{1}{W_i} = \frac{1}{W_i^{(a)}} + \frac{1}{w_{0,i}} + \frac{1}{w_{1,i}}.$$

We want to emphasise that this weight is only used to estimate the OECF in Section 5.5.4, and is not used in the context of Section 2.6.2. The reason is that, in the last case, the algorithm needs the mis-aligned pixels to get a better alignment.

Another question that appears now is: Why not use the high-pass images to perform the motion estimation and do the colour correction once the pictures are aligned? The reason we did not proceed in that way is related to the shape of the error surface. The motion algorithm performs a gradient descent that, given an error surface (in *n*-dimensional space), finds a path that follows the surface downwards. If the surface is irregular, and contains local minima, the algorithm is likely to fail. By filtering the image with a high-pass filter, the error surface gets also high-pass filtered and becomes irregular, consequently, the system requires to have a better initial estimate to converge to the right solution (see Section 2.8 for the details). This is illustrated in Figure 5.8 where the error surface is shown for a translational model, for colour matched images and for filtered images.

5.5.4 Estimating a polynomial OECF

The polynomial model of Equation (5.5) requires a different estimation technique than the one described in Section 2.6.2. We adapt the model proposed by Mitsunaga and Nayar [Mitsunaga and Nayar, 1999]. The idea is to make iterations between motion estimation, and OECF estimation. In the easiest case, we can consider that the images are already aligned. Let $G_j(\mathbf{p})$ be the illuminance that falls on the camera sensor at position $\mathbf{p}$ while taking picture j. From (5.5), we have $G_j(\mathbf{p}) = \sum_{d=1}^{N} c_d I_j(\mathbf{p})^d$. Now, we can express the ratio T_{01} of the exposure settings between image I_0 and image I_1 with the following relation:

$$\frac{G_0(\mathbf{p})}{G_1(\mathbf{p}')} = T_{01}, \tag{5.8}$$

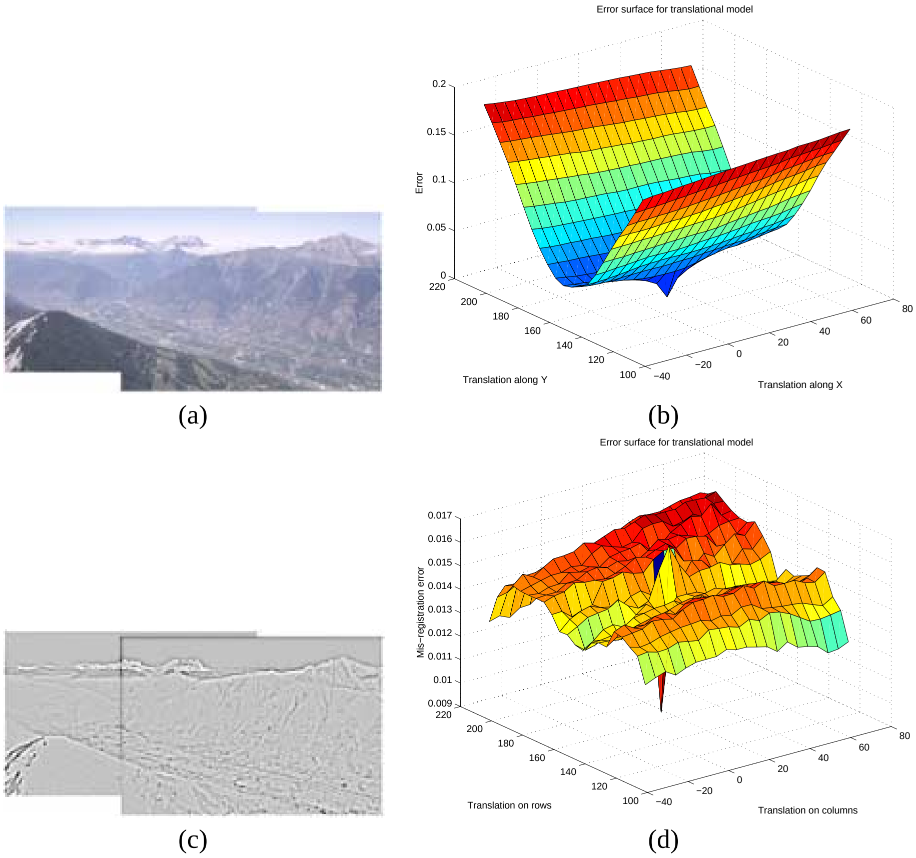
where $\mathbf{p}$ and $\mathbf{p}'$ are the matching positions in the two images (an object of the scene that appears in image I_0 at location $\mathbf{p}$ appears at location $\mathbf{p}'$ in image I_1). Now, if the camera moves between the two pictures (i.e. $\mathbf{p} \neq \mathbf{p}'$), Equation (5.8) has to take the vignetting phenomenon into account, and becomes

$$\frac{V(\mathbf{p}) \cdot G_0(\mathbf{p})}{V(\mathbf{p}') \cdot G_1(\mathbf{p}')} = T_{01},$$

where $V(\mathbf{p})$ is the vignetting correction factor defined in Section 5.4 ($V(\mathbf{p}) = \cos^{-4}\alpha$). Furthermore, to fit the model described in Section 5.2.4, the system has to take the flare (or the black point compensation) into account, hence

$$\frac{V(\mathbf{p}) \cdot G_0(\mathbf{p}) + K_{01}}{V(\mathbf{p}') \cdot G_1(\mathbf{p}')} = T_{01}.$$

The factor K_{01} accounts for the flare difference in the two image, but does not explicitly express the flare difference. The variables K_{01} and T_{01} are related to the variables of Section



(a) (b)

(c) (d)

Figure 5.8: De-regularising the error surface by filtering. (a) Shows an image superposition and (b) the associated error surface for a translational motion model. (c) Shows the same image superposition, but using filtered images, and (d) shows the associated error surface. We can notice that by filtering the images, the error surface becomes irregular and makes a motion estimation algorithm more sensitive to initial conditions.

5.2.4, through

$$K_{01} = \frac{E_{k,0} - E_{k,1}}{E_{w,0} - E_{k,0}},$$
$$T_{01} = \frac{E_{w,1} - E_{k,1}}{E_{w,0} - E_{k,0}},$$

for each colour channel independently. $E_{\cdot,i}$ designate the illuminance variable of Section 5.2.4 for one of the colour channels associated to image i.

The estimation of the OECF inverse (i.e. parameters $\{c_1, ..., c_N\}$) and the exposure parameters (K_{01} and T_{01}) is done in an iterative way. The OECF is found by computing the mis-registration h between the images in the overlap area

$$h(c_1, ..., c_N) = \frac{1}{n}\sum_{i=1}^{n} W_i \cdot \|r_i[c_1, ..., c_N]\|^2 ,$$
$$r_i(c_1, ..., c_N) \triangleq \{V(\mathbf{p}_i) \cdot G_0(\mathbf{p}_i) + K_{01} - T_{01} \cdot V(\mathbf{p}'_i) \cdot G_1(\mathbf{p}'_i)\} \cdot W_i,$$

and set its derivative to zero:

$$\frac{\partial h([c_1, ..., c_N]^T)}{\partial [c_1, ..., c_N]^T} = \mathbf{0}. \tag{5.9}$$

The factor W_i allows to put more weight on reliable pixels, and is computed according to Section 5.5.3. There are n pixels in the overlap area $\{\mathbf{p}_i, \mathbf{p}'_i\}$. The system of equation in (5.9) is a linear system that is solved under the constraint that function $G(\cdot)$ outputs values in the range $[0, 1]$, i.e.

$$\sum_{d=1}^{N} c_d = 1.$$

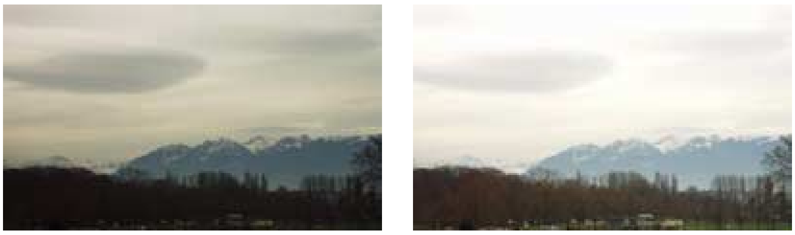
Once the OECF is found, the factors K_{01} and T_{01} are found using either a linear regression, or during a motion estimation iteration described in Chapter 2.

5.6 Results

In this section, the colour correction method are evaluated on image pairs. The image pair is blended using a checkerboard-like technique: the overlap area can be considered as a checkerboard; the black cells of the checkerboard contain the pixels from image I_0 and the white cells of the checkerboard contain the pixels from image I_1. In terms of quality, the checkerboard-like blending is about the worst that can be done to render the final mosaic and is solely aimed at emphasising the mismatch in the registration.

5.6.1 Stitching images from a digital still camera

The images are taken with a digital still camera. The camera Opto-electronic conversion function (OECF) is supposed to follow the polynomial model of Section 5.3.3 with 5 coefficients: $G(I) = a_1 I + a_2 I^2 + a_3 I^3 + a_4 I^4 + a_5 I^5$. The characteristics of the camera (the OECF inverse) has been computed using the two images of Figure 5.9. The result is shown in Figure 5.10.



(a) (b)

Figure 5.9: Images used to fit a polynomial OECF model of the camera. The images have the same white point but different exposures.

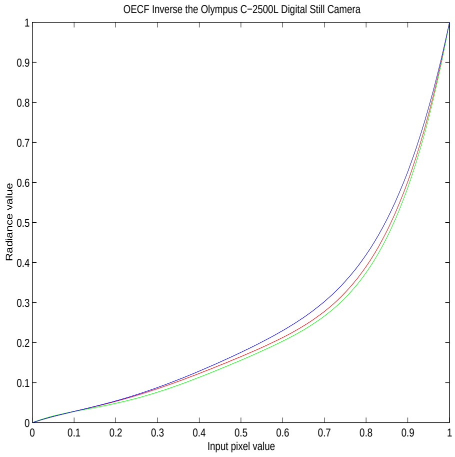


Figure 5.10: Opto-Electronic Conversion Function Inverse of the olympus C-2500L camera obtained by fitting a polynomial model of order 5 to the images of Figure 5.9.

The first example shows an old building, taken with the camera set on automatic mode. The camera used a shutter speed of $1/60$ sec for the lower building and a speed of $1/200$ sec for the top. The aperture was kept constant, and the white point changed. Figure 5.11(a) shows the mosaic of this building without any colour correction. Figure 5.11(b) shows a correction that assumes a fixed white point and a varying exposure. Figure 5.11(c) assumes a varying white point and 5.11(d) corrects for the flare or the mismatch in the black point. The images were aligned using a rotational model with lens distortion correction, and the alignment parameters used to render these images are the ones found by computing image 5.11(c). It is worth mentioning that the colour correction improves the alignment of the pictures, as illustrated in Figure 5.12, where one picture has been computed by assuming a fixed white point and the other by assuming a varying white point. We should also mention that a multi-resolution pyramid has been used and the complexity of the model is adapted to the resolution. This means, for example, that to get the image of Figure 5.11(d), we first assume a fixed white point, and do not adapt the focal length nor the distortion parameter at the lowest resolution. Then, while the resolution of the pictures increases, we allow the white point, the focal length and the distortion to vary. Finally, at the lowest level, we allow the black point to be adapted. If we would have adapted every parameter from the start, the system would have landed in a local minimum, and the result would not have been as good.

Another mosaic estimated with the exact same procedure is depicted in Figure 5.13. The focal length used is about half as big (i.e. 45mm versus 110mm), making the alignment more tricky. Aligning pictures taken with a wide angle lens is more difficult because of the increased distortion of the image, especially when using a lens with a zoom, and because of the greater impact that a bad alignment of the optical centre can have on the image. Figure 5.13(a) shows the mosaic of the original pictures, without any colour adjustments, 5.13(b) shows the mosaic adjusted with the nominal shutter speed given by the camera, 5.13(c) shows the white point corrected mosaic, and 5.13(d) shows the black and white point corrected image. In the original pictures, shown in Figure 5.14, there is a big difference in the white point. The mosaic has been rendered with the motion parameters of Figure 5.13(c). The mosaic shows a gradual improvement with the complexity of the model, except at the end: this time, the black point adjustment did not improve the final result, and in some sense, the result is worse when trying to adapt for the black point.

In the third example, depicted in Figure 5.15, the OECF has been estimated on the picture overlap. Here, the method is quiet sensitive to the order of estimation: The images have first been aligned by assuming a certain shape of the OECF (the sRGB shape presented in Section 5.3.2), and then an iteration between OECF and image alignment has been performed. The result is shown in Figure 5.15(c). For the sake of comparison, Figure 5.15(d) contains the mosaic processed like in the former two examples. Figure 5.15(b) shows the result of using the exponential OECF of Section 5.3.2. There, the result is not as good as in 5.15(c), but the method is less sensitive to the order of estimation. We should also mention that by only working on image pairs and not globally on a whole mosaic, it is not possible to estimate the exposure parameters and the polynomial OECF simultaneously. At least one of the two has to be fixed, because a slight change in the OECF can compensate for a shutter speed change, given that noise is present in the images. If the white point changes between the image, then one has to estimate the OECF using the achromatic channel and estimate the white point like in the former example. If the change in white point is not important, it will be compensated by the shape of the OECF. When working with more than two pictures at a time, [Mitsunaga and Nayar, 1999] showed that on still images (where there is no camera motion involved), only

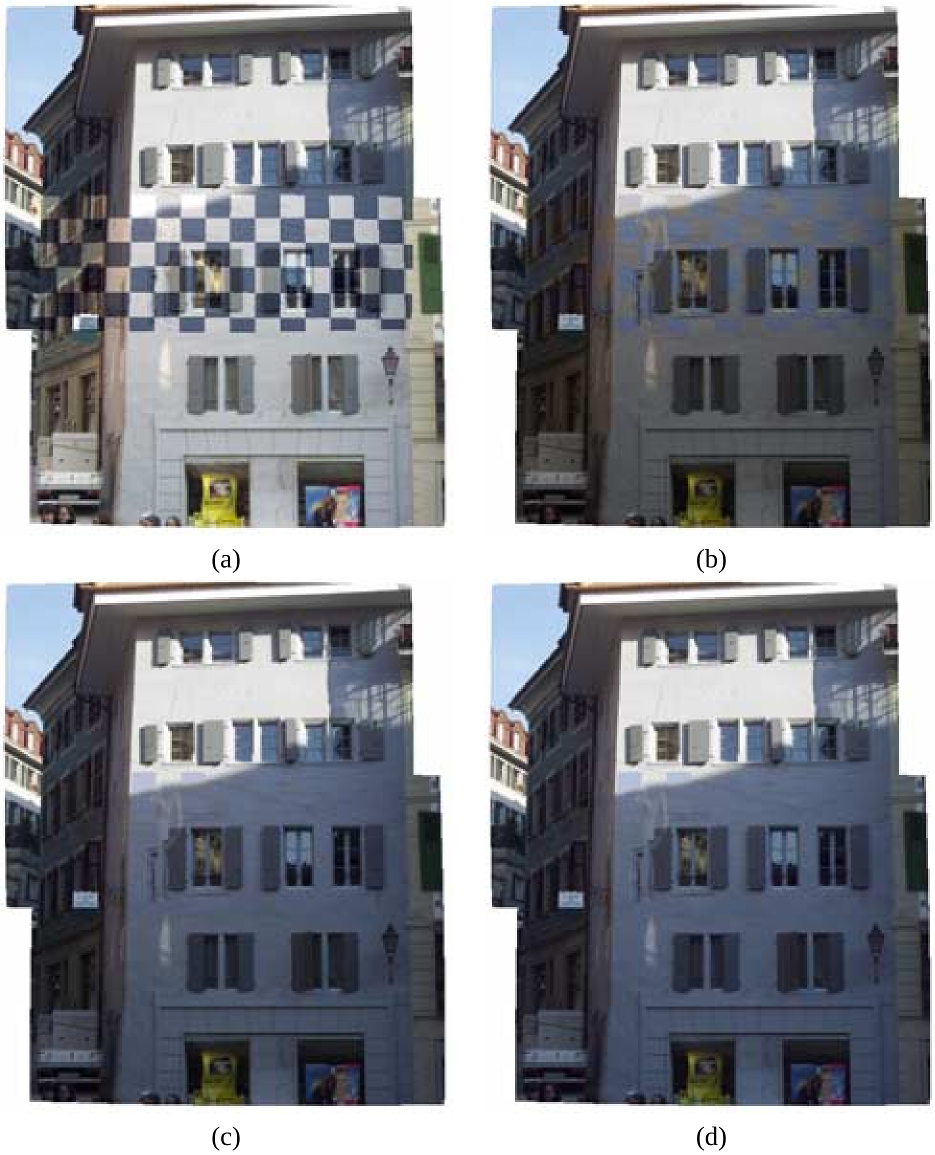
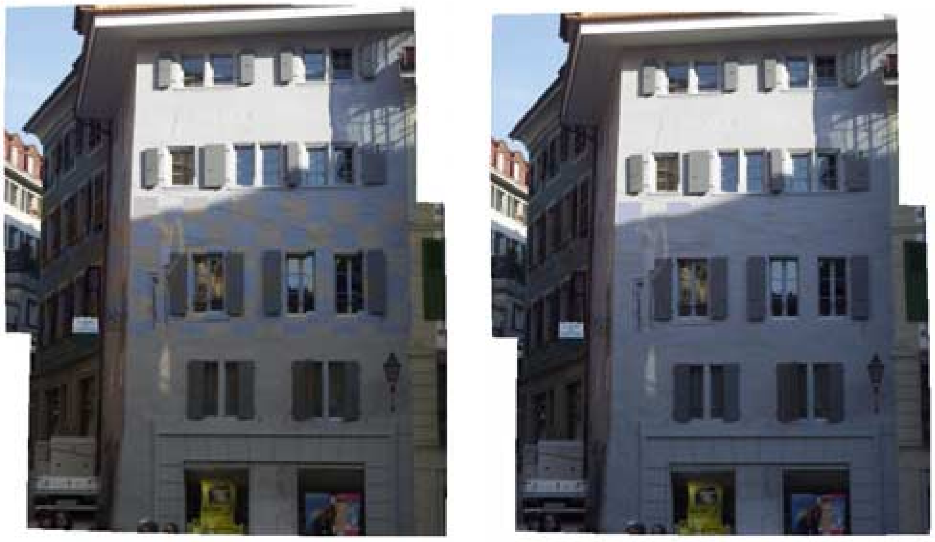


Figure 5.11: Illustration of the colour correction process. (a) original mosaic without correction. (b) Shutter speed compensation (b) shutter speed and white point compensation (d) shutter speed, white point and black point (or flare) compensation. The mosaic shows a gradual improvement of quality (although the difference between (c) and (d) may not appear if the quality of the print is not good enough).



(a) (b)

Figure 5.12: Improving the alignment using colour correction. (a) shows the alignment of the picture for a fixed white point assumption; notice the mis-alignment of the window in the middle-right. (b) Alignment using a varying white point. This example illustrates that the colour correction helps the motion estimation algorithm on the ill-posed problem of the focal length and the lens distortion parameter estimation.

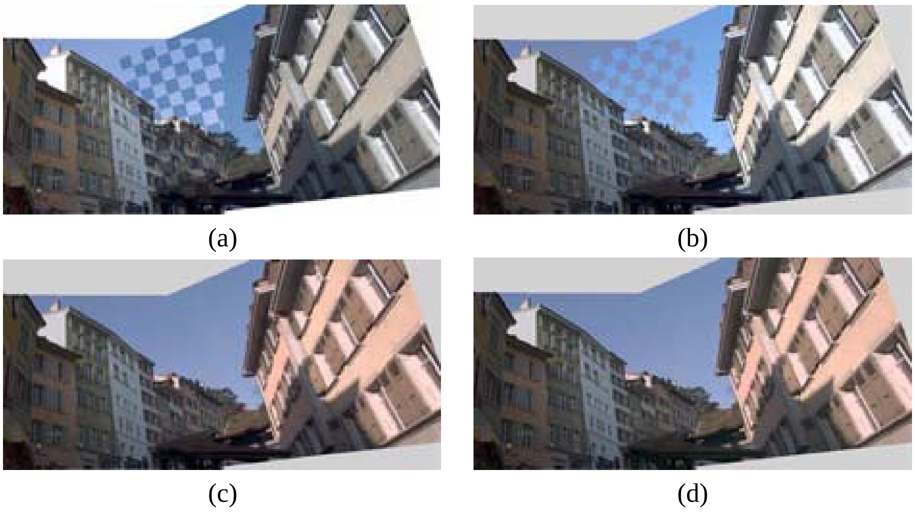


Figure 5.13: Illustration of the colour correction process. (a) original mosaic without correction. (b) Shutter speed compensation with parameters given by the camera (b) shutter speed and white point compensation (d) shutter speed, white point and black point (or flare) compensation. The mosaic shows a gradual improvement of quality until (c) (although the difference between (c) and (d) may not appear if the quality of the print is not good enough).

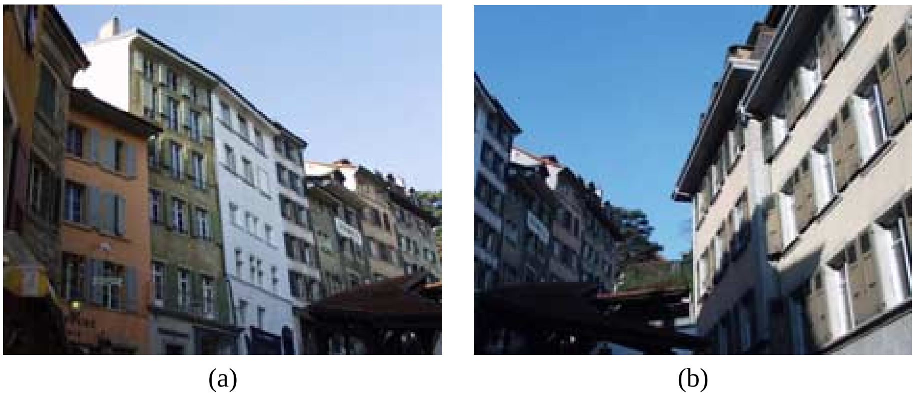
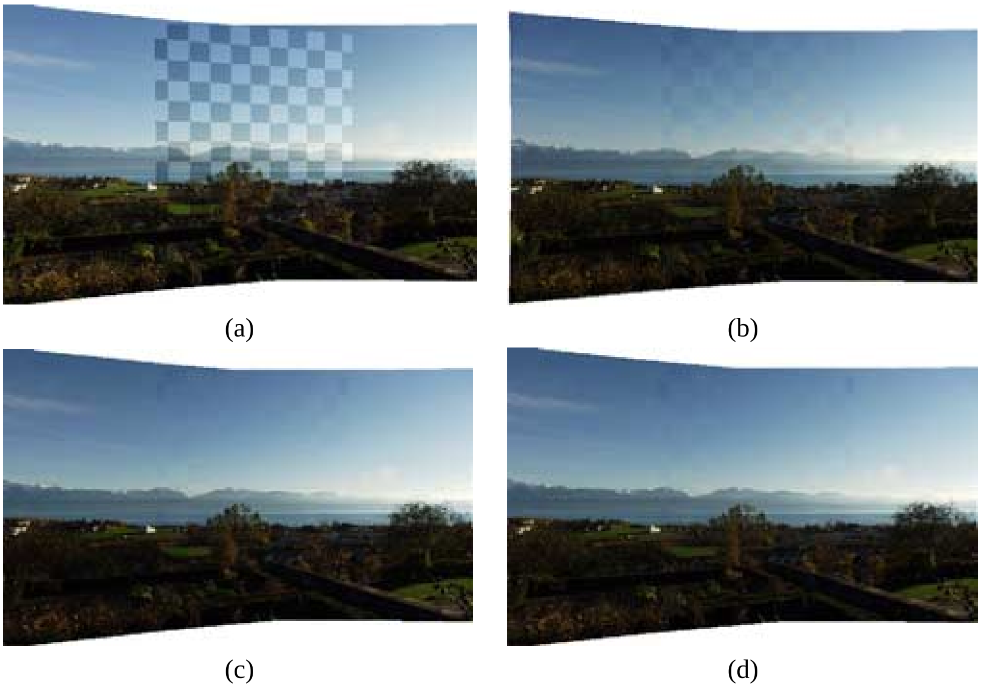


Figure 5.14: original pictures used to build the mosaic of Figure 5.13.



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

Figure 5.15: Comparison of OECF models and estimation techniques. (a) Original - not corrected. (b) White point estimation with exponential OECF. (c) polynomial OECF estimated on image overlap. (d) White point estimation with polynomial OECF (estimated off-line).

an approximate knowledge of the exposure parameter is necessary to get back the OECF. To apply the algorithm in a global way, we would need to use the technique described in Section 3.1.3, but unfortunately, we did not have the time to test this hypothesis on a real mosaic, because of the heavy implementation cost that such an algorithm requires.

The exponential model of Section 5.3.2 does not perform as well as the polynomial model. Nevertheless, its behaviour is more predictable, in the sense that it never goes totally out of control. Again, if the exposure settings of the camera is unknown, then it is advisable to keep the gain parameter a in (5.3) constant and estimate only the exponential term b along with the exposure parameter of the images.

The polynomial OECF model, and its estimation method has the particularity of either working quite well or failing totally. An example of failed estimation is given in figures 5.16(b) and 5.17(c). In the example we tested, when the estimation failed, it delivered a non monotonic curve. This makes a failure easy to detect and in the case when only one or two curve failed (out of the three colour channels) it is possible to use the curve that succeeded and apply it to the other channels. This has been done to compute Figure 5.16(c) where the red curve has been computed from the blue and green ones. The curves found for 5.16(b) are depicted in Figure 5.16(d). In the case of Figure 5.17(c), since every curve failed, this method cannot be applied. By carefully examining Figure 5.17(b), we can see that, since the sky matches more or less and the cathedral colour is in total mismatch, there was an illumination change in the scene that occurred between the two pictures, which violates one of our basic assumptions (to be more specific, $E(\lambda)$ in Equation (5.1) is not the same in both pictures). In other words, the light in the scene changed, and caused a colour change, which is strictly speaking not reversible. To handle this kind of situation in an exact fashion, we would need to know the spectral content of the image for each pixel, and not just an RGB value.

5.6.2 Stitching images from slides

In the first example, we use two pictures that have been scanned from slides. The slides are on Kodachrome 64 film, whose characteristics can be found on the manufacturer Web site. The slide characteristics is given by the D-LogH curve of Figure 5.5, which relates the film density to the log-exposure. Then, the slide is passed through a scanner whose OECF has been computed by the characterisation process of Section 5.3.1. The result of the mosaicking is shown in Figure 5.18. The picture have been taken using a fixed exposure and focal length. Figure 5.18(a) shows the original mosaic without any colour correction. The pixel values are the one originally delivered by the scanner. This results shows what any standard mosaicking algorithm would do. Figure 5.18(b) shows the mosaic processed with colour correction, but without taking vignetting into account. The estimated photometric parameters are the exposure difference between the images, measured independently in the Red, Green and Blue channel. This accounts for the difference in white balancing and gain that may occur in the scanner. In this example, it is not clear to us whether the change in exposure occurred in the scanner or because of the camera shutter imperfections. Figure 5.18(c) shows the final mosaic, with vignetting correction. The only difference between 5.18(b) and 5.18(c) is the vignetting correction.

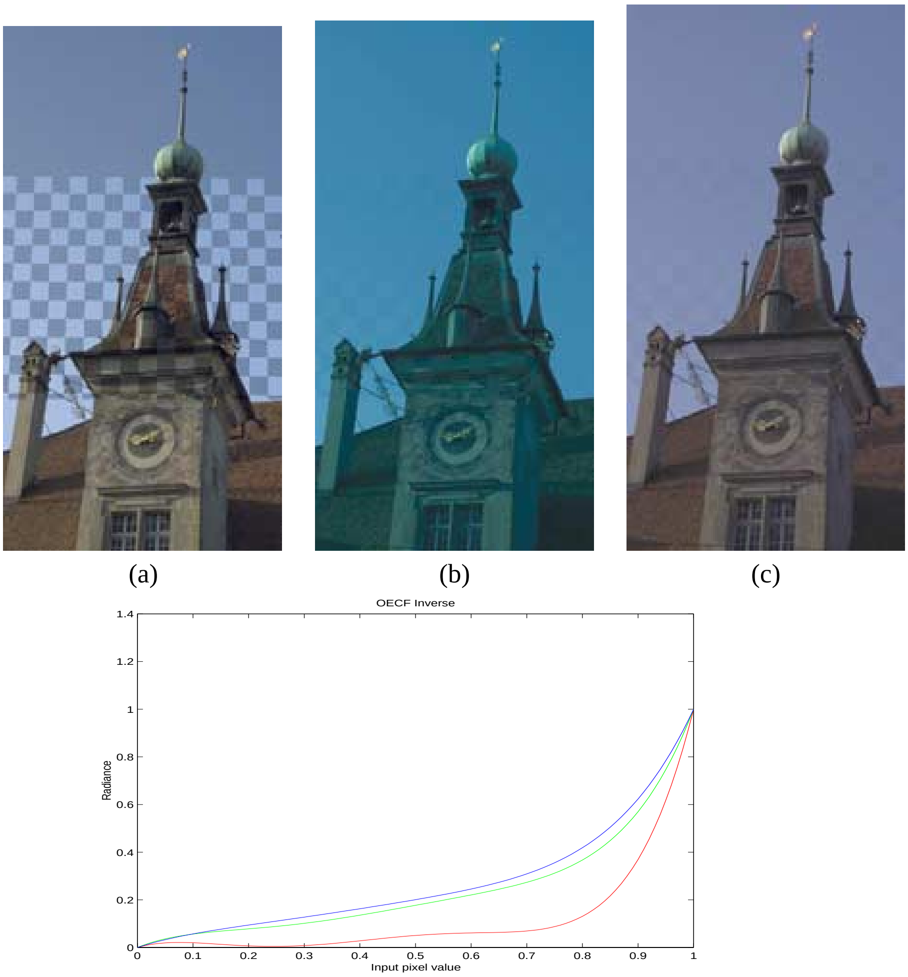
5.7 Conclusions

This chapter presented a set of technique that enable to recover the raw data out of a picture, i.e. a radiance map of the scene, and render the final picture according to the radiance of the scene.

We want to emphasise that this chapter presents some examples of parametric models that are used to retrieve the raw data from the pictures. The important matter are not the models themselves, but rather the approach that is used. Ideally, a good colour correction algorithm should use all the information available from the camera manufacturer, and estimate only the parameters that may change between the pictures.

Among the different techniques used to correct the colours, we preferred the one that uses a set of still pictures to compute the camera characteristics, like in Figure 5.11. This technique gave very satisfactory results, and did not require as much effort as the example using slides in Figure 5.18. Estimating an Opto-Electronic conversion function (OECF) on an image overlap using a polynomial model can be unpredictable, as shown in Figure 5.17. The exponential model of the OECF did not give very good results, but still improved the quality of the mosaic. Nevertheless, in the example we tested, it always gave an improvement compared to the uncorrected mosaic, and has a more predictable behaviour.

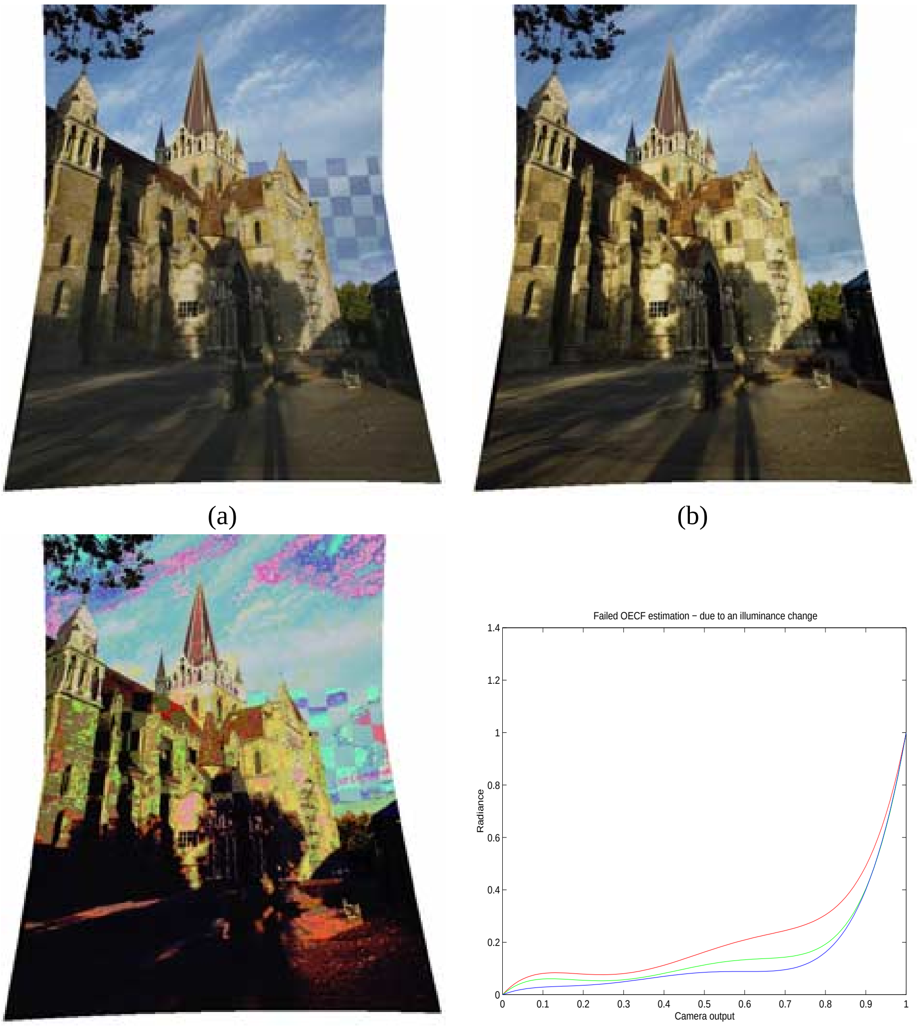
Chapter 4 that deals with outliers, can be combined with this chapter by applying the outlier detection techniques to the *raw data* output by the algorithms outlined in this chapter.



(a) (b) (c)

(d)

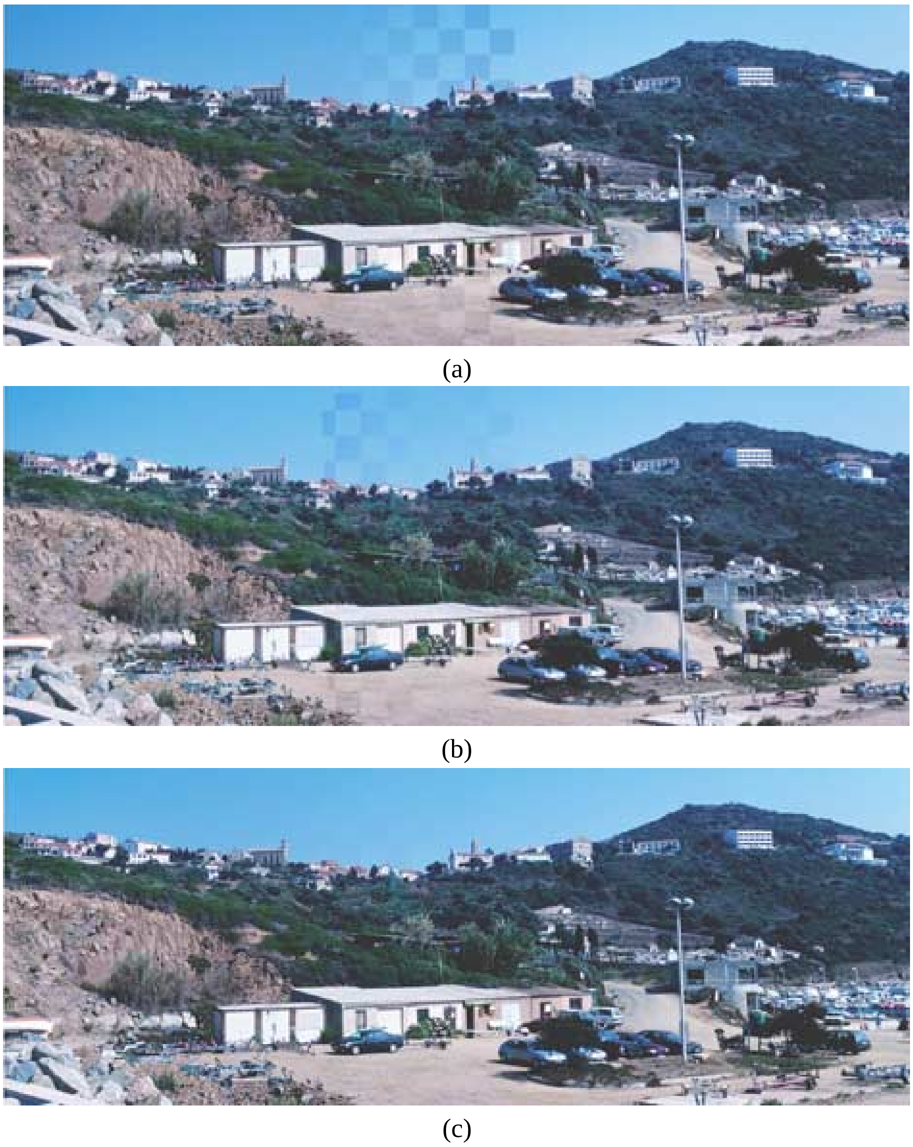
Figure 5.16: Simultaneous OECF and motion estimation. The OECF is estimated on the image overlap. (a) Original mosaic. (b) Correction with OECF estimation - the estimation of the red curve failed. (c) The red curve is computed from the blue and green ones. (d) OECF plot of image (b).



(a) (b)

(c) (d)

Figure 5.17: Simultaneous OECF and motion estimation. The OECF is estimated on the image overlap. (a) Original mosaic. (b) Exposure Compensation - a change in the lighting of the scene occurred: the sky matches in a region right next to a huge mismatch on the wall of the church. (c) Failed correction with OECF estimation (d) OECF plot of image (c).



(a)

(b)

(c)

Figure 5.18: Colour correction using slides with prior calibration. (a) Original mosaic (b) Exposure correction (c) Exposure and vignetting correction. The figure shows a step-wise improvement with the model complexity.

Chapter 6

On the Rendering of High Dynamic Range Images

This chapter presents a new algorithm for rendering high dynamic range images, captured with a digital still camera. It is image dependent, takes spatial information into account, and tries to maintain local perceived contrast as best as possible. If the dynamic range of the starting image is much to high for the capabilities of the output device, a compression of the perceived contrast will be performed. The calculation of the perceived contrast at each position within the image is based on a complex model of the human visual system taking the contrast sensitivity function as well as the contrast discrimination function into account. Naturally, those functions depend on luminance, frequency, and viewing conditions. The algorithm has been applied on raw images (12 bits per colour channel) captured with a professional camera and produces 8 bits per channel images, displayable on a conventional monitor.

6.1 Introduction

Examining a natural outdoor scene on a bright sunny day, one can easily observe huge luminance differences between areas in the shadow and areas in bright sunlight. The ratios can easily be in a range of 1 to 1'000. If we want to capture an image of that scene with a digital still camera and display it on a monitor, we will observe that no matter which camera settings we choose we will either loose the details in the dark area or the details in the bright areas.

In order to overcome that problem, two issues have to be considered: the first one is the capturing of the high dynamic range scene. The raw data, which is captured by the CCD and delivered by the A/D converter, consists currently of 12 to 14 bits per colour channel, which is certainly not enough. The second problem is the rendering of a high dynamic range image, i.e. the transformation of a 14 bit per channel image to an 8 bit per channel image, which is a common number for the final output of a digital still camera. The first problem can either be solved through the combination of several images of different exposures or via improved CCD's. It is an interesting problem, but it will only briefly be discussed in Section 6.2.

The core of this chapter deals with the second problem, how to transform a broad range of luminance values present in a high dynamic range scene into luminance values displayable

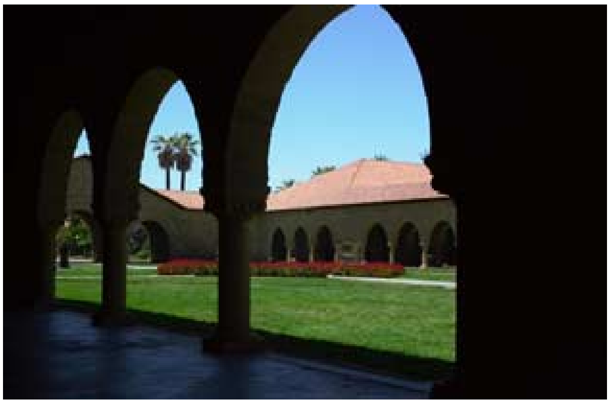
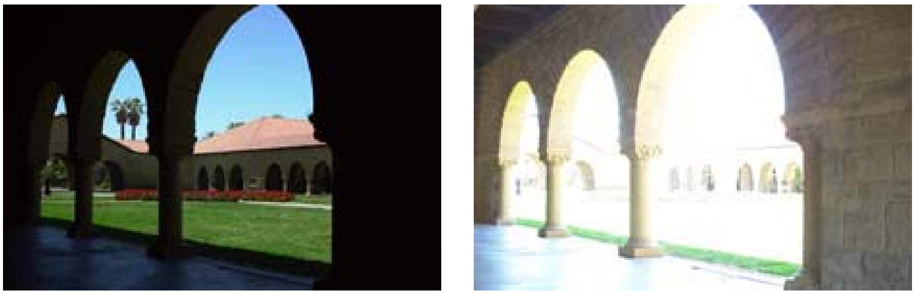


Figure 6.1: Image of a high dynamic range scene captured and rendered with traditional techniques. The details under the arches are totally lost whereas the observer who took the picture was able to see them.

at a monitor. The goal of this transformation is to compress the dynamic range while maintaining the perceived similarity between the original scene and the corresponding image, as best as possible. Sections 6.4 to 6.8 will discuss features of the human visual system that will be used in our new rendering algorithm, proposed and extensively discussed in Section 6.9. Unlike other approaches, our algorithm uses a fairly complex model of the human visual system, which ensures accuracy, robustness and extendibility. Images demonstrate the results of our approach.

6.2 Acquisition of high dynamic range images

Current digital cameras have a poor dynamic range capturing capability. To get high dynamic range pictures, several differently exposed images can be merged together into a single high dynamic range *radiance image*. An example of differently exposed images is given in Figure 6.2. To merge the images together, one has to use the exposure parameters of each image and the camera Opto-Electronic Conversion Function (OECF), i.e. the function that relates the scene illuminance hitting the sensor to the pixel value (see Chapter 5). Methods for computing a high dynamic range image have been proposed by Mitsunaga [Mitsunaga and Nayar, 1999], Debevec [Debevec and Malik, 1997], Robertson [Robertson et al., 1999] and Mann [Mann and Piccard, 1995; Mann, 2000]. Additionally, The ISO standard [ISO 14524:1999, 1999] contains all the information needed to measure an OECF with laboratory equipment. Once a high dynamic range image has been computed, a rendering of that image has to be performed taking the contrast ratio of the individual devices into consideration. In order to separate the generation of a high dynamic range image from the rendering step, a professional camera, which has the capability to output 12 bits raw data pixels was used in our experiments. The use of raw data, i.e. pixel values that are proportional to the scene radiance, avoids most of the artifacts introduced by an OECF. Furthermore, we will assume that the images output by the camera are flare corrected.



(a) (b)

Figure 6.2: Input images to the acquisition of high dynamic images algorithm. (a) Contains the details in the bright region, whereas (b) contains the details in the dark areas. Merging the two images together results in a high dynamic range image.

6.3 State of the art

There are two major trends in the research for dynamic range compression: The first tries to mimic the behaviour of the retina at a very early stage in the vision process and is based on the *retinex theory*, a technique developed by Land and McCann [Land and McCann, 1971] (see McCann [Funt et al., 2000] for implementation details). The second trend uses the properties of the human visual system at a higher stage, by means of the psycho-physical measurement on contrast sensitivity. The more traditional techniques use image dependent lightness rescaling functions, like in Braun [Braun and Fairchild, 1999] or make these function spatially varying, like in Moroney [Moroney, 2000] or in Kobayashi [Kobayashi and Kato, 1999]. Traditional techniques are problematic if they are applied to images with high contrast levels. Other techniques used in computer graphics increase the perceived dynamic range of an image by reproducing the effect of scattering in the eye lens that occurs when there is an strong source of light in the scene [Spencer et al., 1995]. The emphasis of this chapter is on the use of the contrast sensitivity functions of the human visual system to render high dynamic range scenes.

Tumbling [Tumbling et al., 1997] proposed a method to render a scene which varies according to where the observer is supposed to look in the picture. Pattanaik [Pattanaik et al., 1998] and Ferwerda [Ferwerda et al., 1996] proposed an interesting method that applies the contrast sensitivity function of the eye to each frequency subband of an image separately. It also models the response of the 2 types of receptors present in our eyes - the rods and the cones - and is able to reproduce the decrease in colour perception with decreasing illuminance. Their method also adapts to illumination conditions by means of an incomplete chromatic adaptation based on a grey world assumption. Nevertheless, it does not specifically address the question of dynamic range compression: it builds a very general model that computes the contrast perception out of a scene, but does specify how to handle the contrast if it results to be too large to be displayed on the output device. More specifically, the algorithm leaves the reader with two choices to handle the lowest frequency subband and claims that the real solution should lie in-between the two following extremes: one of the method does a lot of range compression, whereas the other leaves the range unchanged, letting the user find the right balance in-between the two solutions.

The method presented in this chapter extends a part of the work of Pattanaik [Pattanaik et al., 1998], by introducing some new concepts: It is based on a model of vision proposed by Barten [Barten, 1999] that merges the measurement on sub- and supra-threshold vision of several authors. It also explicitly defines a way of compressing the dynamic range of an image, by leaving the just-noticeable details unchanged and influencing only the big contrast regions of a scene. Finally, it is able to map a large field of view image into the fields of view of a regular image in a way that is consistent with the perceptual model. The description starts with a review of the human visual system properties that are relevant to the perception of a scene. It then summarises the model of contrast perception proposed by Barten [Barten, 1999]. Finally, it explains how to use this model to render a scene.

6.4 Performance of the human visual system

The human eye contains two types of photo sensors: the rod and the cones. The cones are responsible for the vision at high luminance level, or *photopic* vision, and are capable of colour perception, whereas the rods are more sensitive at low luminance levels, i.e. at *scotopic* vision, and are not capable of distinguishing colours. The distinctions among the colours is enabled by the three types of cones present in the retina [Stockman et al., 1993], whose sensitivities are depicted in Figure 5.3. The human visual system can handle scenes that have luminance values that range from 10'000 to 1 from highlights to shadows, while a single cone is only able to handle a dynamic range of about 100 to 1. This apparent contradiction can be explained by assuming that the human visual system responds to contrast rather than to absolute luminance values. It is believed that it locally adapts to the current luminance. From now on, we will focus our attention on contrast perception, and our goal is to produce an image whose perceived contrast is as close as possible to the contrast perceived in the original scene.

The relationship between a contrast present in a scene and the contrast perceived by a human being is often described by a contrast sensitivity function (CSF). More specifically, the *contrast sensitivity* measures the smallest noticeable contrast - or *threshold* - in a given situation, but does not define how much contrast is perceived in an image. A more general description of contrast perception is given by the *contrast discrimination function*, which measures the smallest perceivable difference between two similar contrasts. The contrast sensitivity is a function that depends on luminance, spatial frequency, temporal frequency, noise and field of view. Unfortunately, it is often presented as a function of two variables in the literature, leading to apparently contradictory results. Nevertheless, the published measurements can be consistently merged into a single multi-variable model, as was done by Barten [Barten, 1999], for monochromatic signals. Even if this last author managed to merge the different contrast sensitivity measurements of a large set of authors into a single model, this model contains many variables that have to be tuned to match each individual measurement, leaving a doubt on what is the right perceptual parametrisation for our purposes.

Since we are dealing with photography, we will assume that the temporal frequency will not come into play for the rest of the chapter, even if it still could play a role because of the eye swapping across an image and generating patterns on the retina varying over time.

Remark. Unless stated explicitly, *contrast sensitivity* refers to the contrast sensitivity to achromatic light.

6.5 Contrast definitions and measurement

Psycho-physical experiments measure the sensitivity of an observer to the contrast of a set of patterns. These patterns are very simple stimuli, for example a square pattern on a uniform background, a sinusoidal pattern, or a Gabor patch [Watson and Solomon, 1997]. In order to characterise the contrast of these patterns, very basic contrast definitions are used: One of these is the Weber contrast, defined as

$$C^W = \frac{\Delta L}{L},$$

where ΔL is the variation in luminance, and L is the background luminance. This definition suits well the stimulus composed of a square patch on a uniform background: ΔL is the difference in luminance of the square and the background, and L is the background luminance. In the case of a sinusoidal pattern, a more suitable definition is given by the Michelson contrast [Michelson, 1927]

$$C^M = \frac{L_{\max} - L_{\min}}{L_{\max} + L_{\min}},$$

where $L_{\max}$ and $L_{\min}$ are the maximum and the minimum luminance that can be found in the sinusoid. These definitions are not equivalent and do not even share the same range of values: Weber contrast can range from -1 to ∞, whereas Michelson contrast can range from 0 to 1. For natural images, none of these definition is suitable: there is a need to know the contrast *locally* in the image. Peli [Peli, 1990] addressed the issue and proposed a band-limited contrast, given by

$$C_j^P(x,y) = \frac{\psi_j * I(x,y)}{\phi_j * I(x,y)}, \tag{6.1}$$

where ψ_j is a band-pass filter, ϕ_j is a low-pass filter and $I(x,y)$ is the image. Peli's contrast defines the contrast value for each pixel of the image. Note that in order to be well defined, and given that $I(x,y)$ and ϕ_j are positive real valued (integrable) functions (and ψ_j is integrable as well), it is sufficient that the support of ψ_j is contained in the support of ϕ_j, and that ϕ_j exhibits no zero-crossings. This prevents the contrast from going to infinity. If the conditions are fulfilled, $\phi_j * I(x,y) = 0$ implies $C_j^P(x,y) = 0$.

When implementing an algorithm using a pyramidal decomposition, the band-pass filter used to compute the contrast can be found by subtracting two low-pass filters at different scales. Equation (6.1) can be rewritten as

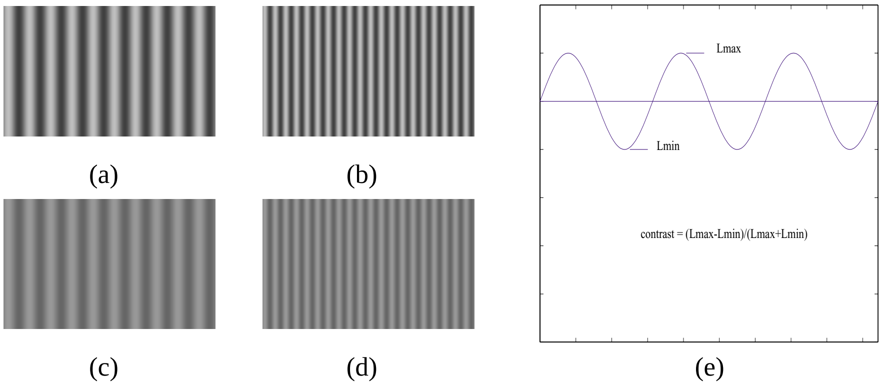
$$C_j^P(x,y) = \frac{(\phi_j - \phi_{j+1}) * I(x,y)}{\phi_{j+1} * I(x,y)},$$

where ϕ_{j+1} is the low-pass filter used to compute the image of lower frequency in the pyramid.

Lubin [Lubin, 1995] modified this contrast measure by subtracting the band-pass band by the next lower low-pass band in the pyramid, leading to the following contrast definition:

$$C_j^L(x,y) = \frac{(\phi_j - \phi_{j+1}) * I(x,y)}{\phi_{j+2} * I(x,y)}.$$

Peli and Lubin definitions of contrast are very similar except that Peli's contrast tend to exhibit more overshoot around edges. Nevertheless, both definitions suffer from the same drawback



(a) (b)

(c) (d) (e)

Figure 6.3: Illustration of contrast (a) High contrast, low frequency (b) High contrast, high frequency (c) Low contrast, low frequency (d) Low contrast, high frequency (e) Michelson contrast definition.

that can be understood by studying the response to a sinusoidal stimulus: The response oscillates at the same frequency than the stimulus. More generally, when there is an edge in the image, the value of contrast is equal to zero at the exact location of the edge, which is a very undesirable property of these contrast measures. This property is visible in Figure 6.5(f).

To circumvent these problems Vandergheynst [Vandergheynst and Gobbers, 2001; Winkler and Vandergheynst, 1999] proposed a contrast measure, which, if used on one-dimensional signals, uses the envelope of the signal. In the case of a sinusoidal signal, the envelope of the signal is equal to the amplitude of the sinusoid; thus the response to a sinusoid is a constant value. The envelope of a signal is illustrated in Figure 6.4, and can be computed using a Hilbert transform [Haykin, 1994]. In two dimensions, there is no single Hilbert transform, and the envelope has to be computed using a set of analytic filters [Vandergheynst and Gobbers, 2001; Winkler and Vandergheynst, 1999].

6.5.1 Notations

The functions used throughout the chapter are often described in Fourier - or frequency - domain [Oppenheim and Schafer, 1989]. Whenever a function is in frequency domain, it will be denoted by a “ ˆ ” , for example if $\phi_j(x,y)$ is a function in spatial domain, then $\hat{\phi}_j\,(\omega,\varphi)$ denotes its Fourier transform.

6.5.2 Isotropic local contrast

A method proposed by Vandergheynst [Vandergheynst and Gobbers, 2001; Winkler and Vandergheynst, 1999] defines a local contrast measure, which is isotropic and exhibits a flat response when confronted to a sinusoidal signal. For each frequency band j, the method uses N oriented filters ψ_{jk}, which are defined by their Fourier Transform $\hat{\psi}_{jk}$ [Oppenheim and Schafer, 1989]. The system is expressed in Fourier domain and in polar coordinates:

$$\hat{\psi}_{jk}\,(\omega,\varphi) = \hat{\Gamma}_j(\omega)\eta_k\,(\varphi)\,,$$

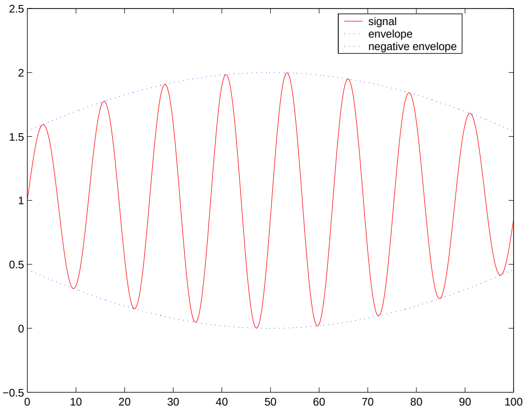


Figure 6.4: Plot of a signal and of its envelope. The envelope measures the amplitude of the signal modulation and can be used to compute the contrast.

where ω is the spacial frequency and φ is the angle in the Fourier plane. $\hat{\Gamma}\left(\cdot\right)$ is the radial frequency response and $\eta\left(\cdot\right)$ is the angular frequency response. A simple constraint on $\hat{\Gamma}\left(\cdot\right)$ and $\eta\left(\cdot\right)$ is given by

$$\sum_{k=0}^{N-1}\eta\left(\varphi-\frac{2\pi}{k}\right)=1,$$
$$\sum_{j=0}^{P-1}\hat{\Gamma}_j(\omega)=1. \qquad (6.2)$$

where P is the number of subbands contained in the multi-resolution pyramidal decomposition of the original image. These function cover the whole frequency plane, and allow a reconstruction of the image I (expressed in polar coordinates)

$$I(r,\varphi)=\sum_{k=0}^{N-1}\sum_{j=0}^{P-1}\eta\left(\varphi-\frac{2\pi}{k}\right)*\Gamma_j(r)*I(r,\varphi),$$

where $*$ denotes the convolution operation. The radial band-pass function $\hat{\Gamma}_j(\omega)$ is given by the difference of the the scaling functions $\hat{\phi}$ used to build the multi-resolution pyramid

$$\hat{\Gamma}_j(\omega)=\hat{\phi}_j\left(\omega\right)-\hat{\phi}_j\left(2\omega\right)\triangleq\hat{\phi}_j\left(\omega\right)-\hat{\phi}_{j+1}\left(\omega\right).$$

In our implementation, the pyramid is a difference-of-Gaussians multi-resolution pyramid, where $\hat{\phi}_j\left(\omega\right)$ is given by

$$\hat{\phi}_j\left(\omega\right)\triangleq\hat{\phi}\left(2^j\omega\right)$$
$$\hat{\phi}\left(\omega\right)=e^{-\frac{\omega^2}{2\sigma^2}}. \qquad (6.3)$$

In order to avoid aliasing when sub-sampling the different bands of the pyramid, σ is set to $\sigma = 0.45$. The angular frequency response is given by a combination of Schwarz functions, which are described in Appendix C.

Finally, the contrast is computed as a sum of the in-phase and quadrature components [Haykin, 1994] of each filtered image

$$C_j\left(x,y\right) = \frac{\sqrt{2\sum_{k=0}^{N/2} \left|\psi_{jk} * I\left(x,y\right)\right|^2}}{\phi_{j+1} * I\left(x,y\right)}, \tag{6.4}$$

divided by a low-pass version of the image $\phi_{j+1} * I\left(x,y\right)$. Figure 6.5 shows an example of contrast computation along with the different filters involved. We must say that the isotropic contrast measure exhibits an artefact where there are edges of big amplitude, as can be seen in Figure 6.7. The contrast decreases smoothly across the edge and shows some bleeding. At the present time, we are unable to say if these artifacts are due to an implementation problem, a numerical problem or are more fundamental.

6.5.3 Yet another contrast measure

The contrast measure of Section 6.5.2 has all the properties needed to describe the contrast, nevertheless it has one major drawback: it is difficult to implement efficiently. Our goal is not to measure the contrast in the image, but to modify the image in order to maintain the contrast, a task that, in some cases, can be performed without measuring the contrast, as will be seen in Section 6.9.1. For our purposes, the contrast should exhibit a maximum at the edge locations, should decreases monotonically away from the edge, and should not exhibit any zero crossings. This can simply be done by taking the derivative amplitude of the image, which can be computed by combining the derivatives along the lines and the columns of the image:

$$\begin{aligned}
\hat{D}_j^{(x)}\left(\omega_x,\omega_y\right) &= j\omega_x \cdot \hat{\phi}_j \cdot \hat{I}\left(\omega_x,\omega_y\right) \\
\hat{D}_j^{(y)}\left(\omega_x,\omega_y\right) &= j\omega_y \cdot \hat{\phi}_j \cdot \hat{I}\left(\omega_x,\omega_y\right) \\
D_j(x,y) &= \sqrt{\left(D_j^{(x)}\left(x,y\right)\right)^2 + \left(D_j^{(y)}\left(x,y\right)\right)^2}
\end{aligned}$$

where D_j is the amplitude of the derivative of the image in the j-th level of the multi-resolution pyramid, and $\hat{\phi}_j$ the low-pass filter (expressed in Fourier domain) used to get the j-th level of the multi-resolution pyramid from image I. If the cutoff of the low-pass filter is small enough compared to the Nyquist sampling rate, the derivative can be computed using a very simple difference operator:

$$D_j(x,y) \simeq 2^{j-1}\sqrt{\left[D_j\left(x+1,y\right) - D_j\left(x-1,y\right)\right]^2 + \left[D_j\left(x,y+1\right) - D_j\left(x,y-1\right)\right]^2}\,.$$

Finally the contrast $C_j^D\left(x,y\right)$ is obtained by dividing the derivative by a low-pass image:

$$C_j^D\left(x,y\right) = \begin{cases} \frac{D_j(x,y)}{\phi_{j+1} * I(x,y)}, & D_j(x,y) > 0 \\ 0, & \text{else} \end{cases}. \tag{6.5}$$

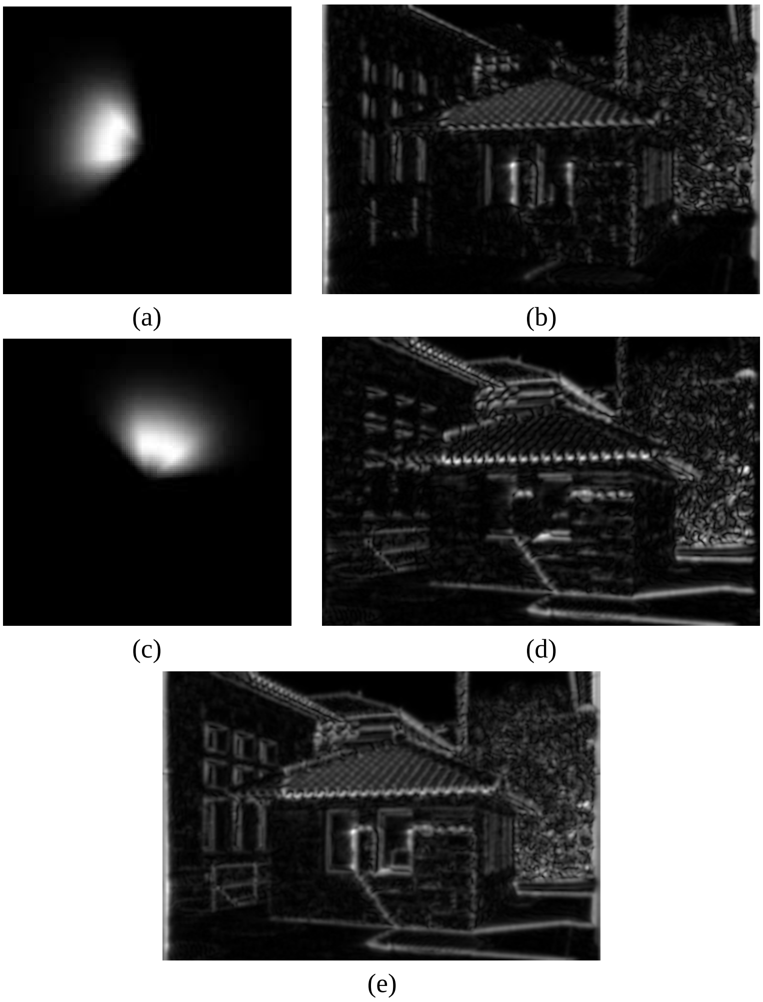
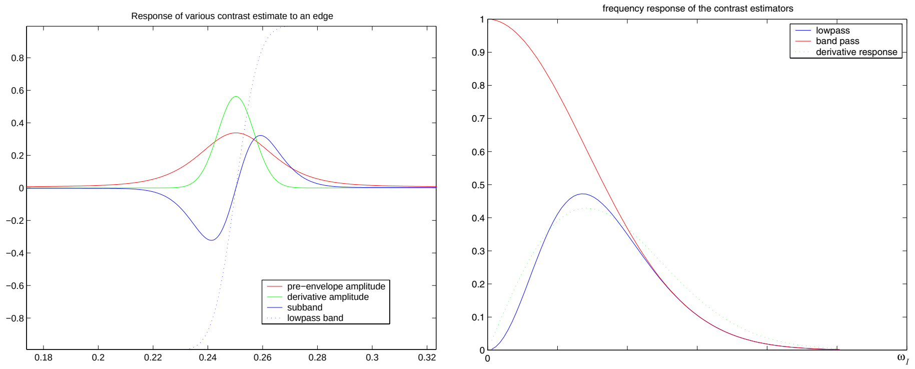


Figure 6.5: Illustration of the isotropic contrast computation (a) Horizontal filter. (b) Horizontal contrast. (c) Vertical filter. (d) Vertical contrast. (c) Isotropic contrast G_j.



(a) (b)

Figure 6.6: Contrast estimator evaluation. (a) Step response of the contrast estimators. Peli's contrast is equal to 0 at the edge location. The derivative contrast is more centred around the edge than the isotropic contrast. (b) Frequency responses of the contrast estimators. Peli's and the isotropic contrast live in the subband region. ω_I denotes the Nyquist frequency of the image [Oppenheim and Schafer, 1989].

This estimate of contrast has its maximum at the edge locations, and has no zero crossings. Nevertheless, with a sinusoidal input, it oscillates with the same frequency of the sinusoid. Furthermore, when considering all the frequency components, it is not normalised like the isotropic contrast through Equation (6.2). Nevertheless, it is perfectly suitable for the application that follows.

Figure 6.6 shows the responses of the contrast measures for a sharp edge; it shows the edge that passed through the low-pass filter $\hat{\phi}_j$, along with Peli's contrast and the isotropic contrast. The curves in the figure have been obtained without dividing by the low-pass function, i.e. if C is the contrast, the plot shows $C \cdot \phi_{j+1} * I\,(x, y)$. We can see that the derivative based contrast is more centred around the edge, and fails to completely catch the support of the subband. Figure 6.7 shows the contrast of an image, estimated with Peli's contrast, the isotropic contrast and the derivative-based contrast.

Remark. The derivative based contrast could be extended by adding Peli's contrast to it; that would be approximately the in-phase and quadrature sum of the signal. Nevertheless, such a measure of contrast would have other problems like, for example, zero-crossings.

6.6 Threshold model of vision

The threshold model of vision describes the just noticeable differences in luminance that the eye is able to perceive in a given situation.

To give some intuition about contrast sensitivity, here are some rules:

- *The contrast perception does not depend on luminance*. This applies at very high lu-

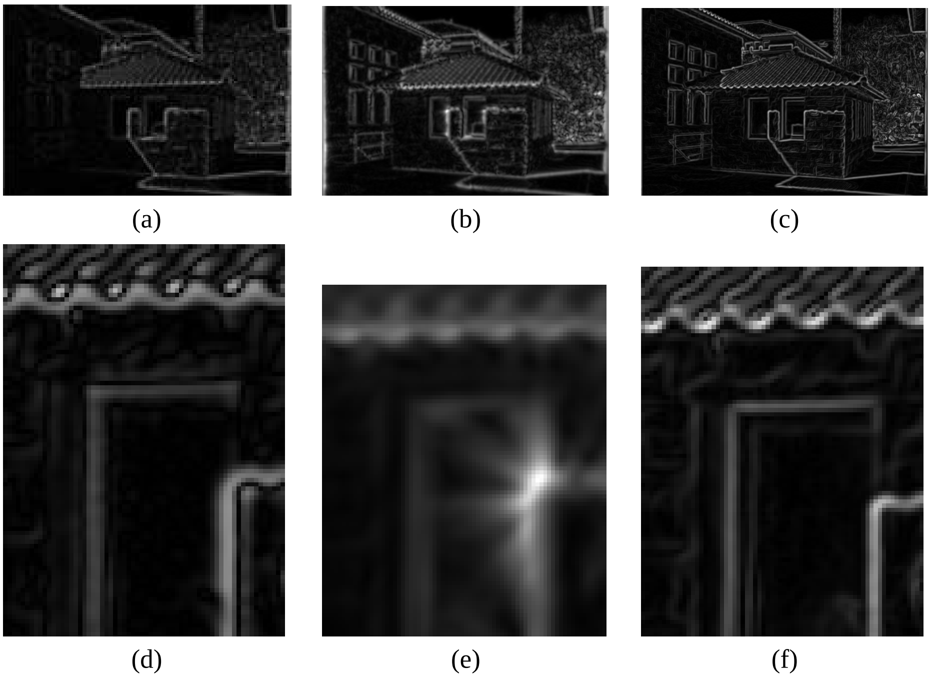
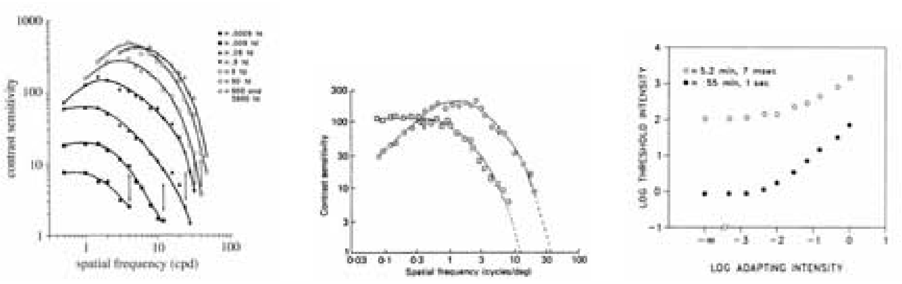


Figure 6.7: Contrast metric comparison. (a) and (d) Peli's contrast (absolute value). (b) and (e) Isotropic contrast. (c) and (f) Derivative-based contrast. Peli's contrast is equal to 0 at the edge location, that appears as a black line in the middle of two side-lobes.



(a) (b) (c)

Figure 6.8: Contrast sensitivity function of the eye. (a) Sensitivity at different retinal illumination levels (b) sensitivity for a monochromatic signal [circles] and a red-green signal [squares]. (c) sensitivity for two different temporal frequencies plotted against the mean luminance of the display. Adapted from [Pattanaik et al., 1998].

minance and/or at low spatial frequency and is known as the *Weber's Law*[1].

- *The contrast perception and the colour saturation increases with luminance*. This is generally true, except at very high luminance or very low frequency. This rule is sometimes referred to as *sub-Weber behaviour*.
- At low spatial frequency, the contrast sensitivity increases linearly with frequency.

Figure 6.8(a) shows an example of measured contrast sensitivity functions. It has been measured for an achromatic signal, at different frequencies and luminance levels. The contrast sensitivity increases with increasing luminance level, thus showing a *sub-Weber behaviour*, until it reaches a particular value and stays constant. Figure 6.8(b) shows the contrast sensitivity function for a monochromatic signal and a red-green opponent colour signal, at a particular luminance level. The figure shows that, at high spatial frequency, the contrast sensitivity of the eye is better for monochromatic signals. It also shows that the sensitivity to monochromatic signals shows a band-pass behaviour, whereas the contrast sensitivity to colour signals shows a low-pass behaviour. Figure 6.8(c) shows the detection threshold against time. Time issues come into play when our eye swaps across an image, inducing timely varying patterns on the retina. The figure shows that if the exposure time to a pattern is short, the contrast sensitivity increases with luminance level. The lower curve tends to a slope equal to 1, and thus follows Weber's law - i.e. contrast perception is independent of luminance. The upper curve shows a sub-weber behaviour, and has a slope of 0.85.

In his book, Barten [Barten, 1999] tries to build a contrast sensitivity function by modelling the various physical phenomena that affect our perception. These phenomena are:

- The photon noise of the light.
- The neural noise.

[1] Weber's law is valid in the range of luminance where the curves of Figure 6.9(a) overlap.

- The lateral inhibition process in the retina.
- The optical point spread function or optical *modulation transfer function*.
- The signal-to-noise ratio of the photo-receptors.

Photon noise occurs because of the discrete nature of light, and its effects disappears at high luminance levels. Photon noise causes Weber's law to fail at low luminance. Neural noise is generated by the statistical fluctuation in the signal transported to the brain. Lateral inhibition is a mechanism that is generated by the neural inter-connection. The optical point spread function is determined mainly by the eye lens and by the structure of the retina [2].

Barten gives a model for the **contrast sensitivity function** of the eye with the following formula:

$$C_T(\omega, L, L_{pup}) = \frac{M_{opt}(\omega, L_{pup})}{k \cdot \sqrt{\frac{2}{T}\left(\frac{1}{X_o^2} + \frac{1}{X_{\max}^2} + \frac{\omega^2}{N_{\max}^2}\right)\left(\frac{1}{\eta p E(L, L_{pup})} + \frac{\Phi_0}{1 - e^{-\left(\frac{\omega}{\omega_0}\right)^2}}\right)}} \tag{6.6}$$

where ω is the spatial frequency. $M_{opt}(\omega, L_{pup})$ is the **modulation transfer function** of the eye optics. k is the detection signal-to-noise ratio of the photo-receptors in the eye. $\frac{1}{\eta p E(L, L_{pup})}$ is the photon noise. $\frac{\Phi_0}{1 - e^{-\left(\frac{\omega}{\omega_0}\right)^2}}$ models the lateral inhibition in the retina and $\frac{2}{T}\left(\frac{1}{X_o^2} + \frac{1}{X_{\max}^2} + \frac{\omega^2}{N_{\max}^2}\right)$ models the spatio-temporal integration volume (measured in the eye) over which the noise has an influence.

The modulation transfer function is assumed to be Gaussian and can be computed using

$$\begin{aligned} M_{opt}(\omega, L_{pup}) &= e^{-2\pi^2\sigma^2\omega^2} \\ \sigma &= \sqrt{\sigma_0^2 + \left[C_{ab} d\left(L_{pup}\right)\right]^2} \end{aligned} \tag{6.7}$$

where $d\left(L_{pup}\right)$ is diameter of the pupil, ω the spatial frequency in $\mathrm{cycles}/\deg$, σ_0 is the point spread function basic width and C_{ab} describes the increase of the point spread function width at increasing pupil size. Typical values for these parameters are given in Table 6.1. The pupil diameter d can be deduced from the mean luminance L_{pup} using [Le Grand, 1969, p. 99]

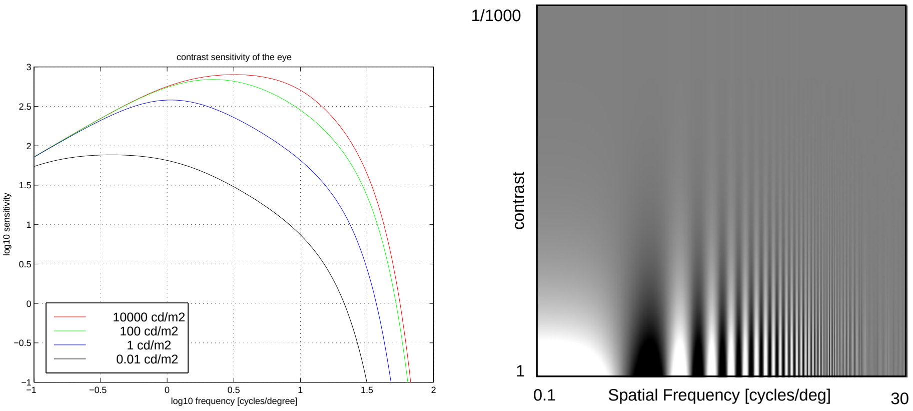
$$d\left(L_{pup}\right) = 5 - 3\tanh(0.4\log_{10} L_{pup})\ [\mathrm{mm}]. \tag{6.8}$$

To compute the **photon noise** $\frac{1}{\eta p E(L, L_{pup})}$, η is the quantum efficiency, p a conversion factor to keep the unit system coherent and $E\left(L, L_{pup}\right)$ the retinal illuminance. The retinal illuminance can be computed using [Moon and Spencer, 1944; Jacobs, 1944]

$$E\left(L, L_{pup}\right) = \frac{\pi d^2}{4} L \left[1 - \left(\frac{d\left(L_{pup}\right)}{9.7}\right)^2 + \left(\frac{d\left(L_{pup}\right)}{12.4}\right)^4\right] \tag{6.9}$$

where d is the pupil diameter expressed in mm and L is the luminance. The retinal illuminance is expressed in Trolands (Td). The volume over which the noise is integrated (in

[2] It is believed that diffraction effects are negligible.



(a) (b)

Figure 6.9: (a) Contrast sensitivity function of the human eye to achromatic light, plotted as a function of spatial frequency and mean luminance. Time considerations are left apart - the stimulus is assumed to be static (b) Illustration of contrast sensitivity: one can see that the height at which the pattern disappears varies with frequency. Adapted from [Nadenau, 2000].

p	$= 1.24 \times 10^6 \frac{\text{photons}}{\text{sec} \cdot \text{deg}^2 \cdot \text{Td}}$	k	$= 3.0$	$N_{\max}$	$= 15$ cycles
σ_0	$= 0.0083$ deg	T	$= 0.1$ sec	$X_{\max}$	$= 12$ deg
C_{ab}	$= 0.0013$ deg /mm	η	$= 0.03$	X_0	$= 60$ deg
Φ_0	$= 3 \times 10^{-8}$ sec deg^2	ω_0	$= 7$ cycles/ deg		

Table 6.1: Typical values for the parameters of the contrast sensitivity function

the eye) is given by the scene: $X_{\max}$ is the maximum angular size over which the eye can integrate, X_0 is the angular size of the object the observer is looking at (i.e. the size of the screen), $N_{\max}$ is the maximum number of cycles that the eye is capable of integrating and T is the integration time of the eye. The sizes are measured in degrees and we assume that the objects of the scene have more or less the same height and width. Typical values for all these parameters are given in Table 6.1, and the contrast sensitivity function is plotted in Figure 6.9.

6.7 Supra-threshold model of vision

In opposition to threshold models of vision that specify if there is any contrast to be seen, supra-threshold model of vision try to quantify how much contrast is perceived in an image. The experiment used to measure the contrast discrimination capability has been carried out by asking people to distinguish between two sinusoidal patterns of different contrasts. For a given contrast, it measures how much contrast has to be added to the other pattern until the difference becomes apparent to the observer. This leads to the *contrast discrimination function*.

It turns out that the contrast discrimination function for monochromatic signals and photopic vision can be approximated by

$$\triangle C_t(m) = \sqrt{\frac{C_t^2 + 0.04\,k^2 m^2}{1 + 0.004\,k\,m/C_t} + m^2} - m, \tag{6.10}$$

where m is the current contrast, C_t is the contrast threshold in the current viewing conditions and $\triangle C_t$ is the amount of contrast that has to be added to m in order that an observer notices a difference[3]. k is the signal-to-noise ratio whose value can be found in Table 6.1. The beauty of this expression lies in the fact that once the contrast threshold has been computed, it only depends on the real contrast (k has to be known to compute the contrast threshold). The expression can be refined in order to account for noise, which can affect the perception by *masking* the contrast

$$\triangle C_t(m) = \sqrt{\frac{C_t^2 + k^2 C_n^2 + 0.04\,k^2 m^2}{1 + 0.004\,k\,m/C_t} + m^2} - m, \tag{6.11}$$

where C_n is the average modulation of the noise component, i.e. the standard deviation of the noise at the considered spatial frequency. This last equation can be used in presence of viewing flare, where C_n represents the amount of flare compared to the local luminance of the displayed image. By integrating this equation, we can compute how much contrast there is in a signal in the sense of how many different distinguishable contrast levels there are between a particular signal m and the contrast threshold:

$$C^{\psi}(m) = a \cdot \int_{m_0=0}^{m} \frac{1}{\triangle C_t(m_0)} \cdot dm_0. \tag{6.12}$$

where $\triangle C_t$ can be computed using either Equation (6.10) or Equation (6.11) and a is a correction used to enforce $C^P(C_t) = 1$ for $C_n = 0$ ($a = 0.61$). We will refer to C^{ψ} as the *perceived contrast* (or *psyco-metric contrast* to get a better mnemonic for the exponent ψ). In perceived contrast space, a value of 1 means a just noticeable contrast, and a difference of 1 between any two values means a just noticeable contrast difference. This is independent of the viewing conditions, i.e. independent of the luminance level, noise, field of view, etc. An illustration of the perceived contrast function can be found in Figure 6.10.

6.8 Contrast sensitivity function for chromatic channels

The model presented so far is valid for achromatic light. The sensitivity of the eye to chromatic stimuli has been measured by Van der Horst [Van der Horst and Bouman, 1969], Granger [Granger and J.C., 1973], Kelly [Kelly, 1983], Mullen [Mullen, 1985], Sankeralli [Sankeralli and Mullen, 1996], Poirson [Poirson and Wandell, 1993, 1996], and Nadenau [Nadenau, 2000]. The measure of the sensitivity to chromatic light is more challenging than for achromatic light: It requires the creation of patterns of uniform luminance and a proper alignment of the opponent colours in play. As a consequence, most of the measurements have been done in the range of luminance of a CRT screen. The sensitivity to chromatic light

[3] m stands for *modulation.*

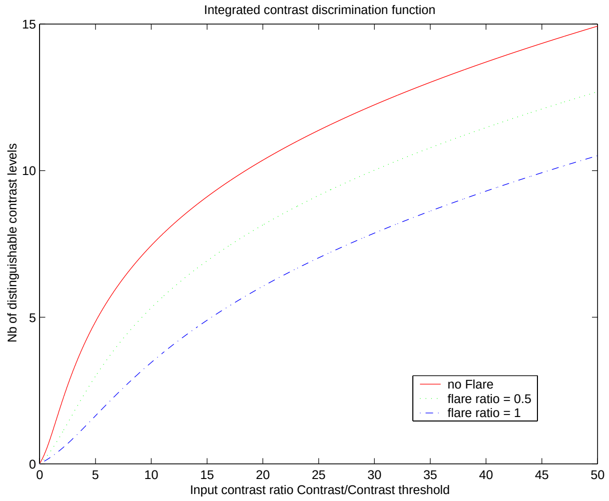


Figure 6.10: Perceived contrast function C^{ψ}. The perceived contrast function is a measure of how much contrast is perceived in a signal. The function is plotted against the contrast relative to the just-noticeable-contrast, for different values of flare. Flare masks our perception of contrast, and more contrast has to be present to get the same impression.

varies with the colour space chosen, leading to differences in the published results. Poirson and Wandell [Poirson and Wandell, 1993, 1996] defined an opponent colour space where the sensitivity in the red-green channel and in blue-yellow channel are uncorrelated, which is a very interesting property for our purposes. Nevertheless, to our knowledge, the chromatic sensitivity for high luminance levels has not been measured yet. In addition, there is little knowledge about supra-threshold models of vision in chromatic channels. Since the measurement of such data is out of the scope of our work, we decided not to use the contrast sensitivity function for chromatic light. Thus, the algorithm will only modify the luminance channel of an image. The colour space used is the opponent colour space defined by Poirson and Wandell [Poirson and Wandell, 1993].

6.9 Rendering algorithm

The goal of this algorithm is to produce an image that, given an output device, tries to preserve the perceived contrast of the scene, defined by (6.12), as much as possible. The algorithm works as follow:

Given a radiance image of the scene, it converts it into a opponent colour space, with one achromatic channel, and two opponent colour channels defined in [Poirson and Wandell, 1993]. Only the achromatic channel is processed by the algorithm. The algorithm uses two parallel work-flows: On the one hand it uses the contrast of Section 6.5.2 or Section 6.5.3, coupled with the perceptual model of Section 6.7 in order to measure the contrast in the image and to determine how this contrast has to be changed in the final image. On the other hand, it uses a subband decomposition of the image to apply the changes in contrast and finally reconstruct a new image. The global algorithm is depicted in Figure 6.11, and the details applied to each subband is shown in Figure 6.12.

Let $I_j^R(x,y)$ be the reconstructed image, let j be the subband index and let N be the number of subbands used. The reconstruction of the image starts with the band of lowest frequency $I_N^R(x,y)$, which in our case is an image composed of one single pixel[4]. This image is set to a value L_{mout}, which is derived from the mean luminance of the output device: $I_N^R(x,y) = L_{mout}$. Then, each subband is added iteratively :

$$I_{j-1}^R(x,y) = I_j^R(x,y) + (\phi_{j-1} - \phi_j) * I(x,y) \cdot G_j(x,y), \tag{6.13}$$

up to the final image $I_1^R(x,y)$. $I(x,y)$ is the original image, ϕ_j is the Gaussian low-pass filter defined in Equation (6.3) and $G_j(x,y)$ is a gain image derived from the contrast and from the model of the eye. In practice, some down-sampling and up-sampling are involved, but are not explicit in Equation (6.13) to simplify the notation[5]. Once, the final image $I_1^R(x,y)$ is obtained, it can be transformed into a standard colour space for rendering. If the dynamic range of the final image does not fit the output device, the parameter that controls the dynamic range companding is modified accordingly, and the whole iteration of Equation (6.13) is run once again to get to the desired image.

6.9.1 Simple gain image computation

To compute the gain image $G_j(x,y)$, we need to know the contrast of the subband, the contrast thresholds for the original viewing conditions, and the contrast thresholds for the output viewing conditions.

For each pixel of the image, a contrast threshold value $T_j\,(x,y)$ for the original viewing conditions is computed using

$$\begin{aligned} T_j\,(x,y) &= C_t\,(\omega, L, L_{pup})\,, \\ \omega &= \omega\left[(\phi_j - \phi_{j+1}) * I\right], \\ L &= \phi_{j+1} * I\,(x,y)\,, \\ L_{pup} &= \phi_p * I\,(x,y)\,, \end{aligned} \tag{6.14}$$

where $C_t\,(\omega, L, L_{pup})$ is defined by Equation (6.6), ω is the centre frequency of the image, and L is the adaptation luminance of the eye, which is assumed to be equal to the local luminance around the pixel and is read from the low-pass image pyramid one level underneath the current image. L_{pup} is the luminance used to compute the pupil diameter, which is assumed to adapt to a frequency of $\frac{1}{5}\,[\mathrm{cycles}/\deg]$, i.e. p is the index of the subband of $\frac{1}{5}\,[\mathrm{cycles}/\deg]$. The centre frequency of the image ω can be computed from the Nyquist frequency ω_I of $I(x,y)$ and from ϕ_j:

$$\begin{aligned} \omega\left[(\phi_j - \phi_{j+1}) * I\right] &= \omega_I \cdot \arg\max_{\omega}\left(\hat{\phi}_j - \hat{\phi}_{j+1}\right), \\ &\simeq \omega_I \cdot 2^{-j} \cdot 0.96 \cdot \sigma. \end{aligned} \tag{6.15}$$

[4] To get a single pixel image, the original image is extended to a size of a power of 2 by mirroring across the borders.

[5] Fundamentally, there is no need of using up- and down-sampling, but it makes the algorithm run faster and with less memory.

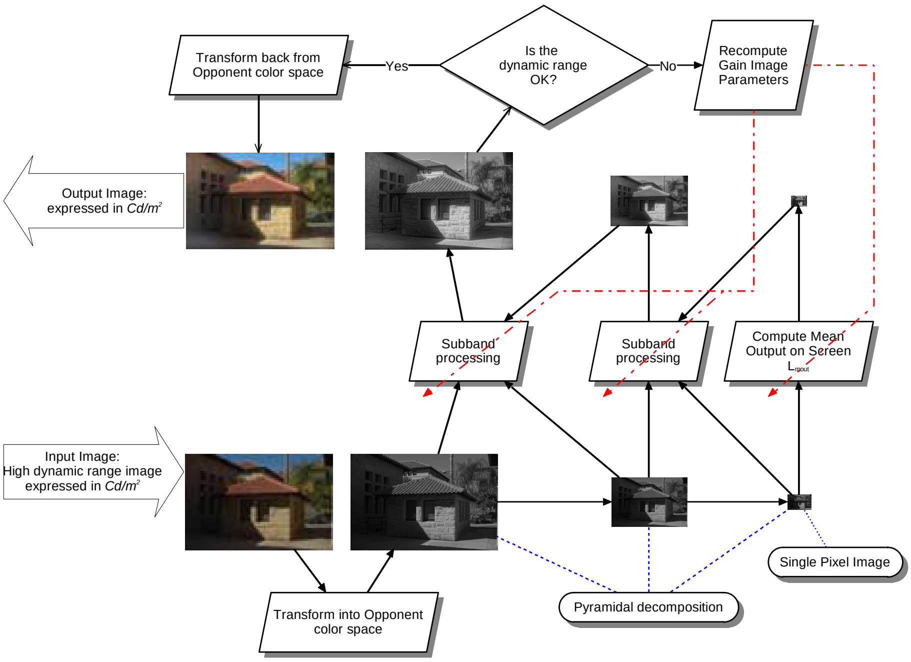

Transform back from Opponent color space
Yes
Is the dynamic range OK?
No
Recompute Gain Image Parameters
Output Image: expressed in *Cd/m*²
Subband processing
Subband processing
Compute Mean Output on Screen L_{mout}
Input Image: High dynamic range image expressed in *Cd/m*²
Single Pixel Image
Pyramidal decomposition
Transform into Opponent color space

Figure 6.11: Rendering algorithm scheme. The image is decomposed into frequency subbands. Each subband is processed by the algorithm described in Figure 6.12. The low frequency subband, i.e. the single pixel image, is set equal to the mean output luminance. The companding function is adapted iteratively such that the output dynamic range converges to the desired value.

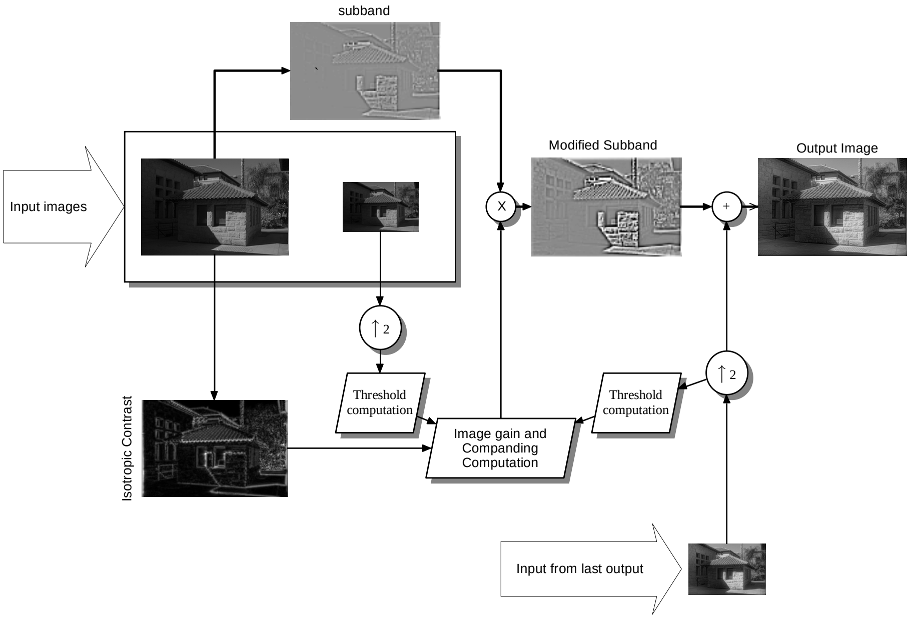


Figure 6.12: Rendering algorithm for one particular subband. The subband is multiplied by a gain image. The gain image is computed using the adaptation of the eye by viewing the original scene, and the adaptation of the eye in the output conditions by assuming that the user watches the image on the computer screen. The gain image also involves a companding function that reduces the contrast by maintaining the little details and affecting the important ones.

The frequency of the image ω_I can be either derived from the focal length of the image (if it is known), or by assuming the observer looks at the image from a distance equal to the image diagonal. The frequency should be expressed in cycles/deg.

The contrast thresholds for the output viewing conditions $T_j^R(x,y)$ can be obtained similarly be using

$$\begin{aligned} T_j^R(x,y) &= C_t\left(\omega, L, L_{pup}\right), \\ \omega &= \omega\left[I_j^R(x,y)\right], \\ L &= \max\left(I_{j+1}^R(x,y), F_{j+1}\right), \\ L_{pup} &= L_{mout}, \end{aligned} \tag{6.16}$$

where $I_{j+1}^R(x,y)$ is the image reconstructed at the previous iteration, and $\omega\left[I_j^R(x,y)\right]$ is the frequency of the image being reconstructed[6], computed using the same approach as in (6.15). F_{j+1} is the output flare value (if any), and L_{mout} is the mean value of the reconstructed image.

In order to reproduce the perceived contrast (without compressing it) by ignoring flare issues, the gain image is computed using

$$G_j^s(x,y) = \frac{T_j^R(x,y) \cdot I_{j+1}^R(x,y)}{T_j(x,y) \cdot \phi_{j+1} * I(x,y)}. \tag{6.17}$$

In this expression, the contrast gain is equal to $T_j^R(x,y)/T_j(x,y)$. Note that the contrast image is absent from the expression. Unfortunately, the image generated by Equation (6.17) cannot be displayed on the output device and some compression has to be introduced. Fundamentally, this is due to the fact that our perception is more accurate at high luminance, thus, by decreasing the luminance of the scene while rendering it on a screen, its contrast has to be amplified; $T_j^R(x,y)/T_j(x,y)$ being in general bigger than 1.

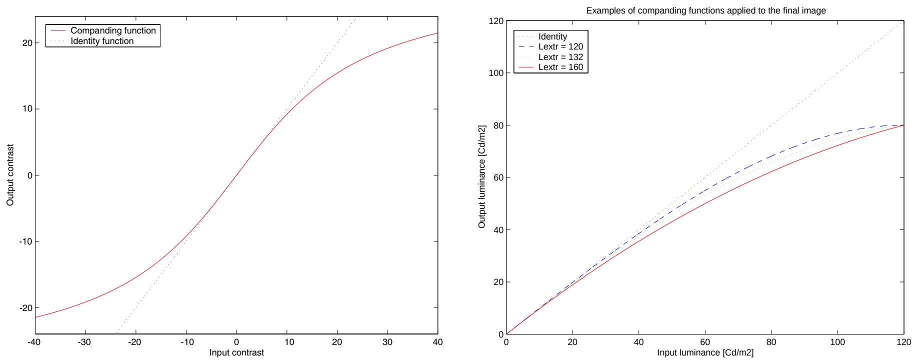
6.9.2 Compressing gain image computation

This subsection introduces a method that compresses the dynamic range of the image by maintaining the little details: Every detail visible in the scene should be visible in the final image.

Let $C_j(x,y)$ be the isotropic contrast defined by Equation (6.4). By using the contrast thresholds $T_j(x,y)$ of Equation (6.14), one can compute how much contrast is perceived in $C_j(x,y)$ by using the perceived contrast $C_j^\psi(x,y)$ given by Equation (6.12) - flare is supposed to be absent in the input image, i.e. $C_n = 0$. Recall that a value of $C_j^\psi(x,y) = 1$, means that there is a just-noticeable-contrast at location (x,y). In order to prevent loosing details in the final image, the companding function should leave the small values unchanged, and is allowed to alter only the large values of contrast. In other words, the companding function should have a derivative equal to 1 at the origin, and should introduce a compression that smoothly increases with increasing inputs. The chosen companding function $S(\cdot)$ is given by

$$S(C) = C_{\max} \cdot \frac{2}{\pi} \arctan\left(\frac{C}{C_{\max}} \cdot \frac{\pi}{2}\right), \tag{6.18}$$

[6] Note that it is not necessary to know the current image content in order to know its frequency.



(a) (b)

Figure 6.13: (a) Companding function used to reduce the contrast of each subband. The small contrast are left untouched whereas the big contrast values are reduced by the function. (b) Companding function used at the very end of the algorithm. The function tries to maintain the mid-tones and the shadows. The dashed line is the identity function.

where $C_{\max}$ is the maximum output of the function. The initial estimate of $C_{\max}$ is set to $C_{\max} = 50$; a plot of this function is depicted in Figure 6.13. By assuming that there is no flare in the output device ($C_n = 0$), the gain image is given by

$$G_j^c(x,y) = \begin{cases} G_j^s(x,y) \cdot \dfrac{\left(C_j^{\psi}\right)^{-1}\left\{S\left[C_j^{\psi}(x,y)\Big|_{C_n=0}\right]\right\}\Big|_{C_n=0}}{C_j(x,y)}, & C_j(x,y) > 0 \\ G_j^s(x,y), & \text{else} \end{cases}, \qquad (6.19)$$

Where G_j^s is given by Equation (6.17). The computation of C^{ψ} as well as the computation of its inverse $\left(C_j^{\psi}\right)^{-1}$ is done numerically and is applied using a lookup table, see Appendix C for the implementation details. Equation (6.19) computes the perceived contrast, applies a companding to it, recomputes the contrast from the companded perceived contrast and evaluates the gain introduced by this whole procedure.

Finally, if flare is present in the output device, the gain image becomes

$$G_j^f(x,y) = \begin{cases} G_j^s(x,y) \cdot \dfrac{\left(C_j^{\psi}\right)^{-1}\left\{S\left[C_j^{\psi}(x,y)\Big|_{C_n=0}\right]\right\}\Big|_{C_n=C_{out}}}{C_j(x,y)}, & C_j(x,y) > 0 \\ G_j^s(x,y), & \text{else} \end{cases}, \qquad (6.20)$$

where C_{out} is the output flare contrast. The output flare contrast is given by

$$C_{out}(x,y) = \begin{cases} \dfrac{F_j}{I_{j+1}^R(x,y)}, & I_{j+1}^R(x,y) > 0 \\ 0, & \text{else} \end{cases},$$

where $I_{j+1}^R(x,y)$ is the output image reconstructed at the previous iteration, and F_j is the flare value (expressed in $\mathrm{cd/m^2}$).

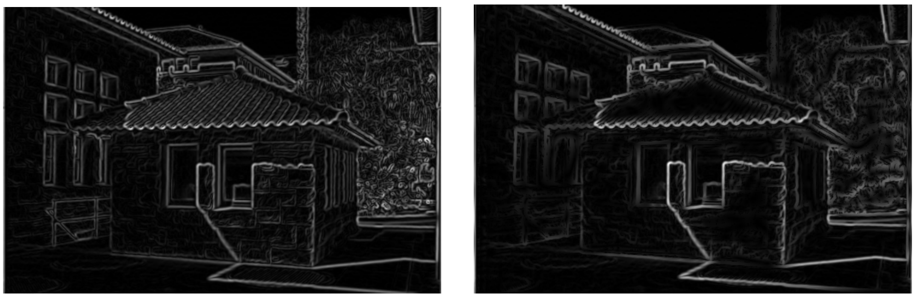


Figure 6.14: Contrast measurements (a) Isotropic contrast G_j (b) Isotropic contrast G_j^*, with small details removed.

6.9.3 Selective compressing gain image computation

To compress the dynamic range of the image, the algorithm compresses the regions of big contrast in the image. This has the undesirable side-effect of reducing the perceived contrast in the image, and gives an impression of haziness to the picture. Compressing details that are localised in a small region of the image does not contribute much to the overall dynamic range compression, but impairs the final image quality. Ideally, the companding function should affect only the edges whose influence span an important part of the image. These edges are present in the high frequency contrast images - if the edge is sharp - as well as in the lower frequency contrast images, whereas a small detail is only present in the high frequency images of the contrast pyramid (see [Dragotti and Vetterli, 2000] for a more formal approach to this statement). From this observation, it is tempting to get rid of the companding of the small details, by comparing the different contrast images: since the contrast images are built on an L_1 normalisation, introduced by Equation (6.2) with the isotropic contrast, a strong edge should generate the same contrast value in different subbands. Following this reasoning, a very simple - but not very consistent - method of suppressing the companding of the small details consists in replacing the contrast images by the minimum between the contrast in the current subband and the one in the lower subband. This is performed up to the frequency where the details are not considered to be small anymore. In other words, the new contrast images are given by

$$C_j^*\left(x,y\right) = \begin{cases} \min\left[C_j\left(x,y\right), C_{j+1}^*\left(x,y\right)\right], & j < J_{\mathrm{det}} \\ C_j\left(x,y\right), & \text{else} \end{cases} .$$

In practice, we chose to perform this transformation up to the band J_{det} corresponding to 0.5 $\mathrm{cycles}/\deg$. The result of this transformation is shown in Figure 6.14(a) and 6.14(b). Then, if flare is disregarded, the gain image can be computed exactly as previously by replacing $C_j\left(x,y\right)$ by $C_j^*\left(x,y\right)$ in Equation (6.19):

$$G_j^*(x,y) = \begin{cases} G_j^s(x,y) \cdot \frac{\left(C_j^\psi\right)^{-1}\left\{S\left[C_j^{\psi *}(x,y)\Big|_{C_n=0}\right]\right\}\Big|_{C_n=0}}{C_j^*(x,y)}, & C_j^*\left(x,y\right) > 0 \quad , \\ G_j^s(x,y) & \text{else.} \end{cases} \tag{6.21}$$

where $C_j^{\psi *}\left(x,y\right)$ is the perceived contrast computed using $C_j^*\left(x,y\right)$. If flare is considered, then the gain image is computed using

$$G_j^{f*}(x,y) = \begin{cases} G_j^s(x,y) \cdot \left\{ \begin{array}{l} \frac{\left(C_j^{\psi}\right)^{-1}\left\{S\left[C_j^{\psi *}(x,y)\Big|_{C_n=0}\right]\right\}\Big|_{C_n=0}}{C_j^*(x,y)} \cdot \\ \cdot \frac{C_j^{\psi -1}\left\{C_j^{\psi}(x,y)\Big|_{C_n=0}\right\}\Big|_{C_n=C_{out}}}{C_j(x,y)} \end{array} \right\} & C_j^*\left(x,y\right) > 0 \\ G_j^s(x,y) & \text{else} \end{cases} .$$

Figure 6.15 illustrates the use of this last expression. The drawback of the algorithm in Equation (6.21) is that the small details close to big edges do not get enhanced like the ones distant from the edges.

6.9.4 Flare

Flare is supposed to be absent in the raw data image used by the algorithm. Furthermore, it is assumed that no flare is involved if observing the real scene. The algorithm is able to account for the viewing flare on the screen. Ideally, the algorithm should compute the input viewing flare from the raw data, i.e. the flare caused by the scattering of the light in the lens and the cornea of the eye. Then, use the function C^{ψ} defined in Equation (6.11) to compute the perceived contrast under the viewing flare conditions, and then process the image as before. Unfortunately, we do not have a model for the input viewing flare, so we just ignored it, and hope that the flare present in the raw data compensates for this approximation.

The scattering of the light in the eye is responsible for the reduced sensitivity of the eye at very high luminance levels. This decrease in sensitivity is *not* modelled by Equation (6.6), thus the algorithm does not take it into account. It is important to keep this fact in mind when fitting the dynamic range of the image to the dynamic range of the output device.

6.9.5 Fitting the dynamic range

The game of fitting the dynamic range of the image to the range of the output device consists in choosing the right output mean luminance L_{mout} and the right companding factor $C_{\max}$. By assuming that the user is looking at the picture on the screen, the output luminance will approximately range between 0 and 80 $\mathrm{cd/m^2}$[7]. The mean output luminance L_{mout} should be chosen such that the image histogram is well centred in the output luminance range. L_{mout} is found by computing the histogram of the original radiance image and applying the following formula:

$$L_{mout} = L_{S\max} \cdot \frac{L_m - L_{2\%}}{L_{98\%} - L_{2\%}} \tag{6.22}$$

where $L_{S\max}$ is the screen maximum luminance (i.e. 80 $\mathrm{cd/m^2}$), L_m is the mean value of the original image, $L_{98\%}$ and $L_{2\%}$ are the luminance values at 2% and at 98% of the histogram data. If the shape of the histogram does not change significantly through the perceptual mapping, this ensures that the output image is properly centred in the screen dynamic range. The

[7] In our case, we use a calibrated screen whose range is known.

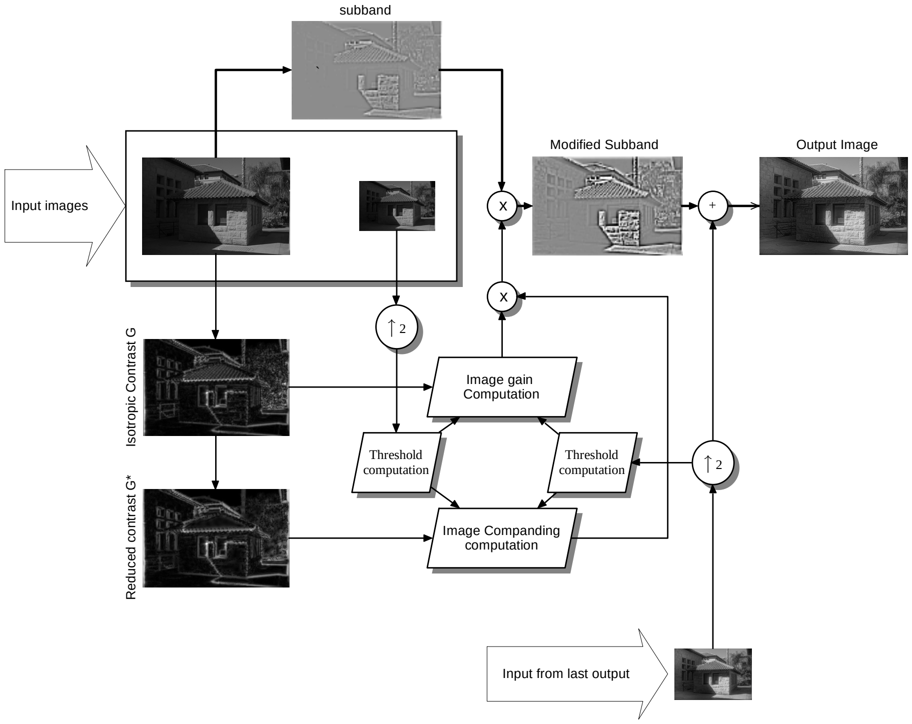


Figure 6.15: Rendering algorithm for one particular subband, using a modified contrast image to maintain the details of high contrast but of small size. The subband is multiplied by a gain image. The gain image is computed using the adaptation of the eye by viewing the original scene and the adaptation of the eye in the output conditions by assuming that the user watches the image on the computer screen. The gain image also involves a companding function, based on a modified contrast image, that reduces the contrast by maintaining the little details and affecting the important ones.

use of the 2% and 98% bounds has been preferred to the image amplitude bounds (0% and 100%) to avoid that a single noisy pixel can influence the value L_{mout}[8]. In practice, the image histogram shape changes, and there is some trial-and-error involved in the determination of L_{mout}. The value of $C_{\max}$ is found by iterating the rendering until the dynamic range of the image fits the output device.

If the dynamic range is too important, then the artifacts due to the companding function become disturbing. In practice, the companding factor C_{max} of Equation (6.18) is always bigger than 30. If the output dynamic range is still too big, several possibilities can be used: One is to clip the image, a solution that is not recommended if the clipped regions are bigger than a fraction of degree in size (no more than 0.2 deg). Another is to linearly reduce the perceived contrast, but this is unadvisable, since some details will be lost in the procedure. The remaining solution is to compress the highlights in the picture, as it is done by traditional rendering algorithm [Braun and Fairchild, 1999]. It is a way of introducing the decrease in contrast perception at very high luminances, a phenomenon that is not modelled by Equation (6.6). The transformation should reduce the highlights, but should leave the shadows and midtones unchanged. Let L^o be the image output by the algorithm, bounded by $[L^o_{\min}, L^o_{\max}]$ and $L_{S\max}$ be the maximum screen luminance; the final output L^f of the highlights compression is defined by

$$\begin{aligned} L^f &= L^o - \beta\,(L^o)^N - L^o_{\min} \\ \beta &\triangleq \frac{(L^o_{\max} - L^o_{\min}) - L_{S\max}}{(L^o_{\max} - L^o_{\min})^N} \\ N &\triangleq \frac{(L^o_{\max} - L^o_{\min})}{(L^o_{\max} - L^o_{\min}) - L_{\mathrm{extr}}}. \end{aligned} \tag{6.23}$$

This function ensures that the output L^f fits in the output device dynamic range (if $L_{\mathrm{extr}} \geq L_{S\max}$), and also that the darkest point in the image is black, which in practice is already true (i.e. $L^o_{\min}$ usually does not exceed 1% of $L^o_{\max}$). The function exhibits a slope equal to 1 at the origin, tends to follow the identity for small values, and has a maximum at $L^o = L_{\mathrm{extr}}$. L_{extr} controls the slope of the function at $L^o = L_{S\max}$, and in our implementation $L_{\mathrm{extr}} = 1.1 \cdot L_{S\max}$. A plot of Equation (6.23) is found in Figure 6.13(b). This function is applied once on the luminance information, and again on each colour channel in the final image, if these channels exceed the output device capability.

6.9.6 Remarks

The image that the algorithm outputs is expressed in units of luminance, i.e. in $\mathrm{Cd/m}^2$. In other words, the image has to be considered as *raw data*. To display it on a monitor, it is necessary to process the image in order to invert to transformation induced by the monitor. This kind of processing is fairly common and is described by the standards of digital imaging. We opted for the sRGB Standard [ITU-R Recommendation BT.709-3, 1998] to view the image, i.e. we applied the transformation defined in (5.3) after having transformed the image into the appropriate colour space.

By applying the companding function independently in each subband, the method shows some bleeding around regions of huge contrast. This is visible on Figure 6.16, where the

[8] Note that this does not mean that the 2% of the pixels in the extremity of the image will be saturated.

bright sunny area bleeds under the arches. Although we should mention that most of this bleeding is already present in the original image, due to bleeding in the sensor of the camera or due to the input flare. This bleeding phenomenon is known by the signal denoising community as the *pseudo-Gibbs effect* [Mallat, 1998, chap. 10]. An illustration of this bleeding effect can be found in a paper by DiCarlo and Wandell [DiCarlo and Wandell, 2000]. In the case of one dimensional signals, the problem may be solved using wavelet footprints [Dragotti and Vetterli, 2000], but to our knowledge, no solution has been found yet for two dimensional signals (edgeprints). A tempting solution is given by performing a companding of the image derivative, and then reconstruct the image from the derivative. This approach implies a regularisation of the derivative to recompute the image, for which we were unable to find a computationally tractable solution. This problem is extensively discussed in a book by Borre [Borre, 2001], who treats the problem of determining a terrain height from altitude difference measurements.

Flare dependence is introduced only when the dynamic range is not too important, like for example in Figure 6.17, although, we must admit that we did not experiment a lot with flare dependence, also because we were unable to exactly know the flare value of the output device (the standard define a mean value for the flare, but does not specify its spatial frequency). We assumed (by trial and error) that the flare has a gaussian spacial frequency distribution equal to

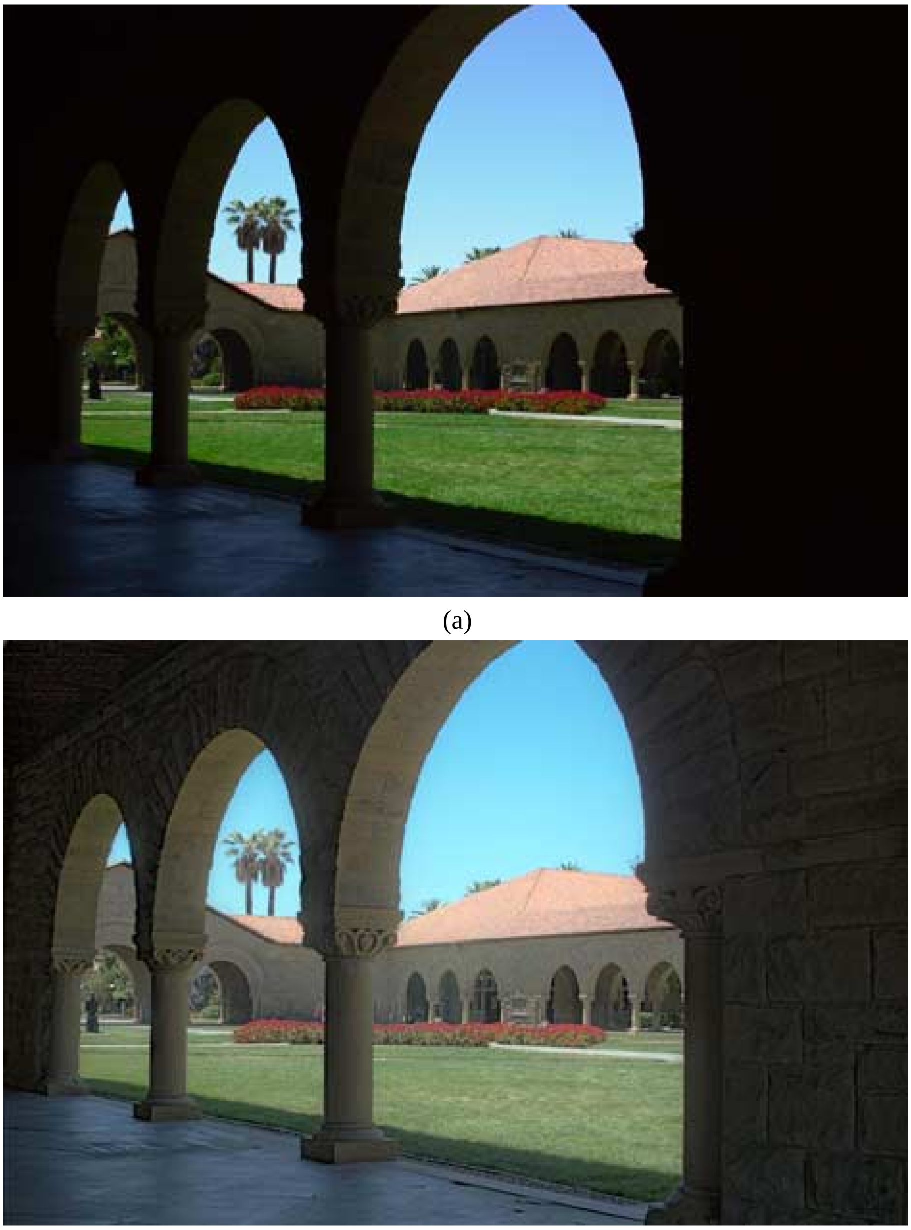
$$\begin{aligned} F(\omega) &= F_0 \cdot e^{-\frac{\omega}{2\sigma^2}} \\ \sigma &= 10^{-1}\,\mathrm{cycles}/\deg \end{aligned}$$

where F_0 is the fare value given by the standard used to render the image. To conclude, we would like to point out that the preferred equation for rendering is (6.21).

6.10 Results

The results of our algorithm are shown on figures 6.16 to 6.21. In general, it gave very satisfactory results, even if it was only applied to the luminance information, leaving the colours unchanged. The image in Figure 6.16 has been computed using several differently exposed pictures, whereas the image in figures 6.17 and 6.18 have been computed out of a single raw image of 12 bits of depth. In Figure 6.17, the part of the pictures that is very dark in the image generated by the standard algorithm shows a big amount of noise on the image processed by the our algorithm. This noise is caused by the limited dynamic range of the camera.

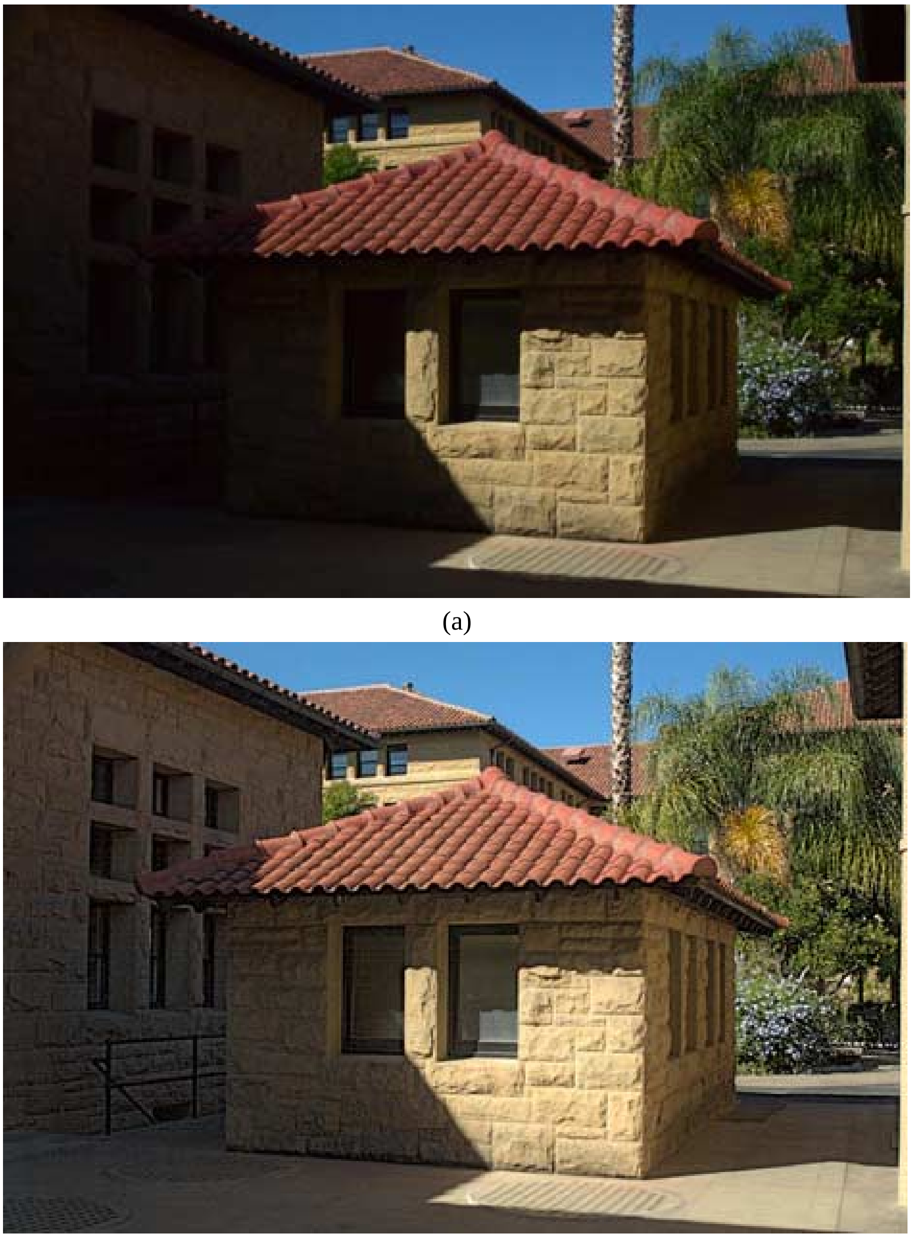
More specifically, Figure 6.19 compares the rendering using both the isotropic contrast measure and the derivative-based contrast measure. The derivative-based contrast measure exhibits less pseudo-Gibbs effect than the isotropic measure. Figure 6.20 shows the influence of adding a flare term to the rendering. We must admit that the flare parameter did not improve the image rendering in the experiments we conducted. Figure 6.22 shows the influence of the companding function on the image, by using two different values for the parameter $C_{\max}$ of Equation (6.18).



(a)

(b)

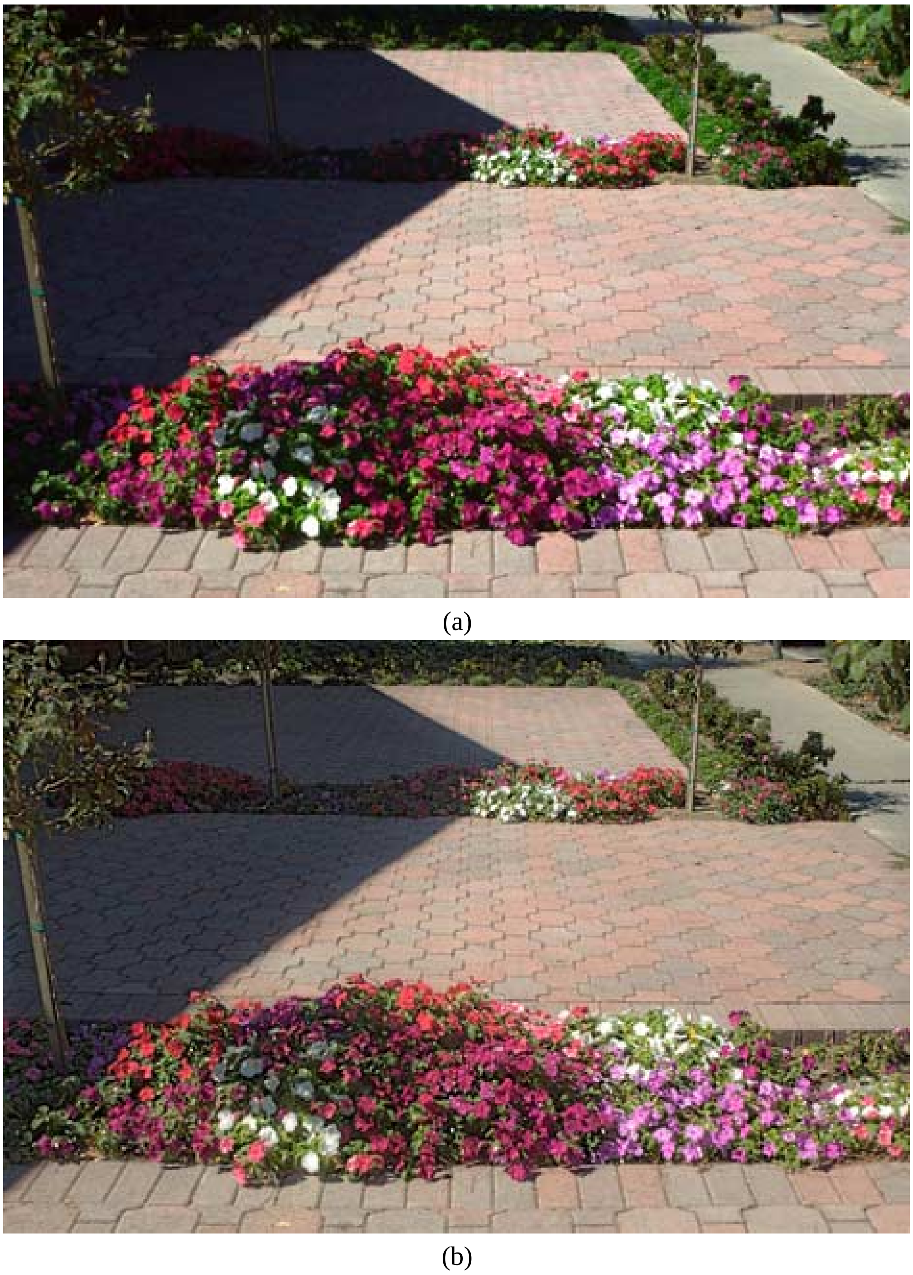
Figure 6.16: Output of the dynamic range compression algorithm: The details under the arches are now visible - provided that the output device is well calibrated. The image is closer to what the user sees of the scene (a) Original (b) Processed. The noise in the upper left corner of the image is caused by the acquisition system and is not due to the rendering algorithm. The blue cast under the arch is due to lens flare at the input.



(a)

(b)

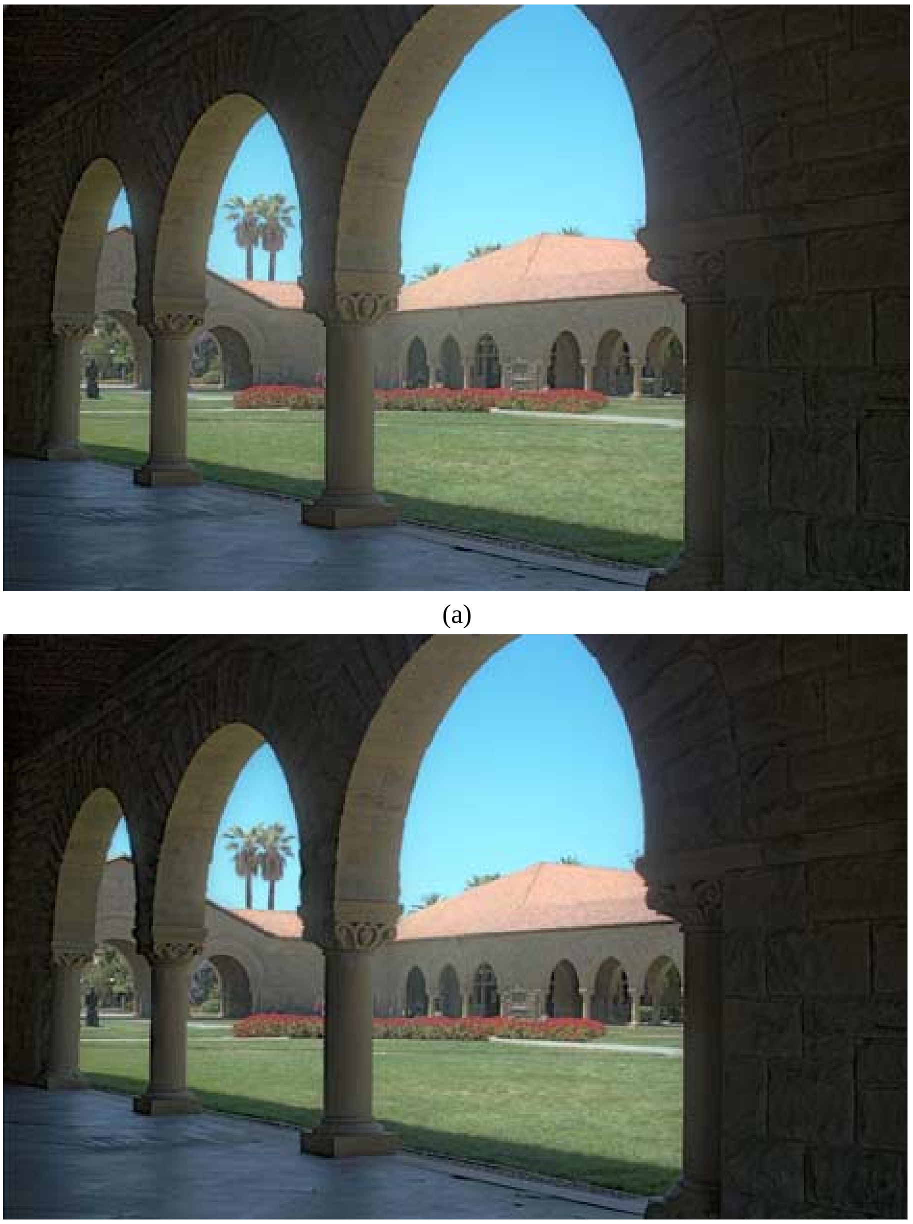
Figure 6.17: Output of the dynamic range compression algorithm (a) Original (b) Processed.



(a)

(b)

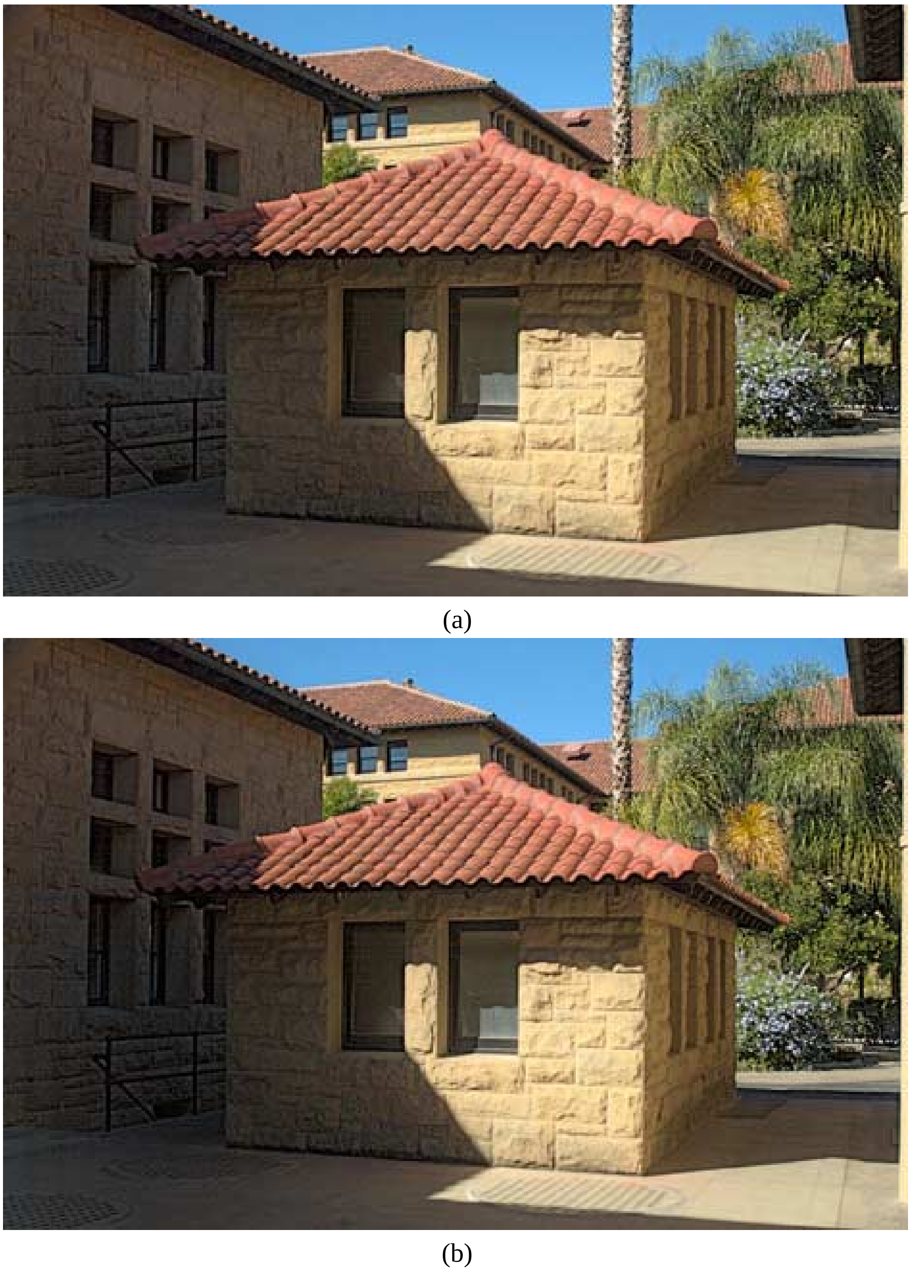
Figure 6.18: Output of the dynamic range compression algorithm (a) Original (b) Processed.



(a)

(b)

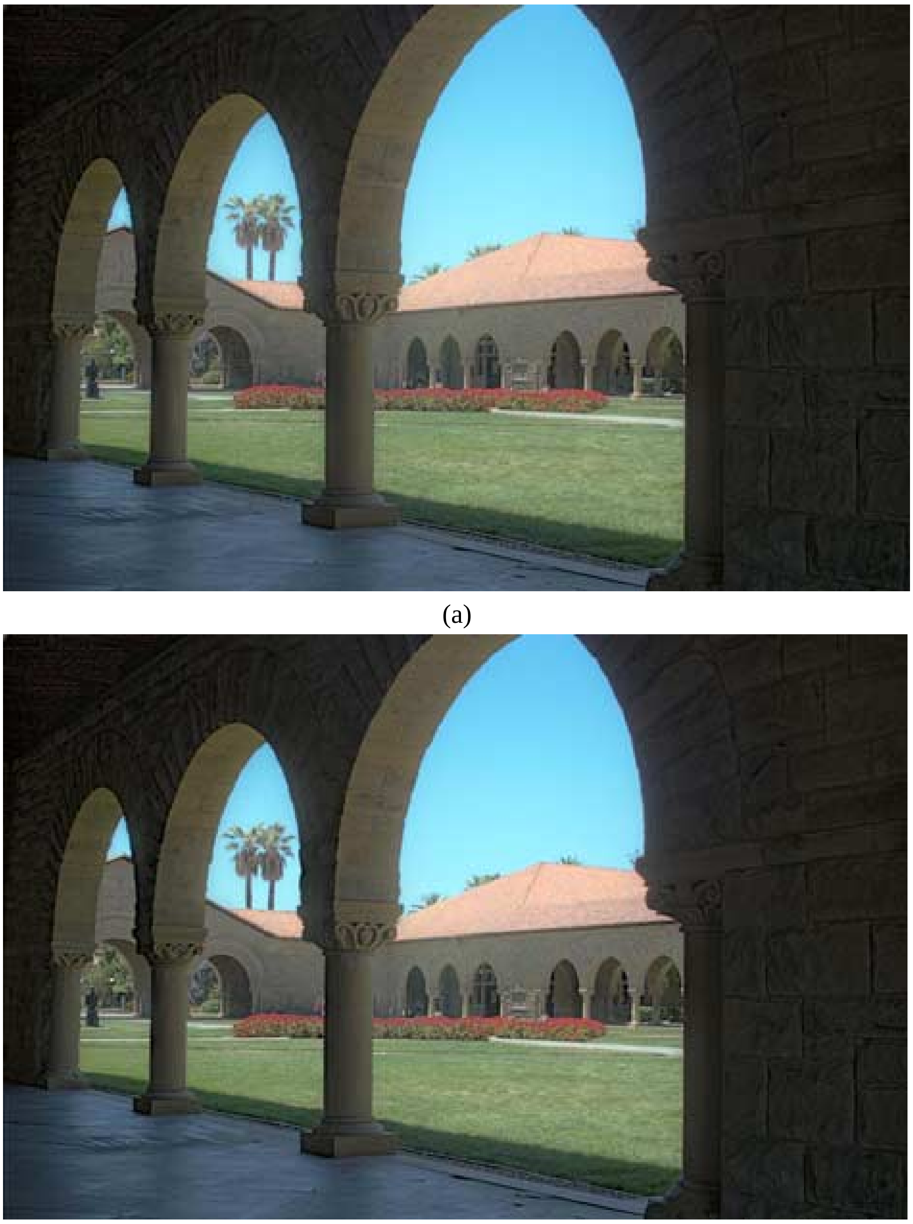
Figure 6.19: Differences in using two different contrast metrics (a) is rendered with the isotropic contrast measure (b) is rendered using the derivative-based contrast. The isotropic contrast delivers a hazy picture.



(a)

(b)

Figure 6.20: Effect of modelling viewing flare (a) is rendered with derivative-based contrast, no flare. (b) is rendered using the same parameters, but with a flare value of $5.57\ \mathrm{Cd/m^2}$. (b) exhibits more contrast in the dark regions, but the consequence is that the dark area appears darker



(a)

(b)

Figure 6.21: Effect of using contrast reduction to render the image (a) is rendered with derivative-based contrast (Equation 6.19). (b) is rendered using a contrast measure that has modified to avoid the companding of small areas (Equation 6.21). We should notice - if the quality of the reproduction allows it - that (b) is slightly sharper.

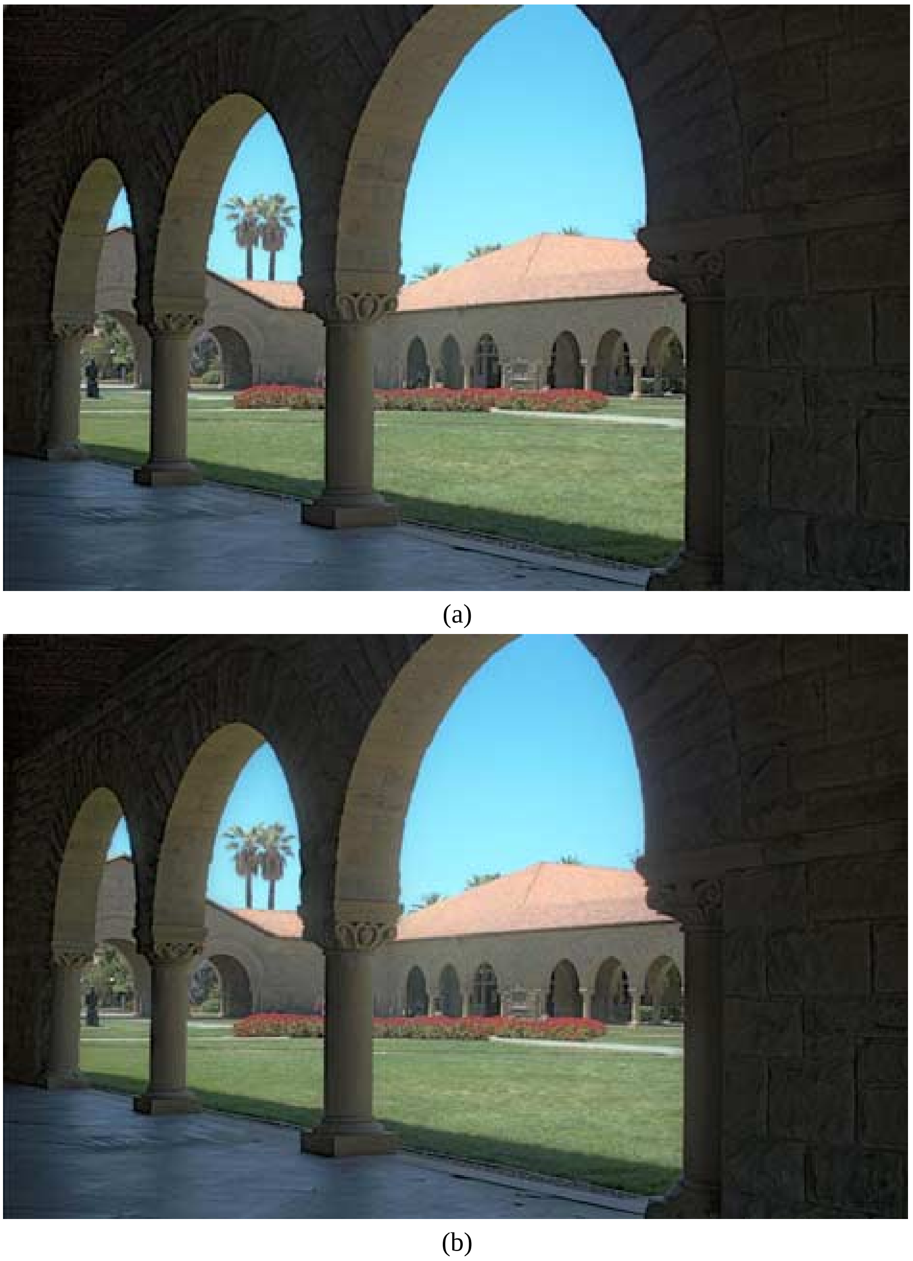


Figure 6.22: Effect of companding the image in perceptual space versus applying the companding on the final image. (a) Image rendered with $C_{\mathrm{max}} = 50$. (b) Image rendered with $C_{\mathrm{max}} = 30$. (b) looks a bit hazy and (a) shows some overshoot around sharp edges.

6.11 Remarks

Several assumptions have been made for designing the algorithm. The main assumption is that the contrast sensitivity of the eye has a linear behaviour across frequencies. We assumed that, given the contrast sensitivity at each frequency, we can apply the sensitivity function to each frequency component separately and add the results, i.e. we assumed that the sum of the sensitivity to each frequency component is equal to the sensitivity to the sum of the all frequency components together. This last assumption is probably wrong, nevertheless, in practice it seems to work quite well.

The contrast sensitivity model has been designed for photopic vision, and the images shown are also seen under photopic vision conditions. The model gives only approximate results for scotopic vision. This brings up the question of whether the image should reproduce the perception in these kind of situation, i.e, when rendering night pictures. In practice, the image 'looks better' if it reproduces more details than the one that are theoretically visible on site; this is accomplished by simply setting the input mean luminance of the image to an arbitrary high value. This also illustrates the limitation of the approach: All the efforts have, until now, been focused in reproducing what an observer sees in a scene, but we never asked ourselves what the observer would want to see in the image. If, for example, the illumination conditions are not ideal, then what the user wants to have on the picture is usually not what he sees in the field. In practice we found that taking the output of the algorithm and applying a simple curve to it, which emphasises either the highlights or the shadows, gives excellent results. The choice of the curve is left to the user.

6.12 Possible improvements

Several methods can be used to improve or complete the algorithm proposed in this chapter. Here is a small list of them:

- Extend the vision model to the chromatic channels. This is possible for the luminance levels of a CRT screen, but would require new measurements of high luminance dependance.
- Change the method of companding the contrast such that is does not produce any pseudo-Gibbs artifact.
- Measure the spatial frequency adaptation of the pupil size.
- Use a model for the scattering of the light in the eye at very high luminance ($> 5000\mathrm{cd}/\mathrm{m}^2$) to compute the input viewing flare, or -equivalently - introduce the decrease in contrast sensitivity at very high luminance in the contrast perception model.
- Render the scattering of the light in the eye on the final image, as was done by Spencer [Spencer et al., 1995].

6.13 Conclusion

In this chapter, a new algorithm for rendering of high dynamic range images has been presented. It is based on a complex model of the human visual system. Local perceived contrast values are calculated using the contrast sensitivity and contrast discrimination function are maintained as best as possible. If the dynamic range of a local area of a scene is too high to be displayed on a particular output device, an adequate compression is applied maintaining the small details and compressing the large contrast values. The compression functions depend on the actual image and on the spatial information. The algorithm also considers the different viewing conditions between perceiving an outdoor scene and perceiving an image of a scene on a monitor. Our work is based on work by Pattanaik [Pattanaik et al., 1998], but has further been improved by using a more complete model of the human luminance contrast perception. Images presented in this chapter show that the results are very encouraging. Extension to this work should be concerned with the reduction of the Pseudo-Gibbs-Effect, which can occur in particular situations and the inclusion of chromatic contrast sensitivity functions.

Chapter 7

Conclusions

This thesis addresses several aspects of image stitching to create a seamless panorama from several pictures. It covers one aspect of parallax correction, global estimation techniques, colour equalisation across pictures, outlier modelling and detection, and finally the rendering of high dynamic range images.

Chapter 3 presents a technique that enable to estimate the three dimensional structure of a scene, by assuming the scene is composed of piece-wise planar surfaces.This model enables to reconstruct an picture from a set of images that contain parallax, a phenomenon that arises when the pictures are not taken from the same location. An example showing “Le Louvre” in Paris is reconstructed using this technique: The floor and the walls of the building are assigned to different planes in space. In general, the technique is computationally very expensive, and lacks of some robustness. There is a fundamental reason for the lack of robustness that is related to the family of method used to perform the estimation. This argument is developed in Chapter 4. The method innovates by introducing shared 3D parameters on a problem using mixtures of models that were, until now, solely based on 2D relationships.

Chapter 3 also presents a method to close the gap in a panorama. The gap occurs when stitching the images of a full panoramic image - an image that covers an angle of 360 deg - between the last and the first picture of the panorama. This gap is due to the accumulation of little errors in the stitching that add up to a large absolute error in the last picture location. The problem is solved with a linear system of equations, and has shown to work well. More generally, the method enables to get from an image-to-image relationship problem toward a global estimation of the mosaic. Solving the image-to-image relationship and solving the mosaic as a whole are standard problems; this method enables to compute a good initial estimate of the global problem given the initial image-to-image relationships. The method can be applied whenever there is a circular relationship among the picture in an image set.

Chapter 4 exposes a method to deal with outliers in image pairs. An outlier is a portion of image that is not consistent in the image pair. Usually, it is an object (or a person) that appears in one image and that is absent in the other, because it simply moved away. The novelty of the approach lies in the outlier distribution characterisation that is assumed to be the result of comparing pixels picked randomly in the two images. By “ outlier distribution ” we refer to the error distribution in a pixel-wise comparison of the part of the image pair that contains the outlier. From this new concept, the outlier detection problem can be formulated as a standard mixture of models problem: the inliers and the outliers, both characterized by a distribution. The performance of the model is demonstrated using two different types of experiments. The first experiment shows two aligned pictures, one of them containing an outlier. The proportion of the image covered by the outlier is varied by framing differently the images of the pair, and the goal is to see when the model is not able to distinguish the outlier from the inlier anymore. In this example, our model showed to be able to handle an arbitrary amount of outlying pixels. The standard methods that base their computation on a median operator tend to fail whenever the outlier proportion exceeds 50% of the image. The second experiment compares the motion estimator derived from the outlier model - which is an M-type estimator - to a standard robust M-estimator. Here, again, the outlier-based model clearly outperforms the standard estimator and has a computational complexity that is very close to the standard model.

Chapter 5 explains how to equalize the colours across the pictures of a panorama. To get consistent colours, the algorithm inverses the transformation introduced by the camera, equalizes the pictures, and renders the mosaic as a single image in the end. When the transformation of the camera is undone, the relationship among the pictures can be modelled by a linear transformation that considerably simplifies the equalisation problem. Different models are presented to reverse the camera transforms, and an example is shown to deal with scanned analog transparencies. In the analog case, because of the large number of transforms that occur between the light entering the camera and the final picture, the process is reversed using a calibration procedure and the film characteristics (that is publicly available for most of the films on the market). Examples of pictures taken with a digital as well as with an analog camera are shown and give very satisfactory results. In addition, the colour equalisation enables a better geometric registration of the pictures. The method fails when there is an illumination change in the scene. Very little has been published in the field so far.

Chapter 6 addresses the issue of rendering pictures of high dynamic range. These pictures contain very bright and very dark areas, and the ratio of luminance between the very bright and very dark area is (much) bigger than the ratio of luminance of a white and a black spot on a computer screen. A complex model of the human visual system, based on contrast perception is used to render the image. On the one hand, the perceptual model tells how to modify the contrast of the image such that the perception of the contrast in the image on the screen is the same than when looking at the real scene. On the other hand, the algorithm proposes a way of compressing the dynamic range of the image in such a way that is affects the big contrast regions, but leaves the tiny details unchanged. The method showed excellent results for middle-to-high dynamic range scenes (bright sunlight with shade areas), but shows some artifacts with very high dynamic range scenes (bright sunlight and interiors), that come from applying a non-linearity in the contrast compression. The method differs from previous ones by using a more complex measure of contrast that has been shown to be more appropriate in natural images. It also explicitly defines a way of compressing the contrast which is consistent with the perceptual model. In the literature, one can find methods that use a model of perception to compute how an image should ideally be rendered on the screen, but are very vague in specifying how to compress the contrast in order to bring the image in the displayable range of the screen. One can also find methods to compress the dynamic range of an image to bring it down to the displaying capabilities of the screen, but these method do not specify what effect this compression will have on the final image perception, even if they are all based on some properties of the human visual system.

Chapters 4, 6 and 5 can be merged in one consistent way of building a mosaic. The equalisation of colours technique brings the images in linear space, and compensate for changes in the camera settings. The outlier detection and robust comparison of the images has to be performed in linear space and also enables a better colour equalisation and registration. Finally, the image is transformed from linear space toward a displayable image on the screen using the technique described in the chapter about rendering. In general, all the method described in the computer vision literature about stitching should be applied in linear space. The alignment of a 360 deg panorama, as well as the segmentation of the scene into planes are less intimately related to the other subjects, although the multi-planar model showed to work much better when applied in linear space.

As a conclusion, we can say that the methods exposed in this thesis allow to build a very high quality picture out of images taken with variable exposure parameters, and containing big luminance changes from bright to dark areas.

Appendix A

Motion estimation equations

In this appendix, the details of the motion estimation equation (2.26) are developed. It starts by settling the notations. Then, the general motion estimation algorithm is developed, followed by a the specialization of the general model to the formulation of Chapter 2, which handles changes in the camera settings.

A.1 Definitions

Let $\mathbf{r}$ be a vector with N components, θ a vector with M components and $f(\cdot)$ be a scalar function, which takes a vector $\mathbf{r}$ as input parameter. The gradient of $f(\cdot)$ with respect to its component is denoted as:

$$\nabla f \triangleq \frac{\partial f(\mathbf{r})}{\partial \mathbf{r}} \triangleq \left[\frac{\partial f(\mathbf{r})}{\partial r_1}, ..., \frac{\partial f(\mathbf{r})}{\partial r_N}\right],$$

which is a matrix of size $1 \times N$ (i.e. a line vector).

The transposed gradient, is written as

$$\nabla f^T \triangleq \frac{\partial f(\mathbf{r})}{\partial \mathbf{r}}^T \triangleq \frac{\partial f(\mathbf{r})}{\partial \mathbf{r}^T} \triangleq \left[\frac{\partial f(\mathbf{r})}{\partial r_1}, ..., \frac{\partial f(\mathbf{r})}{\partial r_N}\right]^T,$$

which is a vector of size $N \times 1$ (i.e. a column vector). Now, if $\mathbf{r}$ is a function of $\theta : \mathbf{r} \triangleq \mathbf{r}(\theta)$, the chain rule for derivatives is

$$\frac{\partial f(\mathbf{r})}{\partial \theta} = \frac{\partial f(\mathbf{r})}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \theta}$$

where $\frac{\partial \mathbf{r}}{\partial \theta}$ is a Jacobian matrix of size $N \times M$, which is defined by

$$\left[\frac{\partial \mathbf{r}(\theta)}{\partial \theta}\right]_{i,j} = \frac{\partial r_i(\theta)}{\partial \theta_j},$$

where $[\cdot]_{i,j}$ denotes the component at the i-th row and j-th column. Conversely, when the vectors are transposed, the chain rule becomes

$$\frac{\partial f(\mathbf{r})}{\partial \theta^T} = \frac{\partial \mathbf{r}}{\partial \theta}^T \frac{\partial f(\mathbf{r})}{\partial \mathbf{r}}^T.$$

A.2 General motion estimation computation

We want to solve Equation (2.25):

$$\mathbf{J}(\theta)^T + \mathbf{H}(\theta)\delta\theta = \mathbf{0},$$

where

$$\begin{aligned}\mathbf{H}(\theta) &= \frac{\partial^2 h(\mathbf{r},\theta)}{\partial\theta^T\partial\theta}\\ \mathbf{J}(\theta) &= \frac{\partial h(\mathbf{r},\theta)}{\partial\theta}.\end{aligned}$$

The objective function h, which is written as $h(\mathbf{r},\theta)$, $h(\mathbf{r})$ or $h(\theta)$, is defined as

$$h(\mathbf{r}) = \frac{1}{N}\sum_{i=1}^{N} W_i \cdot \rho[r_i(\theta)]$$

where N is the number of pixel on the image overlap, r_i the error value of the pixel pair at location i and $\rho(\cdot)$ a function defined in Chapter 4. W_i is a weight factor defined in Chapter 5, we further assume

$$\frac{\partial W_i}{\partial\theta} \simeq 0.$$

The Jacobian is computed using

$$\begin{aligned}\mathbf{J}(\theta) &\triangleq \frac{\partial h(\mathbf{r})}{\partial\theta}\\ &= \frac{\partial h(\mathbf{r})}{\partial\mathbf{r}}\frac{\partial\mathbf{r}}{\partial\theta}.\end{aligned}$$

The Hessian is computed using

$$\begin{aligned}\mathbf{H}(\theta) &= \frac{\partial}{\partial\theta^T}\left\{\frac{\partial h(\mathbf{r})}{\partial\mathbf{r}}\frac{\partial\mathbf{r}}{\partial\theta}\right\}\\ &= \frac{\partial^2 h(\mathbf{r})}{\partial\theta^T\partial\mathbf{r}}\frac{\partial\mathbf{r}}{\partial\theta} + \frac{\partial h(\mathbf{r})}{\partial\mathbf{r}}\frac{\partial^2\mathbf{r}}{\partial\theta^T\partial\theta}\\ &= \left(\frac{\partial\mathbf{r}}{\partial\theta}\right)^T\frac{\partial^2 h(\mathbf{r})}{\partial\mathbf{r}^T\partial\mathbf{r}}\left(\frac{\partial\mathbf{r}}{\partial\theta}\right) + \frac{\partial h(\mathbf{r})}{\partial\mathbf{r}}\frac{\partial^2\mathbf{r}}{\partial\theta^T\partial\theta}\\ &\simeq \left(\frac{\partial\mathbf{r}}{\partial\theta}\right)^T \operatorname{diag}_i\left[W_i\cdot\ddot{\rho}(r_i)\right]\left(\frac{\partial\mathbf{r}}{\partial\theta}\right) + \mathbf{0},\end{aligned}$$

where

$$\operatorname{diag}_i\left[W_i\cdot\ddot{\rho}(r_i)\right] \triangleq \operatorname{diag}\left[W_1\frac{\partial^2\rho(r_1)}{\partial r_1^2},...,W_N\frac{\partial^2\rho(r_N)}{\partial r_N^2}\right].$$

The term $\frac{\partial h(\mathbf{r})}{\partial\mathbf{r}}\frac{\partial^2\mathbf{r}}{\partial\theta^T\partial\theta}$ has been neglected in the computation. The term $\frac{\partial^2 h(\mathbf{r})}{\partial\mathbf{r}^T\partial\mathbf{r}}$ has only diagonal elements, because by assumption, there are no inter-dependence between neighboring pixels. The Hessian is further approximated by using a secant approximation of the second derivative:

$$\ddot{\rho}(r_i) \simeq \frac{1}{r_i}\frac{\partial \rho(r_i)}{\partial r_i} \triangleq \frac{\dot{\rho}(r_i)}{r_i},$$

in order to get a positive definite matrix. Finally, the Hessian becomes

$$\mathbf{H}(\theta) \simeq \left(\frac{\partial \mathbf{r}}{\partial \theta}\right)^T \operatorname{diag}_i \left[W_i \cdot \frac{\dot{\rho}(r_i)}{r_i}\right] \left(\frac{\partial \mathbf{r}}{\partial \theta}\right). \tag{A.1}$$

Now, let us expand the motion estimation equation (2.25)

$$\mathbf{H}(\theta)\delta\theta = -\mathbf{J}(\theta)^{\mathbf{T}},$$

which becomes:

$$\left(\frac{\partial \mathbf{r}}{\partial \theta}\right)^T \operatorname{diag}_i \left[W_i \cdot \frac{\dot{\rho}(r_i)}{r_i}\right] \left(\frac{\partial \mathbf{r}}{\partial \theta}\right) \delta\theta = -\left(\frac{\partial \mathbf{r}}{\partial \theta}\right)^T \frac{\partial h(\mathbf{r})}{\partial \mathbf{r}}.$$

Then setting $\frac{\partial h(\mathbf{r})}{\partial \mathbf{r}} = \operatorname{diag}_i \left[W_i \cdot \frac{\dot{\rho}(r_i)}{r_i}\right] \cdot \mathbf{r}$, gives

$$\left(\frac{\partial \mathbf{r}}{\partial \theta}\right)^T \operatorname{diag}_i \left[W_i \cdot \frac{\dot{\rho}(r_i)}{r_i}\right] \left(\frac{\partial \mathbf{r}}{\partial \theta}\right) \delta\theta = -\left(\frac{\partial \mathbf{r}}{\partial \theta}\right)^T \operatorname{diag}_i \left[W_i \cdot \frac{\dot{\rho}(r_i)}{r_i}\right] \cdot \mathbf{r}. \tag{A.2}$$

Equation (A.2) is of the form $\mathbf{A}^T\mathbf{W}\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{W}\mathbf{b}$, which corresponds to the weighted least square solution of a classical linear system given by $\mathbf{W}\mathbf{A}\mathbf{x} = \mathbf{W}\mathbf{b}$, or

$$\operatorname{diag}_i \left[W_i \cdot \frac{\dot{\rho}(r_i)}{r_i}\right] \left(\frac{\partial \mathbf{r}}{\partial \theta}\right) \delta\theta = -\operatorname{diag}_i \left[W_i \cdot \frac{\dot{\rho}(r_i)}{r_i}\right] \cdot \mathbf{r}.$$

Rewriting the i-th line of this equation system leads to

$$W_i \cdot \frac{\dot{\rho}(r_i)}{r_i}\frac{\partial r_i}{\partial \theta}\delta\theta = -W_i \cdot \frac{\dot{\rho}(r_i)}{r_i} \cdot r_i, \tag{A.3}$$

which is the system that is solved using a minimum mean square error criterion at each iteration of the motion estimation algorithm.

A.3 Specialisation to Colour Images

To get more specific, we can further develop (A.3), according to the formulation presented in Chapter 2. The weighting term $\frac{\dot{\rho}(r_i)}{r_i}$ is defined in Chapter 4. The residual is defined in Chapter 2, by (2.21):

$$r_i(\theta) \triangleq \{G_\theta\left[I_0(V_\theta\left[\mathbf{p}_i\right])\right] - S_\theta\left[I_1(U_\theta\left[\mathbf{p}_i\right])\right]\}.$$

The derivative of the residual gives

$$\begin{aligned}
\frac{\partial r_i}{\partial \theta} &= \frac{\partial}{\partial \theta}\{G_\theta\left[I_0(\mathbf{V}_{\theta i})\right] - S_\theta\left[I_1(\mathbf{U}_{\theta i})\right]\} \\
&= \frac{\partial G_\theta\left[I_0(\mathbf{V}_{\theta i})\right]}{\partial \theta} - \frac{\partial S_\theta\left[I_1(\mathbf{U}_{\theta i})\right]}{\partial \theta} \\
&= \nabla G_\theta\left[I_0(\mathbf{V}_{\theta i})\right]\frac{\partial \mathbf{V}_{\theta i}}{\partial \theta} + \dot{G}_\theta\left[I_0(\mathbf{V}_{\theta i})\right] - \nabla S_\theta\left[I_1(\mathbf{U}_{\theta i})\right]\frac{\partial \mathbf{U}_{\theta i}}{\partial \theta} - \dot{S}_\theta\left[I_1(\mathbf{U}_{\theta i})\right],
\end{aligned}$$

where, by definition,

$$\begin{aligned}
\nabla G_\theta\left[I_0(\mathbf{V}_{\theta i})\right] &\triangleq \left.\frac{\partial G_{\mathbf{u}}\left[I_0(\mathbf{x})\right]}{\partial \mathbf{x}}\right|_{\mathbf{u}=\theta,\mathbf{x}=\mathbf{V}_\theta[\mathbf{p}_i]} \\
\dot{G}_\theta\left[I_0(\mathbf{V}_{\theta i})\right] &\triangleq \left.\frac{\partial G_{\mathbf{u}}(v)}{\partial \mathbf{u}}\right|_{\mathbf{u}=\theta,v=I_0(\mathbf{V}_{\theta i})} \\
\mathbf{U}_{\theta i} &\triangleq \mathbf{U}_\theta\left[\mathbf{p}_i\right] \\
\mathbf{V}_{\theta i} &\triangleq \mathbf{V}_\theta\left[\mathbf{p}_i\right]
\end{aligned}$$

In other words, $\nabla G_\theta\left[I_0(\mathbf{V}_{\theta i})\right]$ is the derivative along the lines and the columns of the radiance image computed from an undistorted version of image I_0. $\nabla S_\theta\left[I_1(\mathbf{U}_{\theta i})\right]$ is the derivative along the lines and the columns of the radiance image that has been warped according to the motion function $\mathbf{U}$. $\dot{G}_\theta\left[I_0(\mathbf{V}_{\theta i})\right]$ and $\dot{S}_\theta\left[I_1(\mathbf{U}_{\theta i})\right]$ are the derivative of the camera characteristics with respect to the parameters that control the characteristics (which might be different for image I_0 and I_1). Finally, we get to the desired equation:

$$\theta_{i+1} = \theta_i + \alpha\delta\theta$$

$$W_i \cdot \frac{\dot{\rho}(r_i)}{r_i}\left\{\begin{array}{c} \nabla G_\theta\left[I_0(\mathbf{V}_{\theta i})\right]\frac{\partial \mathbf{V}_{\theta i}}{\partial\theta} + \dot{G}_\theta\left[I_0(\mathbf{V}_{\theta i})\right] - \\ -\nabla S_\theta\left[I_1(\mathbf{U}_{\theta i})\right]\frac{\partial \mathbf{U}_{\theta i}}{\partial\theta} - \dot{S}_\theta\left[I_1(\mathbf{U}_{\theta i})\right] \end{array}\right\}\delta\theta = -W_i \cdot \frac{\dot{\rho}(r_i)}{r_i}\cdot r_i \quad \blacksquare$$

Appendix B

Outlier model computations

This appendix details the assumptions involved in the OutlierMix model of Chapter 4.

Let Θ_0 and Θ_1 be two independent discrete random variables. Let I_0 and I_1 be two injective functions (I_0 and I_1 are the images that are compared).

We are interested in the probability of error $\mathrm{P}(r)$ when comparing two images:

$$\mathrm{P}(r) = \Pr\left\{I_0\left(\theta_0, \mathbf{p}\right) - I_1\left(\theta_1, \mathbf{p}\right) = r\right\},$$

where $\mathbf{p}$ is the position in the image ($\mathbf{p}$ is not random), and r the error (or *residual*). Let H be the error histogram

$$H\left(r, \theta_0, \theta_1\right) \triangleq \frac{1}{M} \sum_{\forall \mathbf{p}} 1_{\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\}},$$

where M is the number of pixels and $1_{\{\cdot\}}$ is the indicator function ($1_{\{b\}}$ is equal to 1 if b is true, 0 otherwise). The expectation (E) of the error histogram is equal to

$$E_{\Theta_0, \Theta_1}\left[H\left(r, \Theta_0, \Theta_1\right)\right] = E_{\Theta_0, \Theta_1}\left[\frac{1}{M} \sum_{\forall \mathbf{p}} 1_{\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\}}\right].$$

Since the expectation is a linear operator and the sum is finite, we can switch the expectation and the sum:

$$E_{\Theta_0, \Theta_1}\left[H\left(r, \Theta_0, \Theta_1\right)\right] = \frac{1}{M} \sum_{\forall \mathbf{p}} E_{\Theta_0, \Theta_1}\left[1_{\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\}}\right].$$

The expectation can be expressed as

$$E_{\Theta_0, \Theta_1}\left[1_{\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\}}\right] = \sum_{\forall \theta_0} \sum_{\forall \theta_1} 1_{\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\}} \mathrm{P}\left(\theta_0, \theta_1\right),$$

but the indicator function of the error can be expressed as a cross-correlation,

$$1_{\{I_0(\theta_0, \mathbf{p}) - I_1(\theta_1, \mathbf{p}) = r\}} = \sum_{\forall u} 1_{\{I_0(\theta_0, \mathbf{p}) = u\}} 1_{\{I_1(\theta_1, \mathbf{p}) = u - r\}},$$

hence, the expectation is expressed as

$$\begin{aligned}
& E_{\Theta_0,\Theta_1}\left[1_{\{I_0(\theta_0,\mathbf{p})-I_1(\theta_1,\mathbf{p})=r\}}\right] = \\
= \; & \sum_{\forall\theta_0}\sum_{\forall\theta_1}\mathrm{P}\left(\theta_0,\theta_1\right)\sum_{\forall u}1_{\{I_0(\theta_0,\mathbf{p})=u\}}1_{\{I_1(\theta_1,\mathbf{p})=u-r\}} \\
= \; & \sum_{\forall u}\sum_{\forall\theta_0}\sum_{\forall\theta_1}\mathrm{P}\left(\theta_0,\theta_1\right)1_{\{I_0(\theta_0,\mathbf{p})=u\}}1_{\{I_1(\theta_1,\mathbf{p})=u-r\}},
\end{aligned}$$

by assuming that every sum is finite.

Because Θ_0 and Θ_1 are independent, we can separate the probabilities of Θ_0 and Θ_1:

$$\mathrm{P}\left(\theta_0,\theta_1\right) = \mathrm{P}\left(\theta_0\right)\mathrm{P}\left(\theta_1\right),$$

thus, the expectation can be expressed as a cross-correlation:

$$\begin{aligned}
& E_{\Theta_0,\Theta_1}\left[1_{\{I_0(\theta_0,\mathbf{p})-I_1(\theta_1,\mathbf{p})=r\}}\right] = \\
= \; & \sum_{\forall u}\left[\sum_{\forall\theta_0}\mathrm{P}\left(\theta_0\right)1_{\{I_0(\theta_0,\mathbf{p})=u\}}\sum_{\forall\theta_1}\mathrm{P}\left(\theta_1\right)1_{\{I_1(\theta_1,\mathbf{p})=u-r\}}\right], \\
= \; & \sum_{\forall u}\left[\Pr\left\{I_0\left(\theta_0,\mathbf{p}\right)=u\right\}\cdot\Pr\left\{I_1\left(\theta_1,\mathbf{p}\right)=r-u\right\}\right].
\end{aligned}$$

Finally,

$$E_{\Theta_0,\Theta_1}\left[H\left(r,\Theta_0,\Theta_1\right)\right] = \frac{1}{M}\sum_{\forall\mathbf{p}}\sum_{\forall u}\left[\Pr\left\{I_0\left(\theta_0,\mathbf{p}\right)=u\right\}\cdot\Pr\left\{I_1\left(\theta_1,\mathbf{p}\right)=r-u\right\}\right].$$

Now, by assuming that the statistics of the pixel in an image does not depend on the position $(\mathbf{p})$ where the pixel is located in the image, the expectation of the error histogram is equal to the cross-correlation of the pixel value distribution:

$$E_{\Theta_0,\Theta_1}\left[H\left(r,\Theta_0,\Theta_1\right)\right] = \sum_{\forall u}\left[\Pr\left\{I_0\left(\theta_0\right)=u\right\}\cdot\Pr\left\{I_1\left(\theta_1\right)=r-u\right\}\right].$$

By further assuming that the probability density function of the pixel is given by the image histogram

$$\frac{1}{M}\sum_{\forall\mathbf{p}}1_{\{I_0(\theta_0,\mathbf{p})=u\}} = \Pr\left\{I_0\left(\theta_0\right)=u\right\},$$

the expectation of the error histogram (H) is equal to the cross-correlation of the histograms (H_0, H_1) of the images:

$$E_{\Theta_0,\Theta_1}\left[H\left(r,\Theta_0,\Theta_1\right)\right] = \sum_{\forall u}H_0\left(u\right)\cdot H_1\left(r-u\right).$$

Finally, by assuming that the comparison process is constant (or that the error histogram does not depend on the image positioning), we get

$$E_{\Theta_0,\Theta_1}\left[H\left(r,\Theta_0,\Theta_1\right)\right] = H\left(r,\theta_0,\theta_1\right), \quad \forall\theta_0,\theta_1, \tag{B.1}$$

which finally leads to

$$H\left(r,\theta_0,\theta_1\right)=\sum_{\forall u}H_0\left(u\right)\cdot H_1\left(r-u\right).$$

The error histogram is equal to the cross-correlation of the image histograms. In Practice, if the images contain large uniform areas, Equation (B.1) tends to be approximate.

Appendix C

Contrast computation details

C.1 Angular frequency response

The angular frequency response $\eta(\varphi)$ is computed as a combination of Schwarz's functions. We summarise the construction proposed by [Vandergheynst and Gobbers, 2001]. Let us consider the $\mathcal{C}^\infty$, compactly supported function

$$g_a(\varphi) = \begin{cases} e^{-\frac{1}{2}\frac{a^2}{a^2-\varphi^2}}, & \text{if } -a < \varphi < a \\ 0, & \text{otherwise.} \end{cases}$$

Let $d_a(\varphi)$ be the periodisation of g_a

$$d_a(\varphi) = \sum_{n\in\mathbb{Z}} g_a(\varphi - na).$$

Note that, because of the way g_a is defined, this sum is composed of at most two non-zero terms. Let us define

$$w_a(\varphi) = \frac{g_a(\varphi)}{d_a(\varphi)},$$

a function who satisfies

$$\sum_{n\in\mathbb{Z}} w_a(\varphi - na) = 1, \quad \forall\varphi \in \mathbb{R}.$$

Finally, by choosing to subdivide the complex plane into K angular regions, with $K > 2$, the angular frequency response is given by

$$\eta(\varphi) = \sum_{n\in\mathbb{Z}} w_{\pi/K}(\varphi - 2n\pi)$$

and satisfies

$$\sum_{k=0}^{K-1} \eta(\varphi - k\pi/K) = 1.$$

C.2 Perceived contrast computation

Here are some numerical implementation details about the perceived contrast integral of Section 6.7. The perceived contrast integral has to be computed numerically, and is implemented with a $2D$ lookup table. The equations of the integral are reproduced here for convenience:

$$\triangle C_t(m) = \sqrt{\frac{C_t^2 + k^2 C_n^2 + 0.04\ k^2 m^2}{1 + 0.004\ k\ m/C_t} + m^2} - m$$

$$C^{\psi}(m) = a \cdot \int_{m_0=0}^{m} \frac{1}{\triangle C_t(m_0)} \cdot dm_0.$$

where m is the current contrast, C_t is the contrast threshold in the current viewing conditions and $\triangle C_t$ is the amount of contrast that has to be added to m in order that an observer notices a difference. C_n is the average modulation of the noise component, i.e. the standard deviation of the noise at the considered spatial frequency. C^{ψ} is the perceived contrast.

By pre-computing the integral for $C_n = \{0, 0.1, 0.2, ..., 0.9, 1\}$ and using a lookup table, we guarantee that the resulting error of a linear interpolation in the lookup table is less than 1%. This holds true also when computing the inverse function; In other words, we compute a lookup table $C^{\psi}(C_n, m)$ with 11 lines and N columns, and we perform the computation of the perceived contrast by linearly interpolating C^{ψ} as if it was an image. We do the same for the inverse function.

A possible approximation of the perceived contrast with flare is given by

$$C^{\psi}(C_n, m) \simeq C^{\psi}(0, m) - C_n.$$

The error introduced by this approximation is of the order of $2\% - 3\%$ for large (perceived) contrast values (> 5) and may be around 20% to 30% for small contrast values.

Bibliography

Bruno Aiazzi, Luciano Alparone, and Stefano Baronti. Estimation based on entropy matching for generalized gaussian pdf modeling. *IEEE Signal Processing Letters*, 6(6):138–140, 1999.

Serge Ayer. *Sequential and Competitive Methods for Estimation of Multiple Motions*. PhD thesis, Swiss Federal Institute of Technology (EPFL), 1995.

Simon Baker, Richard Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *CVPR*, pages 434–441, 1998.

A.R. Barron and T.M. Cover. Minimum complexity density estimation. *IEEE Transactions on Information Theory*, 37(4):1034 –1054, 1991.

Peter G.J. Barten. *Contrast Sensitivity of the Human Eye and its Effects on Image Quality*. SPIE Optical Engineering Press, 1999.

S. Beucher and F. Meyer. The morphological approach to segmentation: The watershed transformation. *Mathematical Morphology in Image Processing*, pages 433–481, 1993.

M.J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.

Kai Borre. *Plane networks and their applications*. Birkhauser, Boston, 2001.

Victor L. Brailovsky. An approach to outlier detection based on bayesian probabilistic model. In *Proceedings of ICPR*, pages 70–74, 1996.

G.J. Braun and M.D. Fairchild. Image lightness rescaling using sigmoidal contrast enhancement functions. In *IS&T/SPIE Electronic Imaging '99, Color Imaging: Device Independent Color, Color Hardcopy, and Graphic Arts IV*, 1999.

T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Telecommunications. Wiley-Interscience, New York, 1991.

John J. Craig. *Introduction to Robotics*. Addison Wesley Publishing Company, 2nd edition, 1998.

K. Daniilidis and H.-H. Nagel. Coupling of rotation and translation in motion estimation of planar surfaces. In IEEE, editor, *CVPR93*, pages 188–193, New York, USA, 1993.

P. J. Debevec and J. Malik. Recovering high dynamic radiance maps from images. In *SIGGRAPH 97*, pages 369–378, 1997.

A. P. Dempster, N.M. Laired, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Soc., Ser. B*, 39(1):1–38, 1977.

R. Deriche and G. Giraudon. A computational approach for corner and vertex detection. *The international journal of Computer Vision*, 10(2):101–124, 1993.

Jeffrey M. DiCarlo and Brian A. Wandell. Rendering high dynamic range images. In *Proc. SPIE, Image Sensors*, page 3965, 2000.

Pier-Luigi Dragotti and Martin Vetterli. Shift-invariant gibbs free denoising algorithm based on wavelet transform footprints. In *SPIE*, San Diego, CA, 2000.

Mark D. Fairchild. *Color Appearance Models*. Addison, Wesley, Reading, MA, 1998.

Olivier Faugeras. *Three-dimensional Computer Vision: a Geometric Viewpoint*. the MIT Press, 1993.

James A. Ferwerda, Sumanta N. Pattanaik, Peter Shirley, and Donald P. Greenberg. A model of visual adaptation for realistic image synthesis. In *SIGGRAPH '96*, pages 249–258, 1996.

P. Fua and C. Brechtbuehler. Imposing hard constraints on soft snakes. In *European Conference on Computer Vision*, pages 495–506, Cambridge, England, 1996.

Pascal Fua. Modeling heads from uncalibrated video sequences. *Videometrics*, IV, 1999.

Brian Funt, Florian Ciurea, and John McCann. Retinex in matlab. In *IS&T Eighth Color Imaging Conference*, pages 112–122, 2000.

D Gatica-Perez, S Ming Ting, and Gu Chuang. Semantic video object extraction based on backward tracking of multivalued watershed. In *ICIP*, Kobe, 1999.

Gene H. Golup and Charles F. van Loan. *MATRIX Computations*. John Hopkins University Press, 3rd edition, 1996.

E. M. Granger and Heurtley J.C. Visual chromaticity-modulation transfer function. *Journal of the Optical Society of America*, 63(9):1173–1174, September 1973.

C. Guestrin, F. Cozman, and E. Krotkov. Fast software image stabilization with color registration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 19–24, vol. 1, 1998.

Gregory D. Hager and Peter N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1029, 1998.

K.J. Hanna, H.S. Sawhney, R. Kumar, Y. Guo, and S. Samarasekara. Annotation of video by alignment to reference imagery. In *IEEE International Conference on Multimedia Computing and Systems*, pages 38–43, vol.1, 1999.

Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision*, volume 1. Addison-Wesley Publishing Company, 1992.

Richard I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *2nd European Conference on Computer Vision*, pages 579–587, Santa Margherita Ligure, 1992.

Richard I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

David Hasler. *Perspectives on Panoramic Photography*. PhD thesis, Swiss Federal Institute of Technology (EPFL), 2001.

David Hasler, Luciano Sbaiz, Serge Ayer, and Martin Vetterli. From local to global parameter estimation in panoramic photographic reconstruction. In *Proc. IEEE ICIP*, Kobe, Japan, 1999.

Simon Haykin. *Communication Systems*. John Wiley & Sons, New York, 3rd edition, 1994.

Berthold K.P. Horn. *Robot Vision*. McGraw-Hill, New York, 1986.

Xiaoping Hu and Narendra Ahuja. Necessary and sufficient conditions for a unique solution of plane motion and structure. *IEEE Transactions on Robotics and Automation*, 11(2): 304–308, 1995.

P.J. Huber. *Robust Statistics*. Wiley-Interscience, New York, 1981.

R.W.G. Hunt. *Measuring Colour*. Fountain Press, England, 3 edition, 1998.

ISO 14524:1999. *Electronic still picture cameras- Methods for measuring opto-electronic conversion functions (OECFs)*. 1999. 1999 Photography.

ISO16067 WD : 1999. *Electronic Scanners for Photographic Images - Spatial Resolution Measurements, Part 1: Scanners for Reflective Media*. 1999.

ITU-R Recommendation BT.709-3. *Parameter Values for the HDTV Standards for Production and International Programme Exchange.* 1998.

D.H. Jacobs. The Stiles-Crawford effect and the design of telescopes. *Journal of the Optical Society of America*, 34:694, 1944.

Allan Jepson and Michael J. Black. Mixture models for optical flow computation. In *CVPR '93*, pages 760–761, 1993.

A.E. Johnson and Sing Bing Kang. Registration and integration of textured 3-d data. In *International Conference on Recent Advances in 3-D Digital Imaging and Modeling, 1997.*, pages 234–241, 1997.

S.X. Ju, M.J. Black, and A.D. Jepson. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *CVPR*, pages 307–314, 1996.

K Kanatani. *Geometric computation for Machine Vision*. MIT Press, Cambridge, MA, USA, 1991.

Kenichi Kanatani. 3-D motion analysis of a planar surface by renormalization. *IECE trans. Inf. and Syst.*, 8(August), 1995.

D.H. Kelly. Spatiotemporal variation of chromatic and achromatic contrast thresholds. *Journal of the Optical Society of America*, 73(6):742–750, 1983.

R. Kingslake. *Optical Systems Design*. New York, 1983.

Yuishi Kobayashi and Toshikazu Kato. A high fidelity contrast improving model based on human vision mechanisms. In *IEEE Conference on Multimedia Computing and Systems*, pages 578–584, 1999.

Reinhard Koch, Marc Pollefeys, and Luc Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In *5th European Conference in Computer Vision*, volume 1, pages 55–71, Freiburg, Germany, 1998. Springer.

E. H. Land and J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61(1), January 1971.

Y. Le Grand. *Light Colour and Vision*. Chapman and Hall, London, 2nd edition, 1969.

Chia-Hoang Lee. Motion estimation of a rigid planar patch: a robust computational technique. In *the fifth annual AI systems in Government Conference*, Washington DC, USA, 1990. IEEE.

Alberto Leon-Garcia. *Probability and Random Processes for Electrical Engineering*. Addison-Wesley Publishing Company, 2nd edition, 1994.

Z. Liang, R.J. Jaszczak, and R.E. Coleman. Parameter estimation of finite mixtures using the em algorithm and information criteria with application to medical image processing. *IEEE Transactions on Nuclear Science*, 39(4):1126 –1133, 1992.

Bruce G. Lindsay. *Mixture Models: Theory, Geometry and Applications*, volume 5 of *NSF-CBMS Regional Conference Series in Probability and Statistics*. 1995.

H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

H.C. Longuet-Higgins. The reconstruction of a plane surface from two perspective projections. *Proc. R. Soc. Lond.*, B 227:399–410, 1986.

J. Lubin. *Models for Target Detection an Recognition*, chapter Vision Models for Target Detection an Recognition, pages 245–283. World Scientific Publishing, 1995.

Q.-T. Luong, R. Deriche, O. Faugeras, and D Papadopoulos. On determining the fundamental matrix: Analysis of different methods and experimental results. Technical Report RR-1894, INRIA, 1993.

A. Majumder, Zhu He, H. Towles, and G. Welch. Achieving color uniformity across multi-projector displays. In *Visualization 2000*, pages 117–124, 2000.

Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, London, 1998.

S. Mann and R. W. Picard. ‘Video Orbits’: Characterizing the coordinate transformation between two images using the projective group. In *ICCV95*, Cambridge, MA, USA, 1995.

Steve Mann. Comparametric equations with practical applications in quantigraphic image processing. *IEEE Transactions on Image Processing*, 9(8):1389–1406, 2000.

Steve Mann and Rosalind W. Piccard. Being ’undigital’ with digital cameras: Extending dynamic range by combining differently exposed pictures. In *IS&T 46th Annual Conference*, pages 422–428, May 1995.

Jarrold E. Marsden. *Elementary Classical Analysis*. W.H. Freeman and Company, New York, 1974.

Stephen Maybank. *Theory of Reconstruction from Image Motion*. Springer-Verlag, 1992.

G.J. McLachlan and K.E. Basford. *Mixter Models Inference and Application to Clustering*. Dekker, Inc., New York and Basel, 1988.

A. Michelson. *Studies in optics*. University of Chicago Press, 1927.

Tomoo Mitsunaga and Shree K. Nayar. Radiometric self calibration. *Proc. Of Computer Vision and Pattern Recognition*, 1:374–380, June 1999.

P. Moon and D.E. Spencer. On the Stiles-Crawford effect. *Journal of the Optical Society of America*, 34:319–329, 1944.

Todd K. Moon. The expectation maximisation algorithm. *IEEE Signal Processing Magazine*, pages 47–60, 1996.

Nathan Moroney. Local color correction using non-linear masking. In *8th IS&T Color Imaging Conference*, November 2000.

Kathy T. Mullen. The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings. *Journal of Physiology*, 359:381–400, 1985.

Marcus Nadenau. *Integration of Human Vision Models Into High Quality Image Compression*. PhD thesis, Swiss Federal Institute of Technology, Lausanne (EPFL), 2000.

Nathan S. Netanyahu and Isaac Weiss. Analytic outlier removal in line fitting. In *Proceedings of the 12th IAPR International. Conference on Computer Vision and Image Processing.*, volume 2B, pages 406–408, 1994.

Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, Englewood Cliffs, NJ, 1989.

Jong-Il Park, Nobuyuki Yagi, Kazumasa Enami, Kiyoharu Aizawa, and Mitsutoshi Hatori. Estimation of camera parameters from image sequence for model-based video coding. *IEEE Trans. on Circuit and Systems for Video Technology*, 4(3):288–296, 1994.

Sumanta N. Pattanaik, James A. Ferwerda, Mark D. Fairchild, and Donald P. Greenberg. A multscale model of adaptation and spatial vision for realistic image display. In *SIGGRAPH*, 1998.

E. Peli. Contrast in complex images. *Journal of the Optical Society of America*, 7(10): 2032–2040, 1990.

Allen B. Poirson and Brian A. Wandell. Appearance of color patterns: Pattern-color separability. *Optics and Image Science*, 10(12):2458–2470, 1993.

Allen B. Poirson and Brian A. Wandell. Pattern-color separable pathways predict sensitivity to simple colored patterns. *Vision-Research*, 36(4):515–526, 1996.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

S. J. Roberts, D. Husmeier, I. Rezek, and W. Penny. Bayesian approaches to gaussian mixture modelling. *IEEE Transactions on Pattern Analysis and Machine Intellingence*, 20(11): 1133–1142, 1998.

Mark A. Robertson, Sean Borman, and Robert S. Stevenson. Dynamic range improvement through multiple exposures. In *International Conference On Image Processing, ICIP*, 1999.

Peter J. Rousseeuw and Annick M. Leroy. *Robust regression and outlier detection*. Applied probability and statistics. John Wiley and Sons, Inc., New York, 1987.

H. Rushmeier and F. Bernardini. Computing consistent normals and colors from photometric data. In *Second International Conference on 3-D Digital Imaging and Modeling*, pages 99–108, 1999.

B. E. A. Saleh and M. C. Teich. *Fundamentals of Photonics*. John Wiley, 1991.

Marcel J. Sankeralli and Kathy T. Mullen. Estimation of the l-, m-, and s-cone weights of the postreceptoral detection mechanisms. *Journal of the Optical Society of America*, 13(5): 906–915, 1996.

Harpreet S. Sawhney and Serge Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):814–830, 1996.

Harpreet S. Sawhney, Steve Hsu, and R. Kumar. Robust video mosaicing through topology inference and local to global alignement. In Springer, editor, *ECCV98*, pages 103–119, Freiburg, Germany, 1998.

Philippe Schroeter, Jean-Marc Vesin, Thierry Langenberger, and Reto Meuli. Robust parameter estimation of intensity distributions for brain magnetic resonance images. *IEEE transactions on Medical Imaging*, 17(2):172–186, 1998.

G.A.F. Seber and C.J. Wild. *Nonlinear Regression*. John Wiley and Sons, New York, 1989.

H.-Y. Shum and R. Szeliski. Panoramic image mosaicing. Technical Report MSR-TR-97-23, Microsoft Research, 1997.

Greg Spencer, Peter Shirley, Kurt Zimmerman, and Donald Greenberg. Physically-based glare effects for digital images. In *SIGGRAPH 95*, pages 325–334, 1995.

Andrew Stockman, Donald I. A. MacLeod, and Nancy E. Johnson. Spectral sensitivity of the human cones. *Journal of the Optical Society of America*, 10(12):2491–2521, 1993.

Sabine Süsstrunk, Jack Holm, and Graham D. Finlayson. Chromatic adaptation performance of different RGB sensors. In *IS&T/SPIE Electronic Imaging*, San Jose, CA, 2001.

Jun-ichi Takeuchi. Characterization of the bayes estimator and the MDL estimator for exponential families. *IEEE Transactions on Information Theory*, 43(4):1165–1174, 1997.

P. S. Torr, R. Szeliski, and P. Anandan. An integrated bayesian approach to layer extraction from image sequences. In *ICCV '99*, 1999.

R Y Tsai and T S Huang. Estimating three-dimentional motion parameters of a rigid planar patch. *IEEE transactions Acoustics, Speech and Signal processing*, 29(6):1147–1152, 1981.

J Tumbling, J. Hodgkins, and B. Guenter. Display of high contrast images using models of visual adaptation. In *Visual Processing SIGGRAPH 97*, 1997.

Gerard J.C. Van der Horst and Maarten A. Bouman. Spatiotemporal chromaticity discrimination. *Journal of the Optical Society of America*, 59(11):1482–1488, November 1969.

P. Vandergheynst and J.-F. Gobbers. Directional wavelet frames : Design and algorithms. *Submitted to IEEE Trans. Imag. Process.*, 2001. http://ltswww.epfl.ch/ vandergh/publications.html.

J. Von Kries. Chromatic adaptation. In *Festschrift der Albrecht-Ludwigs-Universität*, 1902. Translation : D.L. MacAdam, "Colorimetry Fundamentals", SPIE Milestone series, Vol. MS 77, 1993.

A. Walther. *The Ray and Wave Theory of Lenses*. Cambridge Studies in Modern Optics. Cambridge University Press, Cambridge, 1995.

Brian A. Wandell. *Foundation of Vision*. Sinauer Associates inc., Sunderland, Massachusetts, 1995.

D. Wang. Unsuperwised video segmentation based on watershed and temporal tracking. *IEEE Transactions on Circuits, Systems and Video Technology, Special issue on Segmentation, Description and Retrieval of video content*, 8(5), 1998.

J.Y.A. Wang and E.H. Adelson. Layered representation for motion analysis. In *CVPR '93*, pages 361–366, 1993.

Andrew B. Watson and Joshua A. Solomon. A model of visual contrast gain control and pattern masking. *Jounal of Optical Society of America*, 14(9):2379–2391, 1997.

Y. Weiss and E.H. Adelson. A unified framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *CVPR '96*, pages 321–326, 1996.

Juyang Weng, Narendra Ahuja, and Thomas S. Huang. Motion and structure from point correspondences with error estimation: Planar surfaces. *IEEE transactions on Signal Processing*, 39(12):2691–2717, 1991.

Stefan Winkler and Pierre Vandergheynst. Local contrast in oriented pyramid decompositions. In *Proceedings 6th International Conference on Image Processing (ICIP99)*, pages 4:420–424, Kobe, Japan, 1999.

Z. Zhang. Determining the epipolar geometry and its uncertainty. Technical Report RR-2927, INRIA, 1996.

Zhengyou Zhang, Rachid Derich, Olivier Faugeras, and Quang-tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Technical Report 2273, INRIA, 1994.

THE LITTLE COUSIN SERIES IN MATHEMATICAL SIGNAL PROCESSING
Editor: Martin Vetterli

The Columbia series

1. Karlsson, Gunnar David. *Subband Coding for Packet Video.* CU/CTR/TR 137-89-16, May 1989.

2. Linzer, Elliot Neil. *Arithmetic Complexity and Numerical Properties of Algorithms involving Toeplitz Matrices.* October 1990.

3. Kovačević, Jelena. *Filter Banks and Wavelets: Extensions and Applications.* CU/CTR/TR 257-91-38, September 1991.

4. Uz, Kamil Metin. *Multiresolution Systems for Video Coding.* CU/CTR/TR 313-92-23, May 1992.

5. Radha, Hayder M. Sadik. *Efficient Image Representation using Binary Space Partitioning Trees.* CU/CTR/TR 343-93-23, December 1992.

6. Nguyen, Truong-Thao. *Deterministic Analysis of Oversampled A/D Conversion and Sigma/Delta Modulation, and Decoding Improvements using Consistent Estimates.* CU/CTR/TR 327-93-06, February 1993.

7. Herley, Cormac. *Wavelets and Filter Banks.* CU/CTR/TR 339-93-19, April 1993.

8. Garrett, Mark William. *Contributions toward Real-Time Services on Packet Switched Networks.* CU/CTR/TR 340-93-20, April 1993.

9. Ramchandran, Kannan. *Joint Optimization Techniques in Image and Video Coding with Applications to Multiresolution Digital Broadcast.* June 1993.

10. Shah, Imran Ali. *Theory, Design and Structures for Multidimensional Filter Banks and Applications in Coding of Interlaced Video.* CU/CTR/TR 367-94-14, December 1993.

11. Hong, Jonathan Jen-I. *Discrete Fourier, Hartley, and Cosine Transforms in Signal Processing.* CU/CTR/TR 366-94-13, December 1993.

12. Ortega, Antonio. *Optimization Techniques for Adaptive Quantization of Image and Video under Delay Constraints.* CU/CTR/TR 374-94-21, June 1994.

The Berkeley years

13. Park, Hyung-Ju. *A Computational Theory of Laurent Polynomial Rings and Multidimensional FIR Systems.* Coadv. with Tsit-Yuen Lam, Mathematics, U.C. Berkeley. UCB/ERL M95/39, May 1995.

14. Cvetković, Zoran. *Overcomplete Expansions for Digital Signal Processing.* UCB/ERL M95/114, December 1995.

15. McCanne, Steven Ray. *Scalable Compression and Transmission of Internet Multicast Video.* Coadv. with Van Jacobson, Lawrence Berkeley National Laboratory. UCB/CSD 96/928, December 1996.

16. Goodwin, Michael Mark. *Adaptive Signal Models: Theory, Algorithms, and Audio Applications.* Coadv. with Edward A. Lee, EECS, U.C. Berkeley. UCB/ERL M97/91, December 1997.

17. Goyal, Vivek K. *Beyond Traditional Transform Coding.* UCB/ERL M99/2, September 1998.

18. Chang, Sai-Hsueh Grace. *Image Denoising and Interpolation based on Compression and Edge Models.* Coadv. with Bin Yu, Statistics, U.C. Berkeley. UCB/ERL M99/57, Fall 1998.

The Lausanne time

19. Prandoni, Paolo. *Optimal Segmentation Techniques for Piecewise Stationary Signals.* EPFL 1993(1999), June 1999.

20. Lebrun, Jérôme. *Balancing MultiWavelets.* EPFL 2192(2000), May 2000.

21. Weidmann, Claudio. *Oligoquantization in Low-Rate Lossy Source Coding.* EPFL 2234(2000), July 2000.

22. Balmelli, Laurent. *Rate-Distortion Optimal Mesh Simplification for Communication.* EPFL 2260(2000), September 2000.

23. Marziliano, Pina. *Sampling innovation.* EPFL 2369(2001), April 2001.

24. Horbelt, Stefan. *Splines and Wavelets for Image Warping and Projection.* EPFL 2397(2001), May 2001.

25. Hasler, David. *Perspectives on Panoramic Photography.* EPFL 2419(2001), July 2001.

26. Do, Minh N. *Directional Multiresolution Image Representations.* EPFL 2500(2001), November 2001.