Universidad de la República
Facultad de Ingeniería
Instituto de Computación

Master Thesis
Director: Andrés Almansa

Dense Urban Elevation Models from Stereo Images by an Affine Region Merging Approach.

Javier Preciozzi

Montevideo, September 18, 2006

Abstract

The main subject of this thesis is the computation of Dense Disparity Maps from a pair of satelite or aerial stereo images from an urban scene, taken from two different viewpoints. Several steps are needed to obtain the final disparity map from the pair of images. We focus here on one of these steps: how to match the points in one image with the points in the other one. This matching process is closely related to the computation of the altitudes of the objects present in the scene. Indeed, the precision we can obtain in these altitude values is directly proportional to the precision in the matching process. This precision in the altitude is also inversely proportional to the distance between both viewpoints where the images are taken(*baseline*).

The matching process is a widely studied field in the Computer Vision Community and several methods and algorithms have been developed so far ([31, 27, 49]). Most of them consider a big baseline configuration, which increases the performance in the altitude and also simplifies the matching process. However, this assumption presents a major drawback with objects that are occluded in one image but appear in the other one. The bigger the baseline is, the more objects are occluded in one image and are not occluded in the other one.

Recently, a different approach in which the images are taken with a very small baseline started to be analyzed ([19, 20]). This approach has the advantage of eliminating most of the ambiguities presented when one object occluded in one image is not occluded in the other one. Indeed, if we consider that we have a very small baseline, the occlusions presented in both images are almost the same. Now, this configuration obviously decreases the precision in the final altitude. In order to continue obtaining highly accurate altitude values, the precision in the matching process must be improved. The methods developed so far which consider the small baseline approach, compute altitude values with a high precision at some points, but leave the rest of them with no altitude values at all, generating a non-dense disparity map. Based on the fact that piecewise-affine models are reasonable for the elevation in urban areas, we propose a new method to interpolate and denoise those non-dense disparity maps.

Under lambertian illumination hypothesis [1], it is reasonable to assume that homogeneous regions in the graylevel image, correspond to the same affine elevation model. In other words, the borders between the piecewise affine elevation model are included to a large extent within contrasted graylevel borders. Hence, it is reasonable to look for an piecewise affine fit to the elevation model where the borders between regions are taken from a graylevel segmenation of the image

We present a region-merging algorithm that starts with an over-segmentation of the gray-level image. The disparity values at each region are approximated by an affine model, and a *meaningfulness* measure of the fit is assigned to each of them. Using this meaningfulness as a merging order, the method iterates until no new merge is possible, according to a merging criterion which is also based on the meaningfulness of each pair of neighboring regions. In the last step, the algorithm performs a validation of the final regions using again the meaningfulness of the fit. The regions validated in this last step are those for which the affine model is a good approximation.

The region-merging algorithm presented in this work can be seen as an attempt to incorporate a semantical meaning to real scenes: we have developed a validation method to determine whether the data within a region is well approximated by an affine model or not. Hence, we could analyze more complex models, defining a suitable validation criterion for each of them. In this way, we can search for the model that best explains a given data set in terms of its meaningfulness.

[1] the surface luminance is the same regardless of the angle of view

Contents

Chapter 1

Introduction

1.1 Motivation

The subject of this work is how we can recover the information of a real scene from a set of images taken from different viewpoints. In particular, we focus our research on how to obtain a Digital Elevation Model from Urban scenes, from a pair of satellite or aerial images. This information is required on many applications, for instance, on the visualizarion of a real 3D scene, as we can see in Figure 1.1. In the case of urban scenes, this information is generally obtained from manual measures taken directly on the field, which leads to an extremely slow and error prone process.
Clearly, the missing information when we have an image of the real scene is the distance of the different objects present in the scene to the optical center. The Stereoscopy Principle states that in order to obtain this information, we need at least two images, and in general, two images are enough to recover most of the distances, which in turn enables to recover the whole scene.

(a) A view of Manhattan Island

(b) The same view with 3D reconstruction

(c) A view of Touluse

(d) A view of Touluse from a different angle. Note that in this case is no information on the altitude.

Figure 1.1: Different views with and without 3D information, extracted from Google Earth.

The process to obtain the distance of the objects in the scene to the optical center involves several steps, but the most difficult one is to make the points in one image correspond with the points in the other one. This process is known as *matching* and once we have set this matching, we can obtain the distance of the points by simple geometrical results. We can express this relation with the equation

$$\epsilon = \frac{B}{H}h \tag{1.1}$$

where ϵ is the disparity value of the point, B is the distance between both optical center (the *baseline*) and H the altitude of the cameras. The disparity value of a real point in the 3D scene is defined as

the difference between the position of the projection of the point in one image and the position of the projection of the same point in the other one. Figure 1.2 shows an sketch of these relations.

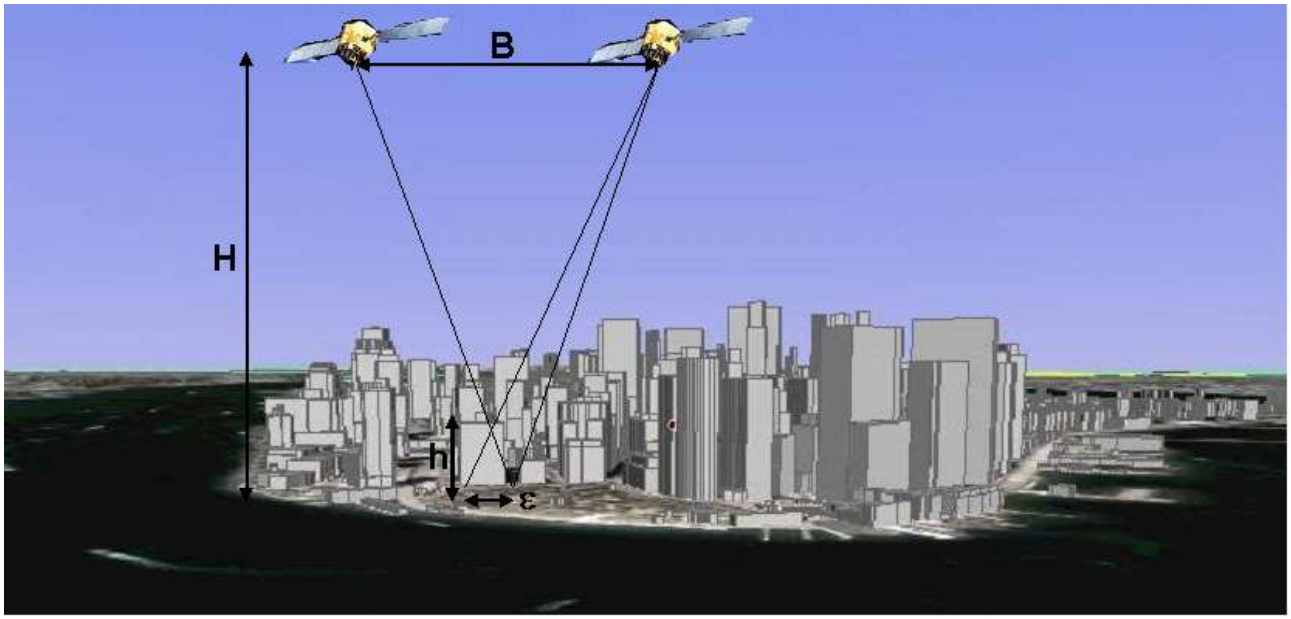


Figure 1.2: Sketch of the capturing process

It is clear from this equation that the relation B/H has a great influence on the precision we can obtain in the altitude. This is why, until now, images have been taken with a big baseline, and most of the methods developed so far use this configuration as their working hypothesis.
In the case of satellite images, as the altitude of the satellite cannot be adjusted to improve this relation, the value of B is the one which is increased. In practice, this means that both images are taken from two viewpoints far enough. Although this improves the precision obtained in h, it introduces two major drawbacks. First, in order to take both images from viewpoints far enough, the satellite must fly several minutes. This introduces problems with moving objects (such as cars) and with shadows because they change the position between one image and the other. The other problem is that if we take the image far from the other one, the occlusions presented in one image are disoccluded in the other one and vice versa.

In this project we consider a different approach, firstly addressed by Rougé et. al.[19]: to consider that both images are taken with a very small baseline. Now, in order to achieve the same precision that we obtain in the case of a big baseline, we must improve the precision in the matching process, which in turns leads to the development of techniques to compute the disparity map with subpixel accuracy. With this approach we cope with the two major drawbacks of the classic approaches: the displacement of the objects in the scene (cars, shadows, etc.) and the occlusion/disocclusion of regions. The differences between occlusions in one image and in the other one are also eliminated.

Most of the classical " big baseline" algorithms produce a non-dense disparity map: there are some points from which the algorithm cannot determine the disparity value. However, in many applications it is necessary to have a complete dense disparity map. Think for instance at any 3D reconstruction of a real scene; using a non-dense disparity map leads to "holes" in the 3D real scene. In those cases, we must interpolate the valid values. We analyze here a region merging approach that receives as input the non-dense disparity map and gets as output a dense one. To do so, an affine hypothesis is assumed: the regions present in the images can be modeled by affine transformations. This is a well suitable model, because in this project we focus on urban scenes. The novel approach here is the criterion defined to merge two regions which are based on the definition of an *a contrario* model [25].

1.2 Context

The present work has been developed as part of a cooperation project between Uruguay, France and Spain which main subject is to validate the small baseline approach in order to incorporate it in the design of future satellites. This project involves several institutes and universities:

- Instituto de Computación (InCo) from Facultad de Ingeniería of the Universidad de la República (UdeLaR) - Uruguay
- Centre de mathématiques et de leurs applications (CMLA) from Ecole Normale Superieure de Cachan - France
- Centre National d'Etudes spatiales (CNES) - France
- Universitat Pompeu Fabra - España.

Most of the work related with this thesis has been developed as part of the PDT project: "Small baseline stereo for Urban Digital Elevation Models using variational and region-merging techniques" [4] led by Andrés Almansa.

1.3 Organization of the document

The document is organized as follows. In Chapter 2 we highlight the most relevant geometrical aspects of the image formation process that in turns enables us to recover a real scene from a pair of images taken from different viewpoints. We also show the relation between the precision obtained in the disparity map and the precision of the altitude. We finally focus our attention on the case of urban scenes and how we can model them with piecewise affine transformations.
Chapter 3 is a review of matching algorithms. In Section 3.1 and 3.2 we review the matching algorithms based on the classical big baseline approach, and in Section 3.3 we present the new sub-pixel methods developed to manipulate pairs of images with small baseline configuration.
In Chapter 4 we introduce a Region-Merging approach, which is the main contribution of this work to the problem of obtaining a sub-pixel dense disparity map from a non-dense one. We start with a review of the *a contrario* models, and the definition to the case of affine transformation fitting. We explain the region-merging algorithm in detail, notably the merging criterion and the validation process, both based on the *a contrario* model.
Chapter 5 presents an alternative way to compute the number of false alarms of a transformation, where instead of considering a discrete model, a continuous approach is used, leading to a more accurate estimation.
In Chapter 6 we study the performance of the region-merging algorithm when we consider different initial segmentation, which is the most important input parameter of the method. We also propose a segmentation technique based on the detection of polygons, more suitable to urban scenes.
In Chapter 7 we present the experiments we have done to analyze the performance of the region-merging algorithm. We compare it with other existing techniques.
Finally, Chapter 8 presents the conclusions we have obtained from this work and we also note some of the directions this work leaves open for future researches.

Chapter 2

Stereo Vision Overview

In this chapter we review the model behind stereo vision and how the information contained in a pair of images of the same scene can be used to reconstruct the real scene. The content of this chapter is a summary of the most relevant concepts of stereo vision, following the books of Hartley and Zisserman [31], Faugeras, Luong and Papadopoulo [27] and the tutorial of Pollefeys [49].

In Section 2.1, a brief introduction on how images are formed is presented. Section 2.2 introduces the epipolar geometry that models the relations between images of the same scene and we also single out some well known results, notably the epipolar constraint that simplifies enormously the computation of the disparity map. In Section 2.3 a brief description of the rectification process (the process to align epipolar lines horizontally) is presented. Section 2.4 presents the stereoscopy principle that enables us to compute the depth of the objects in the scene by the disparity map and the relations between the cameras. Section 2.5 explains how we can obtain digital elevation models from satellite or aerial images. We also analyze urban models which have some specific characteristics that we can use to obtain better results in the computation of the disparity map.

2.1 Image Formation

The problem of reconstructing the information of a real scene from a set of one or more images, is one of the most important fields of image processing and computer vision. In order to study how we can obtain the real scene from a set of images, it is important to understand first how images are formed. The process of image formation performed by any camera can be modeled by two kinds of projections: *perspective or central projection* and *orthographic or parallel projection*. Both projections can be obtained by the same camera model: the *pinhole camera*.

2.1.1 Pinhole camera

The pinhole camera model was developed in the Renaissance as a way to obtain the real image of a scene. Figure 2.1 shows a sketch of this model, that can be described as follows:
Given a plane $\mathcal{I}$ called the *image* or *retinal plane* and a point $\mathbf{C}$ which does not belong to $\mathcal{I}$ that we name *optical center*, the projection $\mathbf{m}$ of a point of the space $\mathbf{M}$ is the intersection of the *optical ray* $(\mathbf{C}, \mathbf{M})$ with the image plane. We define the projection of the optical center into the image plane as the *principal point* $\mathbf{c}$. With all these elements, we can define an orthonormal system of coordinates in the image plane centered at $\mathbf{c}$, and we can define a three dimensional system of coordinates centered at the optical center $\mathbf{C}$ called the *camera coordinate system*. This camera coordinate system has two axes parallel to the image plane, and the other one parallel to the optical axis.

The other important measure is the *focal length*, which is the distance between the point $\mathbf{C}$ and the plane $\mathcal{I}$. The relations between the coordinates of the real point $\mathbf{M}$ $[X, Y, Z]$ with its projection in the image plane $\mathbf{m}$, $[x, y]$ are:

$$x = f\frac{X}{Z}$$

$$y = f\frac{Y}{Z}$$

These relations are obtained as a consequence of Thales' theorem and the properties of similar triangles. Using homogeneous coordinates for $\mathbf{m}$ and $\mathbf{M}$ we obtain a projection equation:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathcal{P}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.1}$$

Matrix $\mathcal{P}$ is known as *perspective projection matrix* and models the *perspective projection*. The form of matrix $\mathcal{P}$ in Equation (2.1) is simplified since we have defined both coordinate systems as orthonormal, with two axes in the real scene parallel to the ones in the image, and the other one parallel to the

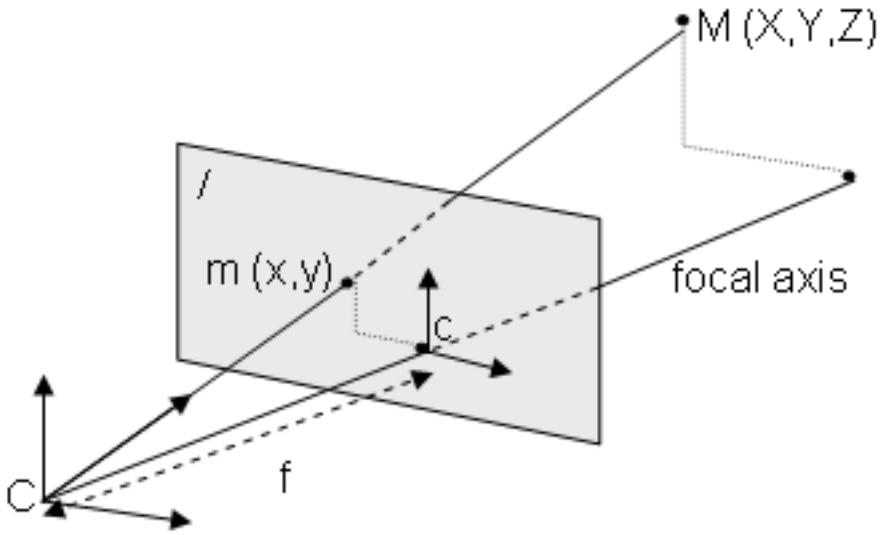


Figure 2.1: The Pinhole camera model for central or perspective projection

optical axis. We also define the same scale for all dimensions in both coordinates systems. In its more general form, the *perspective projection matrix* $\mathcal{P}$ is defined as a 3×4 matrix of rank 3, and can be decomposed as

$$\mathcal{P} = \mathbf{ARt} \tag{2.2}$$

where $\mathbf{A}$ is a 3×3 matrix that maps the normalized image coordinates to the retinal image coordinates, and $[\mathbf{Rt}]$ is the rotation and translation from the world's coordinate system to the camera's coordinate system. If we refer the camera coordinate system to the image plane, which is the same as moving the camera coordinate system along the Z-axis an amount of f, the perspective projection matrix can be written as

$$\mathcal{P} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & f \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix} \tag{2.3}$$

If we let f go to infinity, we obtain the *orthographic projection*; A sketch of this projection is shown in Figure 2.2.

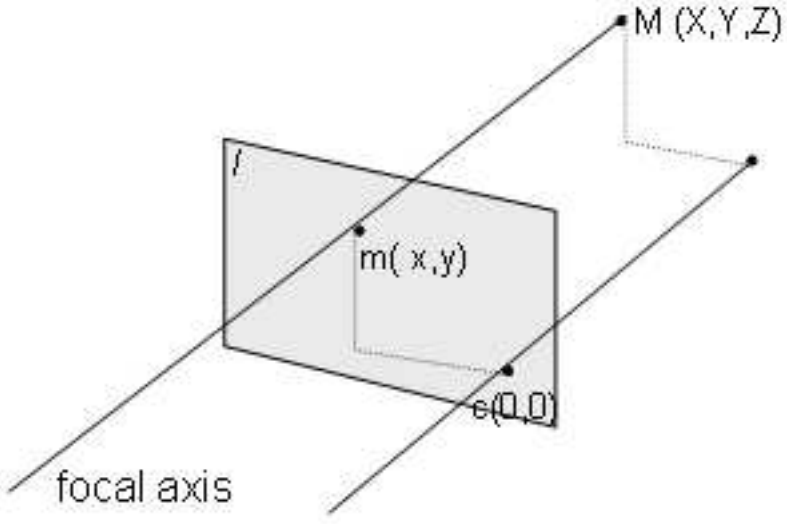


Figure 2.2: The parallel or orthographic projection

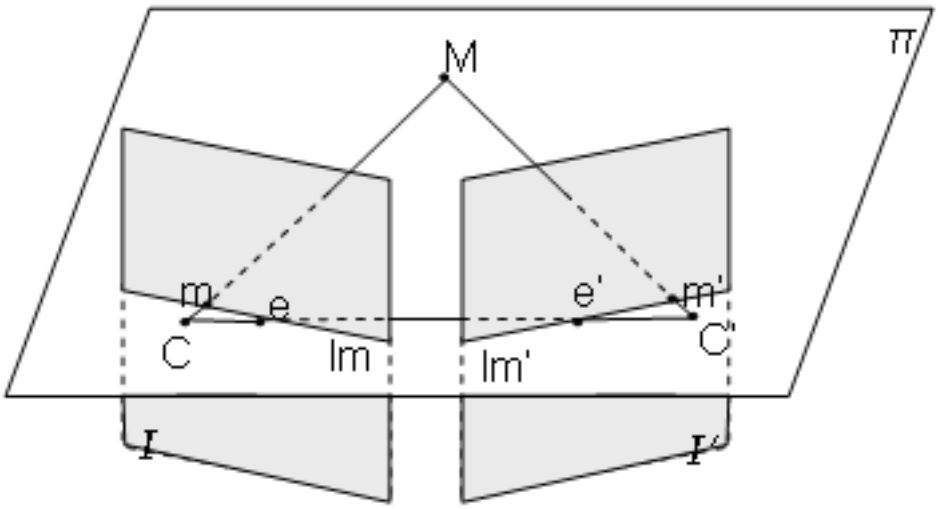


Figure 2.3: The epipolar geometry

Although focal length is always a finite value because the focal center is obviously not at infinity, the parallel projection is a suitable model for images of scenes where the distance from the image plane to the real scene is much greater than the one between the optical center and the image plane.

One obvious observation is that in all the projection processes, one of the dimensions is lost: we pass from a three dimension space (the real scene) to a two dimension space (the images). It is clear that we cannot reconstruct the original scene from only one image. This is because all the points lying in the same projection ray would have the same corresponding point. In order to obtain the real scene, we need at least two images taken from different viewpoints. The different images are related by the *epipolar geometry* which models the relation between images taken from two different viewpoints.

2.2 Epipolar Geometry

Epipolar geometry explains the relations between two or more camera systems. Consider the case of two images shown in Figure 2.3. Let $\mathbf{C}$ and $\mathbf{C}'$ be the optical centers and $\mathcal{I}$, $\mathcal{I}'$ the image planes for each of the camera systems. Given a point $\mathbf{M}$ in the space, the points $\mathbf{C}$, $\mathbf{C}'$ and $\mathbf{M}$ define a plane Π known as *epipolar plane*. This epipolar plane intersects image planes $\mathcal{I}$ and $\mathcal{I}'$ in lines l_m and $l_{m'}$ respectively. Let $\mathbf{m}$ be the projection of point $\mathbf{M}$ in $\mathcal{I}$. The corresponding projection $\mathbf{m}'$ of the point $\mathbf{M}$ in $\mathcal{I}'$ plane must lie in the line $l_{m'}$, the *epipolar line* of $\mathbf{m}$. Varying the position of point $\mathbf{M}$ we can observe that all epipolar lines at image $\mathcal{I}'$ (conversely $\mathcal{I}$) have a point $\mathbf{e}'$ ($\mathbf{e}$) in common. These points are the intersection of the line $\mathbf{C}$,$\mathbf{C}'$ with each image plane. We have the following proposition, known as *co-planarity constraint*[27]:

Proposition 1 (co-planarity constraint) *If a point* $\mathbf{m}$ *of image* $\mathcal{I}$ *and a point* $\mathbf{m}'$ *of image* $\mathcal{I}'$ *correspond to a single real point* $\mathbf{M}$*, then* $\mathbf{m}$,$\mathbf{m}'$,$\mathbf{C}$ *and* $\mathbf{C}'$ *must lie in a single plane.*

This constraint has a strong impact on the matching process, since the correspondence of a point in the first image, must lie in the epipolar line in the second one. Thus, the search for a correspondence is reduced to a one dimension search. This is known as *epipolar constraint.*

Let's analyze which is the relation between the corresponding points $\mathbf{m}$ and $\mathbf{m}'$ of a point $\mathbf{M}$. We have

$$\mathbf{m} = \mathcal{P}\mathbf{M}$$

and

$$\mathbf{m}' = \mathcal{P}'\mathbf{M}$$

where $\mathcal{P}$ and $\mathcal{P}'$ are the corresponding projection matrices of each one of the images $\mathcal{I}$ and $\mathcal{I}'$. Using the decomposition of Eq. (2.2), and assuming, without losing generality, that the world coordinate system is the same one for both images (for instance the coordinates system of the second one), we have

$$\mathbf{m} = \mathbf{A}[\mathbf{Rt}]\mathbf{M}$$

and

$$\mathbf{m}' = \mathbf{A}'[\mathbf{I0}]\mathbf{M}$$

In [27] it is shown that eliminating $\mathbf{M}$ from both equations leads to:

$$\mathbf{mFm}' = 0 \tag{2.4}$$

with

$$\mathbf{F} = \mathbf{A}^{-T}[\mathbf{t}]_{\times}\mathbf{R}\mathbf{A}'^{-1} \tag{2.5}$$

This 3×3 matrix is called the *fundamental matrix*. This matrix has some properties: it is of rank 2 and it is defined up to a scalar factor. Then, there are seven independent parameters. See [27] for a complete demonstration of this derivation and other related results.

The process of estimating the fundamental matrix between two images is known as *calibration*. Several methods exist to carry out this calibration process. In general, all methods try to build manually a set of correspondences between the relevant points of the images, and find the parameters of the matrix that make these correspondences possible. The methods differ mainly in how the system of equations between corresponding points is solved.
Once we have obtained the fundamental matrix, it can be used to rectify images, thus simplifying a bit more the matching process.

2.3 Rectification

Rectification is a transformation performed over the images to align epipolar lines horizontally. This process simplifies the stereo matching problem: due to the epipolar constraint, the search for a matching point is reduced to one dimension, and after rectification, this dimension is the horizontal axis.
There are several rectification algorithms (see [49] for a review). As an example, we show in Figure 2.4 a rectification approach known as *planar rectification*. It consists of projecting both images into a plane parallel to the baseline.

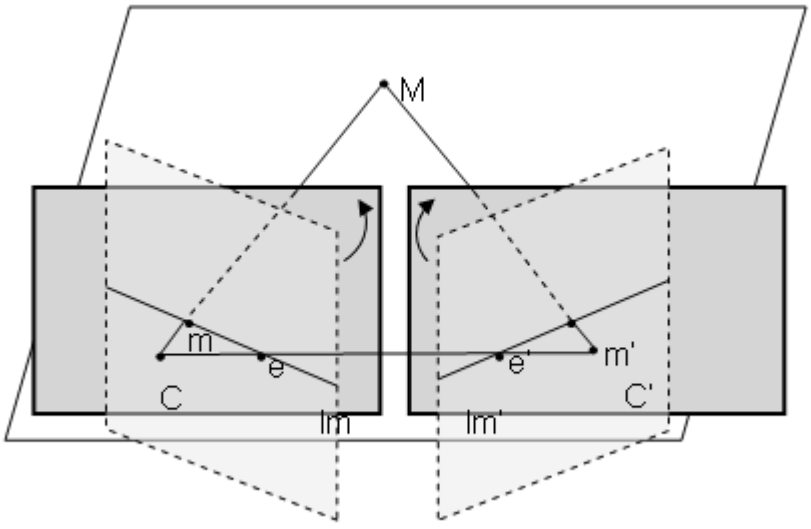


Figure 2.4: Planar rectification: After rectification, images become parallel to the baseline and the corresponding points lie in the same row

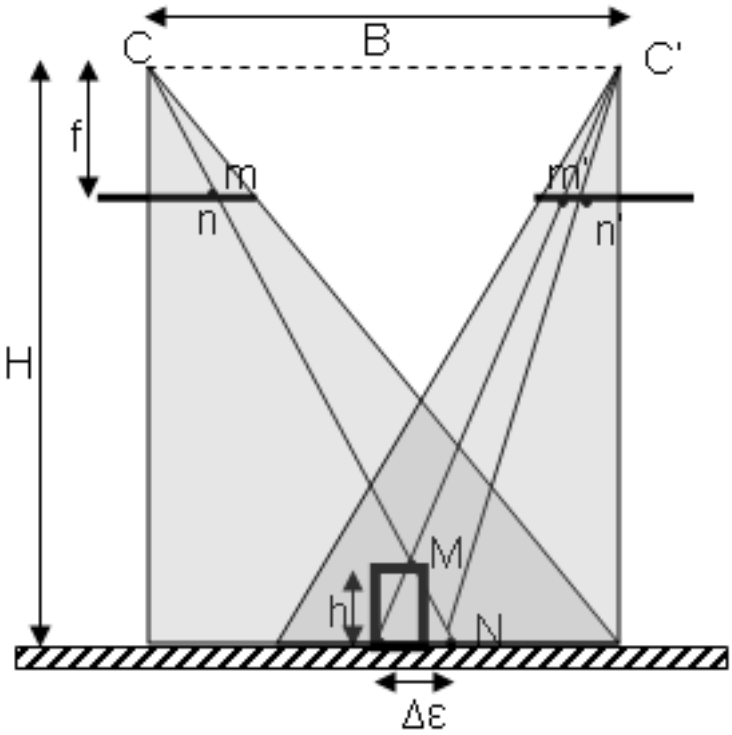


Figure 2.5: Nonverged model

2.4 Stereoscopy principle

Once we have a model for each camera and the relation between them modeled by the fundamental matrix, we can obtain the depth or altitude of the observed objects. The relations between disparity and depth is known as the *Stereoscopy Principle*.

Figure 2.5 shows a sketch of the capturing process performed by two cameras. In that diagram we assume that both image planes are the same plane, and that it is parallel to the scene. If this is not the case, we have already shown how we can obtain this configuration by calibration and rectification for any pair of images. This configuration leads to a region of the scene that was captured in both images (the dark one on the ground). For most of the points in this region, we can obtain the altitude by similar triangles:

$$\frac{B}{Z} = \frac{\epsilon}{h} \tag{2.6}$$

$$\Rightarrow h = \frac{Z}{B}\epsilon = \frac{H-h}{B}\epsilon \tag{2.7}$$

where $Z = H - h$ is the distance from the baseline to the point M, and ϵ is the disparity computed on the ground. If we call d the disparity computed in the image: $d = x_{m'} - x_{n'}$, this disparity in the image is related to the disparity on the ground by:

$$\frac{f}{d} = \frac{H}{\epsilon} \Rightarrow \epsilon = \frac{H}{f}d$$

where H/f models the relation between the pixel and the ground. If we note this relation as r_0, we finally obtain an expression that relates the disparity in the image with the altitude by:

$$d = \frac{h}{H-h}Br_0 \tag{2.8}$$

2.5 Digital Elevation Models

The model described before can be applied directly to satellite or aerial images in order to obtain a digital elevation model of the scene (DEM).
We return to Figure 2.5. If the altitude of the cameras H is much bigger than the altitude of the building h, we can approximate Equation 2.8 by its first order Taylor expansion:

$$f(x) = f(0) + f'(0)x + \mathcal{O}(|x|^2)$$

If $f(h) = \frac{h}{H-h}$, we obtain:

$$f(h) = 0 + \left(\frac{H}{(H-h)^2}|_{h=0} \right) h + \mathcal{O}(|h|^2)$$

$$\Rightarrow \frac{h}{H-h} = \frac{h}{H} + \mathcal{O}(|h|^2)$$

Finally, we can write equation (2.8) as:

$$d(h) = \frac{B}{H} h r_0 \tag{2.9}$$

This expansion is a good approximation in the case of aerial or satellite images because of the relation between H and h (H is much bigger than h).

2.5.1 Urban Elevation Models

Urban images are usually composed by roofs of buildings. Because of the epipolar geometry and parallel projection (assumed in the case of satellite images), the disparity of these regions can be modeled by a subclass of the affine transformation space. Affine transformations are of the form:

$$T(x,y) = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} T_1(x,y) \\ T_2(x,y) \end{pmatrix} = \begin{pmatrix} ax + by + e \\ cx + dy + f \end{pmatrix}$$

where e, f are the translation parameters and a, b, c, d are the parameters of a linear transformation (including scaling in both directions, rotation and shearing).
Due to the epipolar constraint and the rectification process, T_2 is the identity transformation, leading to the following model region:

$$T(x,y) = \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ 0 \end{pmatrix} = \begin{pmatrix} ax + by + e \\ y \end{pmatrix} \tag{2.10}$$

That means that our subclass of affine transformations has three parameters: the translation e, the scale factor a and the shear modeled by the b/a factor in y: $T_1(x,y) = a(x + \frac{b}{a}y) + e$

2.6 Relation between disparity and altitude precision

An error in the matching process leads to an altimetric error, which is the error in the altitude of the scene. Figure 2.6 shows a sketch of this relation.
If we note e_{match} the error performed in the matching process, the altimetric error is:

$$e_{alt} = \frac{H}{B} r_0 e_{match} \tag{2.11}$$

This equation shows that having a big value for B/H reduces the altitude error. Nevertheless, this configuration also introduces more occlusions to deal with, as we can see in Figure 2.7. On the other hand, if we consider the approach of a small B/H, which reduces the problems related to occlusions, we must improve the precision in the matching process in order to obtain accurate altitude values.

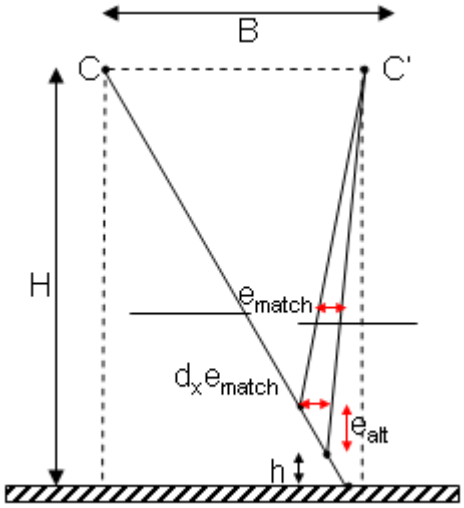


Figure 2.6: Relation between the matching error and the scene error.

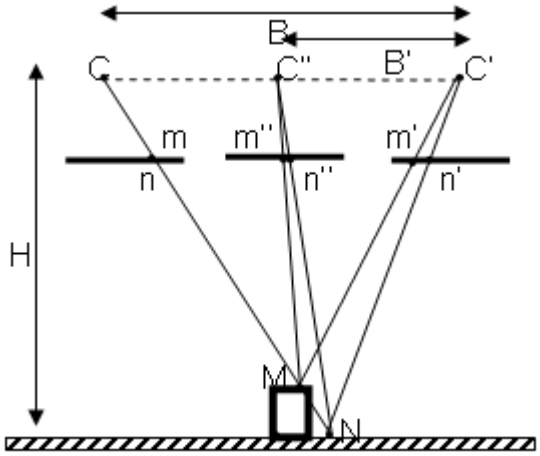


Figure 2.7: Relation between the baseline and occlusions.

This was partially addressed in [19, 20, 26].

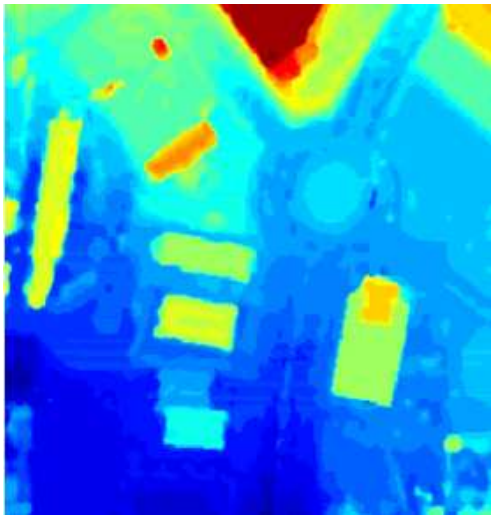
In the next chapter we present a review of the existing matching methods, with a detailed explanation of the approaches that consider a small baseline configuration (Section 3.3). The main subject of our work is how to improve the results obtained by these methods.

Chapter 3

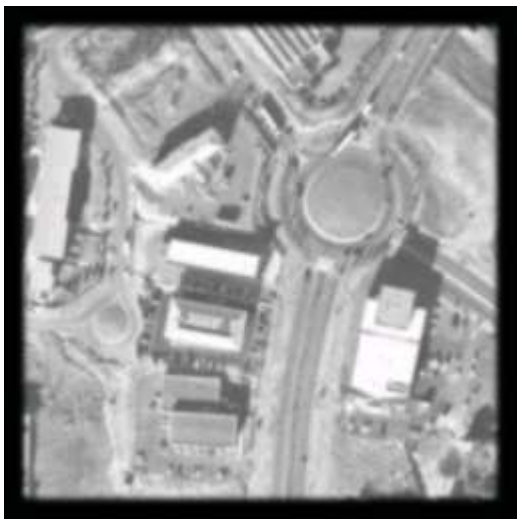
A review on Computational Stereo

In this chapter we review the most well known matching methods. To illustrate most of their characteristics, we use the pair of images shown at Figure 3.1. Most of the implementations of the methods presented here does not work with images with sub-pixel differences. Thus, we have done a simulation of the secondary image from the first one from the original sub-pixel disparity map. As the range of the original ground truth is $[-0.2, 0.5]$ we have generated the secondary image from this ground truth applying a factor of 10 in order to obtain disparities bigger than a pixel. This simulation process is explained in detail in Appendix B. In the cases where the method performs sub-pixel computation, we use the original set of images. Note that the experiments performed in this chapter are only to illustrate the different methods since no valid comparison can be done between the results obtained by the sub-pixel accuracy methods and the other ones.

(a) Original Image



(b) Ground Truth



(c) Reference Image

(d) Simulated Secondary Image

Figure 3.1: Set of images used in the illustration of the different matching algorithms. At the top on the left we have the original image and on the right the ground truth. The bottom row shows the pair of simulated images. The Reference image is the same as the original one with gaussian noise and the secondary image was simulated from the original one and the ground truth, with a scale factor of 10.

Due to the fact that there is no standard classification to matching algorithms, we follow in this work the one presented in [9]. There, the matching methods are classified in Local and Global methods, depending on whether they work on constraints over a small number of pixels or they work in the hole image. All methods presented in this section assume that the pair of images were previously calibrated and rectified, making extensive use of the epipolar constraint. See [9, 53] for a review and

performance analysis of most of the matching methods presented in the literature.

3.1 Local Methods

Local methods are the most commonly used ones for the computation of DEM. In [9] all local methods are classified in three groups: Block Matching, Gradient-Based Optimization or Optical Flow and Feature Matching.

3.1.1 Block Matching

Methods based on a block matching approach try to estimate the disparity at a given point in one image comparing a small region about that point with a series of small regions from the other image. The most classical ones are based on L^2-norm restricted to a window function φ centered at the point we want to obtain the disparity value:

$$\| u \|_{2_{\varphi_{x_0}}} = \left(\int_{\varphi_{x_0}} u(x)^2 dx \right)^{\frac{1}{2}} = \left(\int u(x)^2 \varphi(x_0 - x) dx \right)^{\frac{1}{2}} \tag{3.1}$$

With this norm, we can obtain the disparity value at a point x_o by a minimization of the sum of square differences (SSD):

$$d(x_0) = \min_m \left(\int \varphi_{x_0}(x) \left(u(x+m) - \tilde{u}(x) \right)^2 dx \right)^{\frac{1}{2}} \tag{3.2}$$

and with the corresponding L^1-norm:

$$\| u \|_{1_{\varphi_{x_0}}} = \int_{\varphi_{x_0}} |u(x)| dx = \int |u(x)| \varphi(x_0 - x) dx \tag{3.3}$$

we obtain the disparity value minimizing the sum of absolute differences (SAD):

$$d(x_0) = \int \varphi_{x_0}(x) \left| u(x+m) - \tilde{u}(x) \right| dx \tag{3.4}$$

In both equations, $u(x)$ and $\tilde{u}(x)$ represent the gray level of image u and $\tilde{u}$ respectively. These norms are sensitive to the radiometric differences between both images (the performance of the measure is affected by contrast changes). To avoid this problem, the Normalized Cross Correlation (NCC) is used instead of the original SSD or SAD. NCC is based on a weighted L^2-norm.

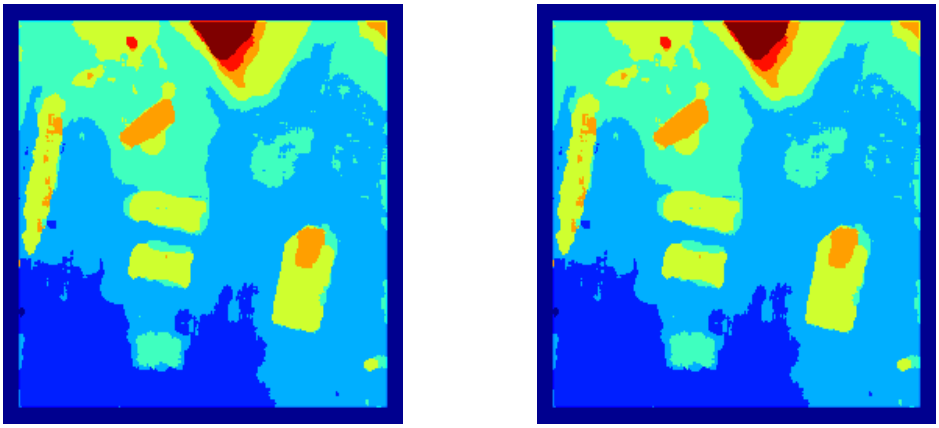
$$\left(\int \varphi_{x_0}(x) \left(\frac{u(x+m)}{\| u(x+m) \|_{2,\varphi_{x_0}}} - \frac{\tilde{u}(x)}{\| \tilde{u} \|_{2,\varphi_{x_0}}} \right)^2 dx \right)^{\frac{1}{2}} \tag{3.5}$$

which is equivalent to maximize:

$$\rho_{x_0}(m) = \frac{\int_{\varphi_{x_0}} u(x+m) \tilde{u}(x) dx}{\| u(x+m) \|_{2,\varphi_{x_0}} \| \tilde{u} \|_{2,\varphi_{x_0}}} \tag{3.6}$$

Figure 3.2 presents the disparity map obtained from SAD (3.2(a)) and SSD (3.2(b)), obtained using an implementation from Middlebury College [2, 53]. Figure 3.3 shows the relation between the window size and the final results. Note that small window sizes lead to a noisy disparity map (Figure 3.3(a)), whereas with a bigger window size, the borders of the objects are not well located and the small ones are lost (Figure 3.3(d)). Depending on the data, reasonable results can be obtained with a window size between 7 or 9 pixels size.

In Section 3.3, the correlation is used in a more sophisticated way to obtain sub-pixel disparity values. One of the major drawbacks of correlation methods is also presented, known as *adhesion phenomenon*. [19].



(a) Disparity map obtained with SAD (b) Disparity map obtained with SSD

Figure 3.2: Different disparity maps obtained with block matching: no relevant differences can be noticed at this precision level.

3.1.2 Optical Flow

Optical Flow methods are based on the *optical flow constraint* which states that the image intensity remains unchanged from two consecutive frames along the true motion path. The *optical flow equation* is derived from this constraint:

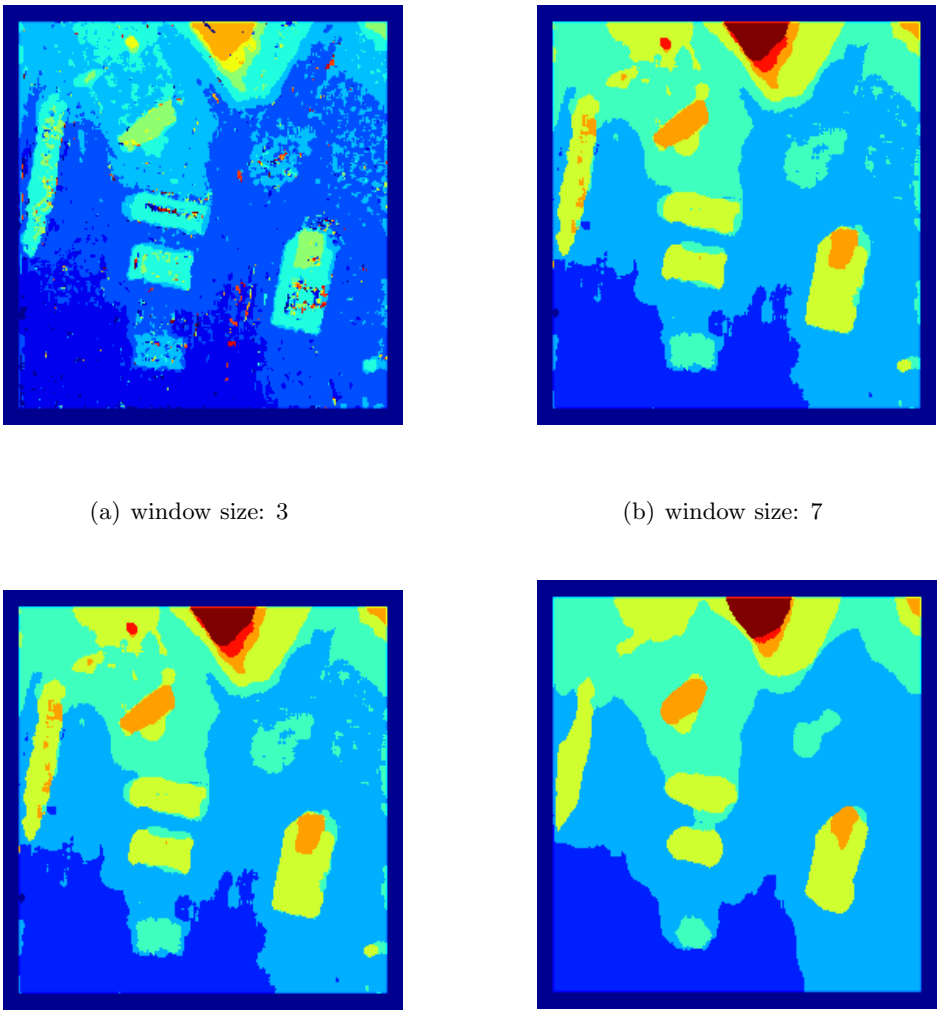
$$\partial_x I u + \partial_y I v + \partial_t I = 0$$

where $I(x, y, t)$ is the image sequence and (u, v) the vector field. The movements of the objects may be recovered by minimizing this equation. This is an "ill-posed" problem: the solution may not be unique in zones where the gradient is very small. To avoid this problem, additional assumptions on the structure of the motion field are imposed on the model. The most well known approaches are the Horn-Shunck [34] and the Lucas-Kanade [41] methods.
In the Horn-Shrunk method we look for a motion field that satisfies the optical flow equation with a regularization term on the motion field. The energy to be minimized is:

$$E_{hs}(w) = \int_\Omega (\partial_x I u + \partial_y I v + \partial_t I)^2 + \alpha^2((\partial_y u)^2 + (\partial_x v)^2 + (\partial_y v)^2) \tag{3.7}$$

Another approach is the Lucas-Kanade method, where the motion field is estimated by imposing that it must be locally regular in some known zones. This method gives one translational motion vector for each of the zones. That means that the motion field obtained with this method is non-dense. These two methods are combined in [10] to obtain a more robust method to the noise that also builds a dense motion field.
Methods based on the Optical flow constraint assume that the frames are taken almost simultaneously, something similar to the hypothesis of having a small baseline. So, these methods can be used directly with a set of images taken with a very small baseline. Figure 3.4 show this set of images and the results obtained using the Horn-Shunck approach. These examples were obtained using an implementation of the method in Megawave Package [1] (hs_flow module).

The optical flow constraint is generally violated in real images. Small global or local changes in illumination can lead to miss-computation of the real motion field. New methods based on features more robust to contrast changes or illumination are proposed. In [16] is shown that the direction of the intensity gradient is invariant to global illumination changes. Recently, an approach exploiting this property has been presented [14]. The method finds the shapes presented on the sequence and estimates the motion field under the assumption that each shape moves along the image sequence with



(a) window size: 3

(b) window size: 7

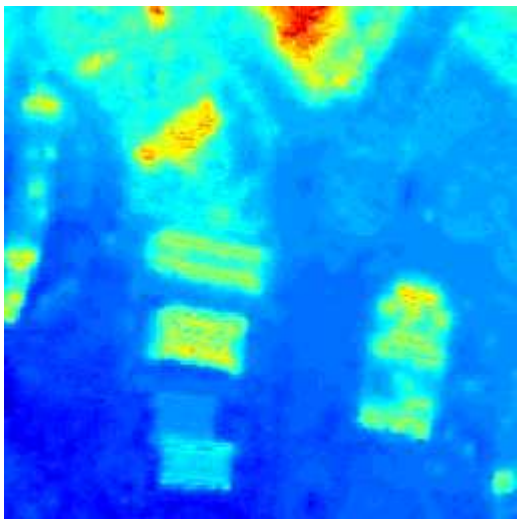
(c) window size: 9

(d) window size: 19

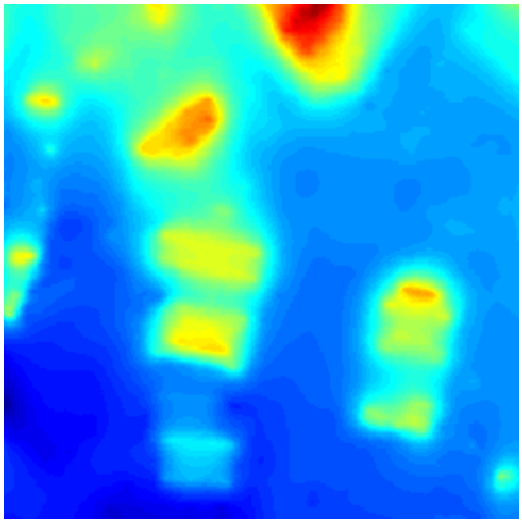
Figure 3.3: Relation of the window size in the computed disparity map. Small window size produces highly noisy maps, whereas a bigger window size leads to the removal of small objects and the lost of accuracy in the borders of the remaining objects.

(a) Reference Image

(b) Secondary Image



(c) Horn-Shunck with a lower regularization term



(d) Horn-Shunck with a more suitable regularization term

Figure 3.4: Results of the disparity map obtained using the Horn-Shunck method. By adjusting the regularization term, the result can be improved to obtain a more suitable result.

some deformation. This will be explained in detail in Section 3.3.2.

3.1.3 Feature Matching

Both block matching and optical flow methods are very sensitive near discontinuities, mainly for the convolution with the local window. Feature matching techniques try to avoid this problem by considering only reliable features to compute the disparity map. Such features are in general edges, curves, corners, etc. or any combination of them. The main drawback of these methods is that the disparity map obtained is not a dense map, and some interpolation must be performed to obtain a dense one. One technique is to first segment the image and then match the segmented regions [7, 14]. This technique is very sensitive to the original segmentation. In chapter 4 we analyze an approach based on previous segmentations in which we try to minimize the impact of this initial segmentation with a region merging technique.

In [50] we explore the applicability of a shape-elements matching algorithm [12, 45] to the low baseline stereo pairs. This approach was later abandoned because it did not provide accurate enought disparities. However, the problems encountered were more related to the implementation of the method rather than with the theory behind it. Improvements on the implementation on the method, and adaptations to the case of stereo matching are under study in [51].

3.2 Global Methods

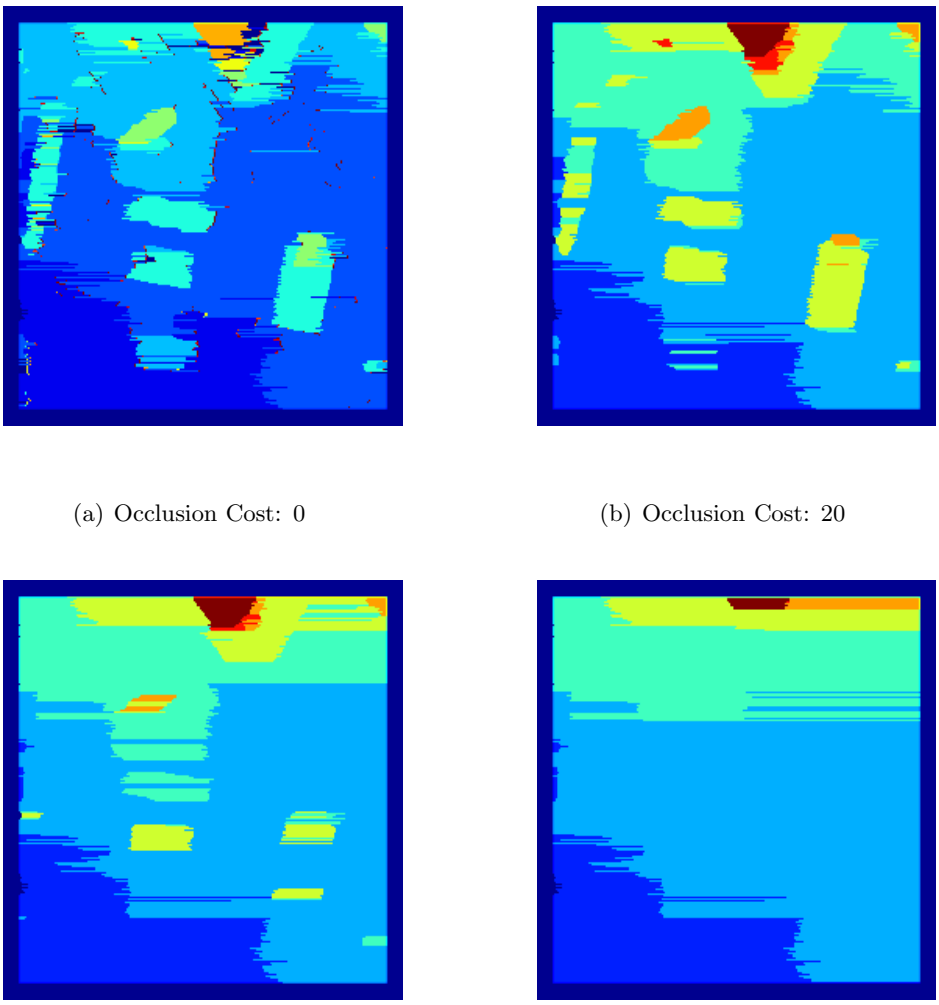
Most of the local methods described before are very sensitive to occlusions and regions with uniform texture. Global methods perform better in these cases by imposing global constraints that are less affected by local problems like occlusions and textureless zones. Two of the most used global techniques are dynamic programming and graph cuts.

3.2.1 Dynamic Programming

Dynamic programming is a general method that reduces the computational complexity of optimization problems by decomposing it into smaller subproblems, where a global cost function is defined and then minimized in stages. In the case of stereo matching, the epipolar constraint is used to decompose the problem of obtaining the disparity values in the whole image, to the one of obtaining the values at each *scanline* (recall that due to the epipolar constraint, the points in one epipolar line must lie in the corresponding epipolar line of the other one).
One of the best known dynamic programming methods is the one introduced by Bobick & Intille [36]. In this approach, scanlines are processed one at a time in the following way: for each of the scanlines, we buid a *Disparity Space Image* (DSI) by dividing the range of possible disparity values with discrete values at the precision we want to obtain. The dimension of the DSI is defined by these values that are set as the rows of the image, whereas the columns correspond with the columns of the original image. Then, at each value (i, j) of the DSI, we assign a measure of the matching between point j at image u and the point $i + j$ at image $\tilde{u}$ (remember that for each scanline we have one of these DSI, so the value j is the x coordinate of the point in the image, while the y coordinate is defined globally for all the DSI). The local cost for each point in the DSI is defined using one of the block matching measures presented before.
Once we have built the DSI, the values for all the points in the scanline are obtained by finding the minimal path from the first column to the last one (that is, processing all the scanline). Now due to the presence of occlusions, we can only move in three different directions:

1. **Horizontal**, when the disparity value of a point is the same as in the previous one.

2. **Diagonal**, that represents an increase in the disparity values, but this can also happen when we have an occlusion of a point in the reference scanline.



(a) Occlusion Cost: 0

(b) Occlusion Cost: 20

(c) Occlusion Cost: 80

(d) Occlusion Cost: 200

Figure 3.5: Dynamic Programming: Relation of the occlusion cost in the obtained disparity map. Note that with a big occlusion cost, the method tends to generate a constant disparity map, while using a small value generates noisy disparity maps. Independently of the occlusion cost, all the generated disparity maps tend to be bad estimated near object boundaries, due to the lack of coherence between scanline process.

3. **Vertical**, that represents a decrease in the disparity. This can only happens when we have occlusions in the secondary scanline (several points in the reference scanline corresponds with the same point in the secondary image, which leads to a path in the DSI to reach the next matching value).

Finally, the cost at each point is defined using an occlusion cost for those points with vertical or diagonal direction. The final cost at the point (x, d) of the DSI is defined as:

$$DSI_{occ}(d, x) = DSI(d, x) + OC$$

where $DSI(d, x)$ is the matching cost computed with one of the methods explained in Section 3.1.1. Note that if we assign a big OC, the dynamic programming method would favor the path between horizontal directions, generating a constant disparity map as is shown in Figure 3.5(d)

The main problem that we find in the dynamic programming approach is the propagation of mismatches along a scanline, that can be clearly noted in Figure 3.5[1] , near the boundaries of the objects of the scene. This is also related to the fact that each scanline is processed separately, without taking into account the existing relation between scanlines. Other approaches are presented that make use of this relation between scanlines generating a more coherent space disparity map [40].

3.2.2 Graph Cuts

As we have seen before, the major drawback of dynamic programming is the lack of relation in the process of each scanline, where the cost function is minimized at each one separately. A better approach is to minimize the cost function in the entire space at the same time. This can be expressed as the problem to find the maximum flow in a graph. The graph must be constructed in such a way that finding the maximum flow corresponds to minimize a certain energy, generally of the form:

$$E(L) = E_{data}(L) + \lambda E_{smooth}(L)$$

where $\lambda > 0$ is a weight term to define the level of smoothness we want to obtain in the final disparity map. Figure 3.6 shows the results obtained with different λ values, using again the implementation of Middlebury College.
The construction of the graph varies from one author to other. For instance [52] defines the set of vertices as

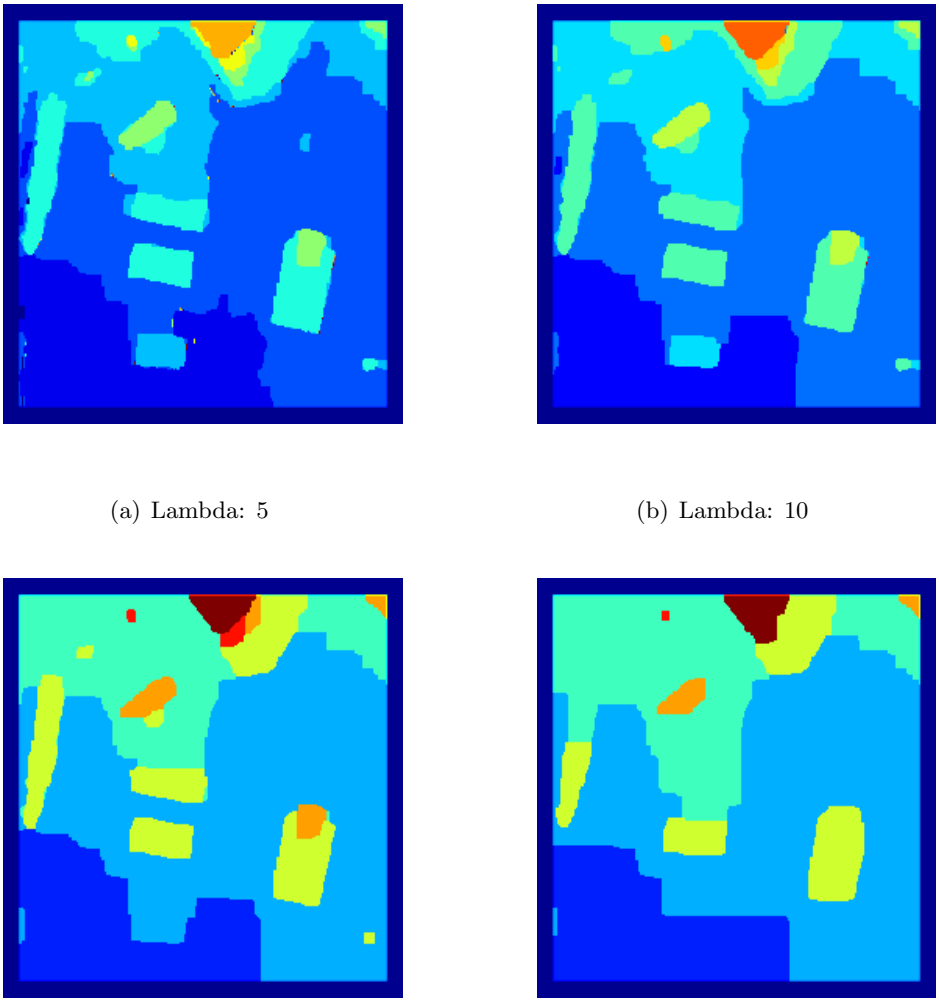
$$V = \{(x, y, d), x \in [a_1, b_1], y \in [a_2, b_2], d \in [a_3, b_3]\}$$

This is a discretization of the image domain coordinates x, y (in general already discrete) and the disparity values d, where $[a_1, b_1], [a_2, b_2]$ and $[a_3, b_3]$ are the range of x, y and d. We also incorporate the sink and the source vertex to this set. Each of these vertices has a cost associated to it that can be computed by any of the local methods described before (NCC, SSD, SAD, etc.). A set of edges is also defined to link different vertices. Each edge has an associated flow capacity that is a function of the costs of the adjacent nodes. A cut is a partition of the set V into two subsets separating the source and the sink. Each cut has an associated capacity, which is the sum of the edges capacities of the given cut. The cut with minimum capacity is the one that maximizes the flow through the graph and this is the obtained result. In [39], different graph architectures and energies are introduced. This implementation indeed is the one reported in [53] that has the better results under different data sets (all of them with disparities bigger than a pixel).

3.3 Small baseline methods

The methods presented before perform very differently when we have images that were taken with a very small baseline. Indeed, this assumption means that in general the disparity will be sub-pixel,

[1] Obtained using the implementation of Middlebury College



(a) Lambda: 5

(b) Lambda: 10

(c) Lambda: 20

(d) Lambda: 50

Figure 3.6: Graph Cuts: Disparity maps generated with different λ value. Note that for bigger values of λ, the regularization term makes the disparity map to lose important information of the objects presented in the scene.

something that most of the methods cannot cope with. In this section, we review methods that perform a sub-pixel computation of the disparity map. This is done by an exact interpolation of the images, which enables us to resample the data at any new grid. To perform this exact interpolation we are assuming that the images satisfy the Shannon Sampling Theorem. The results of the methods presented in this section are shown and discussed extensively in Chapter 7, where we compare them to each other and to the method presented in Chapter 4, which is the main subject of this work.

3.3.1 MARC - Multiresolution Algorithm for Refined Correlation

MARC is an algorithm that implements a multi window multi scale correlation, coded by Nathalie Camlong[11] and Vincent Muron[44], and is part of the CNES patent 0403143: "Appariement fin d'images stéréoscopiques et instrument dédiée avec un faible coefficient stéréreoscopique" by A. Giros, B. Rougé and H. Vadon (2004).
This multi resolution implementation enables MARC to obtain faster and more reliable results that are robust to the lack of valuable information at a certain resolution. Its multi-resolution strategy compensates one image with the disparity calculated at previous scales in the following way: suppose u_0 and $\tilde{u}_0$ are the reference and secondary images at the smaller scale that produce the disparity ε_0 (by correlation). During the change to the next scale, the secondary image $\tilde{u}_0$ is "corrected" (applying a deformation to neutralize the disparity) with an interpolated version of ε_0:

$$\tilde{u_1}(x) = \tilde{u_1}(x + \varepsilon_0)$$

The resulting image pair (u_1 and $\tilde{u}_1$) presents disparities in the subpixel range corresponding to the difference between the "real" disparities ε and the previous result, which allows us to apply the correlation again.
One of the problems of the correlation process is that it presents artifacts near the boundaries of the objects in the image, something known as *adhesion phenomenon* [19, 20].

Adhesion phenomenon

This phenomenon is an artifact strongly related to the correlation method that appears near the discontinuities of the image (that is, the boundaries of the different objects present in the scene). The correlation performs a dilation near these discontinuities, leading to a wrong estimation of the disparity map.
Given the reference image u, the secondary image $\tilde{u}$ and if we note ϵ the real disparity map, we can express the following relation:

$$\tilde{u}(x) = u(x + \epsilon(x)) \tag{3.8}$$

If we have a configuration with a very small baseline (or in a multiscale setting as explained before) we can perform first order approximations of the normalized cross correlation equation presented in Section 3.1.1:

$$\rho_{x_0}(m) = \frac{\int_{\varphi_{x_0}} u(x+m)\tilde{u}(x)dx}{\| u(x+m) \|_{2,\varphi_{x_0}} \| \tilde{u} \|_{2,\varphi_{x_0}}} \tag{3.9}$$

which leads to the *fundamental equation of correlation* [19]:

$$m(x_0) \int_\Omega d_{x_0}(x)\varphi(x - x_0)dx = \int_\Omega \epsilon(x) d_{x_0}(x)\varphi(x_0 - x)dx + \mathcal{O}(\| \epsilon \|_\infty^2) \tag{3.10}$$

that we can abbreviate as

$$m(d_{x_0} *_x \varphi) = (\epsilon d_{x_0}) *_x \varphi + \mathcal{O}(\| \epsilon \|_\infty^2)^2 \tag{3.11}$$

[2] $(d *_x \varphi)(x_0)$ is a compact notation of $\int_{\varphi_{x_0}} d(x)dx$

The function d_{x_0} is called the *correlation density* around point x_0, the center of the window φ, and is concentrated near points with high gradient:

$$d_{x_0} = \frac{\| u \|^2_{\varphi_{x_0}} - uu' \int_{\varphi_{x_0}} uu'}{\| u \|^4_{\varphi_{x_0}}} \tag{3.12}$$

Equation 3.11 relates the real disparity ϵ with the computed one m. This computed disparity is proportional to an averaged mean, by d_{x_0}, of the real disparity values ϵ in the window φ of center x_0. Solving the equation for ϵ leads to the real disparity, eliminating the adhesion phenomenon. The problem is that it is very hard to solve this equation. In [19] another approach is analyzed to correct adhesion using a *barycentric correction* which is explained below.

Barycentric Correction

This method [19] consists of approximating d_{x_0} by a *delta* function at the barycentre of the region $x_1 = \frac{\int_\varphi d_{x_0}(x) x dx}{\int_\varphi d_{x_0}(x) dx}$. The disparity value computed at x_0 is not associated to x_0 but to the barycentre x_1, obtaining the "real" disparity value at x_1: $\epsilon(x_1) = m(x_0)$.

Correlation Curvature

The second order derivative of the correlation function gives an idea of the reliability in the obtained maximum. The zero order development can be written as:

$$\rho''_{x_0}(m(x_0)) = -(d_{x_0}(u, u') *_x \varphi)(x_0) + \mathcal{O}(\| \epsilon \|_\infty) \tag{3.13}$$

Equation 3.13 is the *correlation curvature* and gives information about the points where the obtained disparity is reliable. Given the correlation curvature and the standard deviation of the image noise σ_{noise} we can impose a threshold λ to determine the zones where the disparity map ϵ has at least a precision λ:

$$N(u, \varphi, x_0) = \frac{\sigma_{noise}}{\| u \|_{\varphi_{x_0}} \sqrt{(d_{x_0}(u, u') *_x \varphi)(x_0)}} < \lambda \tag{3.14}$$

Using equation 3.14 we can obtain a mask for the values where the disparity map is accurate enough.

As a result of the whole process, MARC builds a non-dense disparity map with sub-pixel accuracy and a mask of valid points using Equation 3.14 as threshold. In fact, it also generates a "precision map" which is the values of $N(u, \varphi, x_0)$ at each point x_0. As the resulting disparity map is a non-dense one, we must interpolate the missing data using a criterion coherent with the underlying urban model.

Variational approach

The variational approach introduced in [26] is an alternative to the Barycentric Correction, which also incorporates a regularization of the existing disparity values and an interpolation in the missing ones. Here, a regularization term (minimal surface) S and a data fitting term D are used, minimizing:

$$\min_\varepsilon S(\varepsilon) + \lambda D(\varepsilon) \tag{3.15}$$

The data term forces the final solution to remain close to the data points according to the selected λ while the regularization term diffuses the information present at the valid points while imposing regularity on the final solution.
In [4], some options for the implementation of each of these terms are discussed. Here, we present the choice that produces the best results, as was concluded in [4], which is a minimal surface with global data fit. Thus, the data term is defined as:

$$D_1(\varepsilon) = \sum \sqrt{a^2 + (\varepsilon - m)^2} \tag{3.16}$$

It is similar to the more robust L^1 norm of the error $\varepsilon - m$, but the small a parameter avoids its non-differentiability at 0.

The regularization term is an anisotropic regularization, that avoids regularization across the highly contrasted level lines of the image u:

$$S_3(\varepsilon) = \sum \sqrt{\beta^2 + \left| \nabla\varepsilon - c < \nabla\varepsilon, z > \frac{z}{|z|} \right|^2} dx \tag{3.17}$$

with $0 \leq c \leq 1$ and the vector field z defined as

$$z = \frac{\nabla u}{\sqrt{b^2 + |\nabla u|}}$$

The heuristic that leads to such a regularization term is the same as the lambertian hypothesis explained in the abstract.

3.3.2 Region-based Affine Motion Estimation

The motion estimation algorithm explained in this section is an Optical Flow method, and it was developed by Caselles, Garrido & Igual [14, 46]. It is based on a functional that matches the gradient orientation between the images (thus being contrast invariant). The techniques of motion estimation searches for the displacement of the objects in the scene along a sequence of frames. In particular these estimation methods can be applied between the two frames of a stereo pair to determine the apparent motion of the objects.

Let Ω be the image domain and $I : \Omega \to R$ be a given image. Mathematical morphology offers a complete image description in terms of its level sets. Level lines can be defined as the boundaries of the level sets and the family of level lines is a basic contrast invariant geometric description of the image I.

Let denote by ϕ the disparity map computed between two images I_0 and I_1. For notation purposes, $\phi = (\phi_1, \phi_2)$, where ϕ_1 and ϕ_2 are the components of ϕ. Let $X = (x(s), y(s))^T$ be the arclength parameterization of a given level line $\mathcal{C}$ of the image $I_0(\mathbf{x})$, s being the arc length parameter. The curve $\mathcal{C}$ may be described by its normal vectors $Z = (-y'(s), x'(s))^T$, where $(.)'$ denotes the first derivative with respect to s. Let us describe the normal vectors to the curve $\phi(\mathcal{C})$ in terms of ϕ and the normal vectors to $\mathcal{C}$. Since the curve $\phi(\mathcal{C})$ is described by $X_\phi = (x_\phi(s), y_\phi(s))^T = \phi(x(s), y(s))$, its tangent vector is given by $X'_\phi = D\phi\ X'$, $D\phi$ being the differential of ϕ. Thus, the normal vector Z of the deformed curve is

$$Z_\phi = \begin{pmatrix} -y'_\phi(s) \\ x'_\phi(s) \end{pmatrix} = \begin{pmatrix} \partial_y\phi_2 & -\partial_x\phi_2 \\ -\partial_y\phi_1 & \partial_x\phi_1 \end{pmatrix} \begin{pmatrix} -y'(s) \\ x'(s) \end{pmatrix} \tag{3.18}$$

where ∂_x and ∂_y denote the partial derivative with respect to x and y respectively. The matrix in the right hand side of (3.18) is the *cofactor matrix* associated to $D\phi$ which we denote by $(D\phi)^\dagger$. We normalize the vector Z_ϕ to be of unit norm by redefining

$$\overline{Z}_\phi = \frac{(D\phi)^\dagger\ Z}{\|(D\phi)^\dagger\ Z\|} \quad \text{if } (D\phi)^\dagger Z \neq 0; \quad 0 \text{ otherwise,} \tag{3.19}$$

where $\|.\|$ denotes the modulus of a vector in R^2 .

Usually, the energy functional whose minimum gives the disparity tries to impose the brightness constancy assumption which in turns leads to the optical flow equation (Eq **??**). Instead, the authors use another assumption: that shapes move with possible deformation between the two images. Thus, the idea is to align the level lines of the images by a map ϕ. This map is obtained by aligning the unit normal vector field $Z^1(\mathbf{x})$ to the level lines of I_1 with the transformed vector field of $Z(\mathbf{x})$ by the map ϕ (i.e., the vector field $\overline{Z}_\phi$). Finally, the disparity map is obtained by minimizing the energy functional

$$E(\phi) = \int_\Omega \rho\left(\| Z^1(\phi(\mathbf{x})) - \overline{Z}_\phi(\mathbf{x}) \|^2 \right) dx\, dy. \tag{3.20}$$

where ρ is some robust function [3].

Moreover, we assume that the disparity fields can be expressed locally by a particular model and we follow a region-based strategy to minimize Eq. (3.20). We consider a partition $\mathcal{R}$ into disjoint connected regions of the image I_0 bounded by level lines. The partition is computed with the segmentation algorithm *Mumford-Shah functional subordinated to the level lines of the image* [46].

We discretize the functional (3.20) as follows:

$$E_{\mathcal{R}}(\phi) = \sum_{j=1}^{N_R} \sum_{\mathbf{x}\in R_j} \rho\left(\| Z^1(\phi(\mathbf{x})) - \overline{Z}_{\phi}(\mathbf{x}) \|^2 \right), \tag{3.21}$$

where N_R is the number of regions of the partition $\mathcal{R}$.

If we suppose the images to be rectified, we can assume the same motion model presented in Equation 2.10:

$$\phi(\mathbf{x}) = \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ 0 \end{pmatrix} \tag{3.22}$$

where e is the translation parameter along the x axis and a, b are the parameters that model the linear transformation (thus, including scaling, and shearing along the x axis). No restriction on the numerical values of the parameters is imposed. The minimization is carried out using a conjugate-gradient technique applied over E_R for each region. Moreover a multiresolution scheme is applied. For more details on this issue we refer to [46].

3.4 A brief discussion about the presented methods

In this chapter we have reviewed the most well known matching methods based on the big baseline approach. We have also explained how Normalized Cross Correlation (Section 3.1.1) and Optical Flow (Section 3.1.2) have been successfully adapted to obtain sub-pixel disparity maps.
In [53], methods for images with disparities bigger than a pixel are analyzed, and the conclusion is that in general Graph Cuts performs the best results. Thus, we can wonder whether the implementation of subpixel accuracy disparity maps can be obtained using Graph Cuts. This has not been tackled yet, so we cannot be sure whether the same results will be valid for subpixel processing. However, in [40] it is shown that Graph Cuts has problems to estimate tilted planes. Existing implementations have also another drawback which is that in order to compute subpixel disparity values, we have to work with very big images, making the execution of these implementations impractical in terms of time and computational cost. Recent works present efficient implementations of the graph cuts based on binary search [15, 17, 18] which may eventually avoid this drawback.

In general, global methods have better results because they deal better when we have occlusions, which is not the case we are studing. On the other hand, local methods are well adapted to the case of sub-pixel computation (Correlation and Optical Flow). The results obtained using both techniques are deeply analyzed in Chapter 4.

One of the characteristics of images taken from urban scenes is that most of the objects present on them (except for natural structures like trees) can be modeled by affine transformations. If we have a way to decide whether an affine model is valid or not within each object, we would obtain a more descriptive interpretation of the scene. This is the motivation to study and develop the region-merging piecewise affine approach presented in the following chapter.

[3] A function is robust if it is symmetric, positive-definite with a unique minimum at zero.

Chapter 4

An *a contrario* affine region merging algorithm

Most of the existing stereo matching techniques generate a non dense disparity map. That is, there are some points, labeled as not valid, in which we cannot estimate the disparity value. In these cases, to obtain a dense disparity map, we must interpolate the valid data, which also improves the final result by reducing the noise presented in the valid disparity values.

In this section, we propose an approach which improves the disparity map, and achieves a better definition of the different elements present in the stereo pair. At the same time, disparity boundaries are aligned with those of real objects in the scene, since the initial partition is computed using intensity image information or geometry information of the reference image.

This algorithm is based on a region-merging approach [30]. We consider an initial segmentation $\mathcal{R}$ of the reference image in a set of connected and disjoint regions that we denote $R_1, ..., R_N$. Since the images we are dealing with are taken from urban scenes, they are generally composed by roofs and other planar elements. The disparity values of these structures can then be modeled by affine transformations. If we assume that the disparity of each region follows a three-parameters affine model of the form:

$$T(x, y) = ax + by + c.$$

we can obtain the affine transformations T_i that best fits the data at each region R_i as follows:

$$T_i = \arg\min_T \sum_{x \in R_i^*} \rho(T(x) - d(x)) \tag{4.1}$$

where d is the original disparity map we want to interpolate, R_i^* are the valid points [1] of region R_i and ρ is a function with the following characteristics: is symmetric, positive-definite, with a unique minimum at zero, and it is generally less increasing than the square function. Such a function is known as *robust* function.
There are several estimation techniques to minimize Eq 4.1, some of them less sensitive to the outliers present in the original data than the others. See Appendix A for a review of these functions and estimation techniques.

In our approach, "coherent regions" are iteratively merged together. By "coherent regions" we mean those neighboring regions with a very similar affine transformation (which means that in fact the regions are part of the same structure). The way we decide if two regions can be merged or not is based on a significance measure: the *number of false alarms* (NFA) which comes from the definition of an *a contrario* model. In the next section, we summarize the basic concepts related to the definition of an *a contrario* model and the number of false alarms.

4.1 A Review on Computational Gestalt Theory

Computational Gestalt Theory was first presented by Desolneux, Moisan and Morel [21, **?**, 23, 24] as a way to obtain a quantitative theory of the Gestalt laws. Gestalt theory [37] states that visual perception is a grouping process where geometric objects are grouped together by similar characteristics or *gestalts* (color, size, shape, etc). Although the Gestalt theory is consistent from a qualitative point of view, it lacks of a quantitative way to determine when a set of objects have the same gestalt. The approach presented in [21] uses the *Helmholtz Principle* to define a quantitative measure of a given gestalt:
Helmholtz Principle - Assume that n objects $O_1, O_2, ..., O_n$ are present in an image. Suppose that k of them have a common feature (for example, same color, same size, etc). We want to know whether this feature is present by chance in k objects or if it is meaningful enough to group these objects together. So, we assume *a priori* that the feature has been randomly and uniformly distributed throughout all objects $O_1, ...O_n$. Then, we assume that the observed objects are a random realization

[1] The transformation is estimated using only the valid points of region R_i, not all the points present at region R_i.

of this uniform process. We finally ask the question: is the observed distribution probable or not in our model?. If not, this proves *a contrario* that these objects must be grouped together.

The Helmholtz principle can be formalized by the definition of an ϵ-meaningful event:

Definition 1 (ϵ-meaningful event) *We say that an event of the type "a given configuration of objects has a property" is ϵ-meaningful if the expectation of the number of occurrences of this event is less than ϵ under the uniform random assumption.*

Definition 2 (Number of false alarms - NFA) *Given an event of the type "a given configuration of objects has a property", the number of false alarms (NFA) is the expectation of the number of occurrences of this event under the uniform random assumption.*

Definition 1 can be rewritten in terms of the NFA defined before:

Definition 3 (ϵ-meaningful event) *An event E of the type "a given configuration of objects has a property" is ϵ-meaningful if the NFA is less than ϵ:*

$$\mathrm{NFA}(E) < \epsilon \tag{4.2}$$

Let H_1 be the background model: "a given configuration of objects has a property and is produced by chance". We define a random variable $\mathcal{E}$, and we analyze the observation E of this random variable considering the number of false alarms. A correct definition of this NFA is the central problem in all a contrario methods. However, quite often this definition can be reduced to an expression of the following form which gives an upper bound of the actual NFA as defined before

Definition 4 (Number of false alarms - NFA) *The number of false alarms (NFA) of an event E is defined as:*

$$\mathrm{NFA}'(E) = \mathcal{N} \cdot \mathrm{P}[\mathcal{E} \geq E | H_1] \tag{4.3}$$

where $\mathcal{N}$ is the number of possible configurations of the event E.

Moreover, we can often show that the expectation of the number of occurrences of an event E satisfying $\mathrm{NFA}'(E) < \varepsilon$ is actually less than ε (see [25] for a proof of this in the case of alignment detection, or later in section 4.6 for a proof in our case). For this reason, defining an event as ϵ-meaningful, whenever $\mathrm{NFA}'(E) < \epsilon$, is still consistent with the original definition 1 and ensures that the method is robust in the sense that no more than ϵ "false detections" will be obtained due to noise.

Recall that $\mathrm{P}[\mathcal{E} \geq \delta | H_1]$ is a decreasing function of δ [25]. Thus, for a given event E, we reject H_1 if the $\mathrm{P}[\mathcal{E} \geq \delta | H_1]$ is small. We express this using the NFA: we reject H_1 if $\mathrm{NFA}(R) < \epsilon$ which in turns means that the event could not be explained by the background model. The reader is referred to [25] for a complete review on Computational Gestalt Theory.

In order to implement the region-merging algorithm, we need to define three concepts: the region model, the merging criterion and the merging order.

4.2 The region model

The merging procedure has two inputs: the initial disparity map and a segmentation of the reference image. Note that this segmentation is independent of the disparity map and can be computed using any image segmentation algorithm.[2]
For each region R_i of the initial segmentation $\mathcal{R}$, we approximate the disparity values d on it, by an affine transformation:

$$T_i(x, y) = ax + by + c$$

[2]In Chapter 6 we discuss the segmentation techniques used in the experiments and the relation of these segmentations in the final performance of the algorithm.

Meaningfulness of the model

The estimation of the transformation at R_i is obtained by the minimization defined in Eq. (4.1). We cannot ensure that the minimum corresponds to the correct model associated to the region. In fact, the affine model is appropriate only for objects that can be approximated by planes (trees for instance, could not be modeled by affine transformations). Thus, we present an approach for measuring how well the estimated transformation adjusts the disparity values of a region and it is based on the *a contrario* approach presented in Section 6.

Let T_i be the computed transformation at region R_i and n the number of valid points in this region. Given a precision θ, we say that the transformation T_i fits the disparity map d at a point $\mathbf{x} \in R_i$ if

$$|T_i(\mathbf{x}) - d(\mathbf{x})| \leq \theta \tag{4.4}$$

If we define an independent random variable associated to the disparity measure d, the probability of having at least k points among n satisfying the condition in Eq. (4.4) is the tail of the binomial distribution:

$$B(k, n, p) = \sum_{j=k}^{n} C_j^n p^j (1-p)^{n-j}$$

where p is the probability of having a point $\mathbf{x}$ that satisfies the condition (4.4) over the *a contrario* model, and k is the number of points that satisfy this condition:

$$k = \#\{\mathbf{x} \in R_i \ / \ |T_i(\mathbf{x}) - d(\mathbf{x})| < \theta\}$$

In our implementation, the *a contrario* model was obtained as the histogram of the error values. That is, for each transformation within a region, we compute the error at each point and we build an histogram with all these error values. Now, given a transformation T and a region R we want to know whether the transformation that fits the values at region R can be explained by this background model or not. We define the number of false alarms of a region and a transformation as follows:

Definition 5 *The number of false alarms (NFA) of (R, T) is defined as:*

$$NFA(R, T) = N_{tests} B(p, k, n) \tag{4.5}$$

where N_{tests} is the number of all possible configurations we can have for the pair R, T or in other words, all the transformations we test at region R.

Given a threshold ϵ, we say that the observation (R, T) is ϵ-meaningful if the number of false alarms is less than ϵ, i.e., if

$$NFA(R, T) \leq \epsilon.$$

We finally define the region model of a region R_i as the $NFA(R_i, T_i)$.

To state that Definition 5 is an upper bound of the number of false alarms given at Definitions 1 and 2, we must prove that the expectation of the event : "the number of false alarms of (T, R) is less than ϵ" is less than ϵ. For doing so, we need a formal definition of the Number of Tests in Equation 4.5, so it is delayed until Section 4.6.

4.3 Merging criterion

Given two regions, we must have a criterion to decide wether we merge them or not. This merging criterion is defined as follows:

$$NFA(R_i \cup R_j, T_{ij}) \leq NFA((R_i, T_i), (R_j, T_j)) \tag{4.6}$$

We compare the NFA of the union of both regions and its new estimated transformation, with the NFA of having each region separately, each one with a different transformation. This condition has the following interpretation:

Interpretation:

Given two regions R_i and R_j, we want to know whether to consider both regions as a new one, with a new estimated transformation T_{ij}, is more meaningful than to keep them separately, each one with its own transformation (see Figure 4.1). For that, we define the following random variables, $\mathcal{E}_{\text{joint}}$ and $\mathcal{E}_{\text{sum}}$ in the following way:

$\mathcal{E}_{\text{joint}}$ measures the number of points at $R_i \cup R_j$ that fit transformation T_{ij}. We note this number of points as k_{ij}.

$\mathcal{E}_{\text{sum}}$ measures the number of points of region R_i that fits transformation T_i and the number of points of R_j that fits transformation T_j. We note the number of points as k_i and k_j respectively.

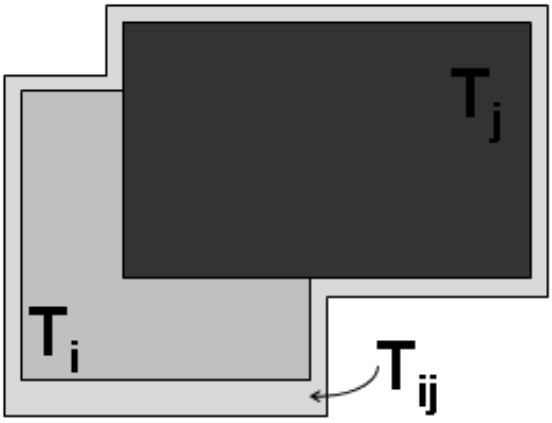


Figure 4.1: A simple model, considering one transformation associated to both regions, versus a more complex one, where we have two transformations, one for each region.

For the adjacent regions R_i, R_j we have to compare $\text{NFA}(R_i \cup R_j, T_{i,j})$ with $\text{NFA}((R_i, T_i), (R_j, T_j))$, which is the number of false alarms of having one different transformation for each of the regions. We define this new number of false alarms as follows:

Definition 6 (Number of false alarms of two regions) *The number of false alarms (NFA) of having two regions R_i and R_j, each one with a different associated transformation (T_i and T_j), is defined as:*

$$NFA((R_i, T_i)(R_j, T_j)) = N'_{tests} P((R_i, T_i)(R_j, T_j)) \tag{4.7}$$

where N'_{tests} is the number of all possible configurations we can have for each of the pairs (R_i, T_i), (R_j, T_j) and the probability $P((R_i, T_i)(R_j, T_j))$ is the probability of having two regions, each one with a different transformation, in the a contrario *model.*

The probability $P((R_i, T_i), (R_j, T_j))$ of the event "transformation T_i fits region R_i and transformation T_j fits region R_j" is obtained by taking into account all points $n_i + n_j$ as possible points and $k_i + k_j$ as the points that satisfy Eq. 4.4:

$$P((R_i, T_i), (R_j, T_j)) = \mathcal{B}(k_i + k_j, n_i + n_j, p).$$

In Section 4.6 we will show that the number of tests of having one transformation for both regions (N_{tests} of Eq 4.5), is not the same of having one transformation for each of the regions (N'_{tests} of Eq 4.7). Indeed, N_{tests} is smaller than N'_{tests}, which is a kind of balance between having a simple model (one different transformation for each region) which better fits the data, or a simpler, more regular one (one transformation for both regions) which fits the data less well but still good enough to comensate for the simplicity of the model.
The final merging criterion is:

$$N_{Tests} * \mathcal{B}(k_{ij}, n_{ij}, p) \leq N'_{Tests} * \mathcal{B}(k_i + k_j, n_i + n_j, p) \tag{4.8}$$

where N_{Tests} is the number of test of having one model (T_{ij}) and N'_{Tests} the number of tests of having two different models (T_i,T_j). The merging criterion is similar in a certain sense to a variational approach, where $\log(N_{tests})$ plays the role of the regularization term and $\log(P)$ plays the role of the data fitting term.

Now, it is clear that the definition of the joint probability of having two transformations, one for each region, is crucial to the merging criterion. This problem was first addressed for clustering problems in [45, ?] in the case where both regions R_i and R_j may share same points, leading to a trinomial distribution because of the non-independence of the events. Under certain hypotheses, this trinomial distribution can be approximated with a term that is more easy to compute.
This is not the case of the region merging where the neighbouring regions are disjoint. Thus, we have to look for another solution. In [57] the merging problem is being studied in the context of multisegment detection, and the following expression is considered as the "ideal" joint probability, assuming that both regions are independent with a binomial distribution under the *a contrario* model:

$$P((R_i,T_i)(R_j,T_j)) = \sum_{(k'_i,k'_j):B(p,n_i,k'_i)B(p,n_j,k'_j)\leq B(p,n_i,k_i)B(p,n_j,k_j)} b(p,n_i,k'_i)b(p,n_j,k'_j) \tag{4.9}$$

where b is the binomial distribution and B id the binomial tail. The idea is to consider as more meaningful than the observed event all the (k'_i,k'_j) which have a smaller joint tail B. Then we consider as joint probability the sum of probabilities b of all those events that are meaningful or more meaningful than the observed one.
In one dimension this reduces to a simple threshold problem, but in two dimensions the geometry of this region is unknown and numerical computation of Eq. 4.9 can be quite complicated. For this reason the authors considered a lower bound of Eq. 4.9:

$$P((R_i,T_i)(R_j,T_j)) = B(p,n_i,k_i)B(p,n_j,k_j) \tag{4.10}$$

and an approximation:

$$P((R_i,T_i)(R_j,T_j)) = B(p,n_i,rn_i)B(p,n_j,rn_j) \tag{4.11}$$

where

$$r = \min(\frac{k_i}{n_i},\frac{k_j}{n_j})$$

which is based on imposing a "balanced" data fit between both regions to avoid that the joint probability be dominated by one of the two terms.

In our context, we found that our criterion gives better results than either the ones defined using Eq. 4.10 or Eq. 4.11, and since we have a simpler explanation in terms of comparing a simple losely fit model with a more complex tightly fit model, we kept the new merging criterion as defined in Eq. 4.8.

4.4 The merging order

To define the order in which we want to process the regions to be merged, we need to know how we can merge two regions: we can merge only neighboring regions, since the goal of the merging process is to merge regions that belongs to the same physical object. Thus, we consider all possible pairs of adjacent regions. For each of these pairs, we estimate the transformation associated to the region defined by the union of both regions in the pair, and then we compute the NFA of this new region. Each of these pairs with its NFA is inserted in a priority queue, ordered by NFA (the lowest NFA is the first one). The merging order is defined as the lowest NFA: we process first the pair of regions with the lowest NFA, which is the first pair in the priority queue.

4.5 Merging Procedure

The merging procedure can be summarized as follows:

Algorithm sketch

1. Build a graph $\mathcal{G} = \langle V, A \rangle$ where a node $v_i \in V$ represents a region R_i of the image partition $\mathcal{R}$, and an edge $a_{i,j} \in A$ exists if regions R_i and R_j are adjacent.
2. For each region R_i, $i = 1, \ldots, n$ estimate the transformation T_i and compute the $NFA(R_i, T_i)$.
3. For each joint region $R_i \cup R_j$, where R_i and R_j are adjacent regions, estimate the transformation T_{ij} and compute the $NFA(R_i \cup R_j, T_{ij})$.
4. Build a priority queue of joint regions $R_{ij} = R_i \cup R_j$ ordered by NFA.
5. Iterate until the queue is empty:
   (a) Take the first element and remove it from the queue. This element is the pair with the lowest NFA.
   (b) If the pair satisfies the merging criterion defined in Eq. (4.6) then:
      i. Remove all of the entries in the queue with one of these two regions.
      ii. Reestimate the transformations and the NFA for the new region with all its neighbors.
      iii. Insert each of these new pairs with its NFA into the queue.
   (c) If the merging criterion is not satisfied, ignore this merge.
   (d) Go to step (a).

The merging procedure presented so far can be seen as an implementation of the *iterative exclusion principle* [25]. The exclusion principle can be defined in the following generic form [25]:

Definition 7 (Exclusion Principle) *Let A and B be two groups of objects obtained considering the same characteristic (gestalt law). Then no object is allowed to belong to both A and B. In other terms, each object must either belong to A or B.*

The exclusion principle is presented in [25] in the form of an algorithm (the iterative exclusion principle). Particular implementations of this iterative exclusion principle to detect alignments [5, 25] and vanishing points [3] have been developed with success. The merging process presented in this section can be seen as an implementation of this Exclusion Principle Algorithm to the case of region-merging.

Once we have introduced the general sketch of the region-merging algorithm, we can compute the number of tests of Definition 5, since it is strongly related to the merging process itself.

4.6 Number of tests

The merging algorithm can be divided in two parts: first of all, it performs an initialization (steps 1 to 4 of the algorithm sketch): it estimates the transformation associated to each region in the initial segmentation, testing all the possible transformations. Then, for each of the possible pair of neighbors, it estimates the transformation of their union and computes the NFA associated to them. Let N_{transf} be the number of transformations we test at each region, N the number of regions in the initial segmentation, and M the number of neighbor region pairs. The number of test of the first step of the algorithm is:

$$N_{test}^{(1)} = N_{transf}(N + M)$$

where we have assumed that the number of transformations N_{transf} is the same for all regions. If we define $\bar{C}$ as the mean of the number of neighbors of each region, then the number of tests of the first part of the algorithm can be approximated by

$$N_{test}^{(1)} = N_{transf} N(\bar{C} + 1)$$

In the second part of the algorithm, we perform an iterative process where at each step we merge two neighbor regions. Once we have them, we must reestimate the transformations for the new region and all its neighbors. The number of tests at each iteration is

$$N_{test}^{(i)} = C_{r_i} N_{transf}$$

where C_{r_i} is the number of neighbors of the new region. This process is repeated until no pair can be merged. The total number of iterations is bounded by $N - 1$ because we have N initial regions and at each iteration we merge only two of them.
We summarize the number of tests as follows:

$$N_{tests} = \begin{cases} (N+1)\bar{C}N_{transf} & \text{in the initialization step} \\ C_{r_i} N_{transf} & \text{for iteration } i = 1..N-1 \end{cases}$$

One important observation is that we are computing the number of false alarms of a region R and a transformation T considering the whole merging process. That is, we want to know whether (R, T) is meaningful after the merging process was done. This is not the same of the number of false alarms of a transformation T and a region R without a region-merging process. Indeed, in the last case, the number of test would be simply $N_{tests} = N N_{transf}$.

If we take into account the whole region merging process, we can derive the conditions for the initialization step:

$$(N+1)\bar{C}N_{transf} B(p, k_0, n_0) \leq \frac{\epsilon}{2}$$

and for each step i of the iteration:

$$C_{r_i} N_{transf} B(p, k_i, n_i) \leq \frac{\epsilon}{2N}$$

Finally, if we use the mean of the number of neighbors for all the steps, the final number of tests at each step can be approximated by

$$N_{test}^{(i)} = 2(N+1)\bar{C}N_{transf}$$

The way in which we have defined the number of tests is not an arbitrary one. Indeed, with this definition we can now prove that Definition 5 is an upper bound of the NFA given at Definition 4:

Proposition 2 *The expectation of the number of ϵ-meaningful fits taking into account the whole region-merging process is less than ϵ, under the assumption that all transformations are generated by the* a contrario *model.*

Proof. Let's start by formalizing the number of ϵ-meaningful fits. Let $\chi_{ij}^{(n)}$ be the indicator function of the event $e_{ij}^{(n)}$: "Transformation T_i and region R_j is an ϵ-meaningful fit at step n".Now, recall that depending on the step of the merging process, the number of tests to be done is different: in the initialization step of the region-merging, we test all the transformations at each region and at each pair of neighbor regions. Let $S^{(1)}$ be the number of ϵ-meaningful fits in the first step:

$$S^{(1)} = \sum_{i=1}^{N+M} \sum_{j=1}^{N_{tranf}} \chi_{ij}^{(1)}$$

where N is the number of regions in the initial segmentation, M is the number of all the neighboring pairs and N_{transf} the number of all transformations we test at each region (we assume that the

number of transformation to test is the same for all regions).

During the iteration process of the region merging algorithm, we test the transformations only between the new merged region and its neighbors. If we note C_n the number of neighbors of the new region at step n, the number of ϵ-meaningful fits is:

$$S^{(n)} = \sum_{i=1}^{C_n} \sum_{j=1}^{N_{tranf}} \chi_{ij}^{(n)}$$

The number of ϵ-meaningful fits is summarized by $S = \sum_{n=1}^{N} S^{(n)}$.
Now, in order to prove the proposition, we have to demonstrate that:

$$\mathbb{E}[NFA(R,T) \leq \epsilon] = \mathbb{E}[S] \leq \epsilon$$

Because of the linearity of the expectation we have:

$$\mathbb{E}[S] = \mathbb{E}[\sum_{n=1}^{N} S^{(n)}] = \sum_{n=1}^{N} \mathbb{E}[S^{(n)}] = \mathbb{E}[S^{(1)}] + \sum_{n=2}^{N} \mathbb{E}[S^{(n)}] \tag{4.12}$$

The condition we have to prove is then:

$$\mathbb{E}[S^{(1)}] + \sum_{n=2}^{N} \mathbb{E}[S^{(n)}] \leq \epsilon \tag{4.13}$$

Applying again the linearity of the expectation in the first term of Eq 4.13 we obtain:

$$\mathbb{E}[S^{(1)}] = \mathbb{E}[\sum_{i=1}^{M} \sum_{j=1}^{N_{tranf}} \chi_{ij}^{(1)}] = \sum_{i=1}^{M} \sum_{j=1}^{N_{tranf}} \mathbb{E}[\chi_{ij}^{(1)}]$$

The expectation of an indicator function is the probability of the event that we have:

$$\mathbb{E}[S^{(1)}] = \sum_{i=1}^{M} \sum_{j=1}^{N_{tranf}} \mathbb{P}[e_{ij}^{(1)}]$$

Recall that the events we are analyzing are those that are ϵ-meaningful. Then by definition, we have

$$\mathbb{P}[e_{ij}^{(1)}] \leq \frac{\epsilon}{2MN_{transf}}$$

and finally

$$\mathbb{E}[S^{(1)}] = \sum_{i=1}^{M} \sum_{j=1}^{N_{tranf}} \mathbb{P}[e_{ij}^{(1)}] \leq \sum_{i=1}^{M} \sum_{j=1}^{N_{tranf}} \frac{\epsilon}{2MN_{transf}} = \frac{\epsilon}{2}$$

Thus, the first term of Equation 4.13 is bounded by $\frac{\epsilon}{2}$.

For the second term of Equation 4.13, we have:

$$\sum_{n=2}^{N} \mathbb{E}[S^{(n)}] = \sum_{n=2}^{N} \sum_{i=1}^{C_n} \sum_{j=1}^{N_{tranf}} \mathbb{E}[\chi_{ij}^{(n)}] = \sum_{n=2}^{N} \sum_{i=1}^{C_n} \sum_{j=1}^{N_{tranf}} \mathbb{P}[e_{ij}^{(n)}]$$

Recall that by definition:

$$\mathbb{P}[e_{ij}^{(n)}] \leq \frac{\epsilon}{2NC_nN_{transf}}$$

So,

$$\sum_{n=2}^{N} \mathbb{E}[S^{(n)}] \leq \sum_{n=2}^{N} \sum_{i=1}^{C_n} \sum_{j=1}^{N_{tranf}} \frac{\epsilon}{2NC_nN_{transf}} = \sum_{n=2}^{N} \frac{C_nN_{tranf}\epsilon}{2NC_nN_{transf}} = \frac{\epsilon}{2}$$

Finally, the second term of Equation 4.13 is also bounded by $\frac{\epsilon}{2}$, which completes the proof. □

4.6.1 Reformulation of the merging-condition

Once we have defined the number of tests performed to evaluate whether a transformation fits a region on a ϵ-meaningful way taking into account the whole region-merging process, we can reformulate the merging criterion. To do so, we need to know the number of tests done when we have the more complex model of two different transformations for each pair of regions. This number of tests is indeed the same one of having one model, by a factor of N_{transf} since in this case we test all possible transformations at each of the regions. Then, the number of tests done when we have one transformation for each region is:

$$N'_{tests} = 2(N+1)\bar{C}N^2_{transf}$$

With this number of tests, we can reformulate the merging criterion. Recall that the merging criterion is

$$NFA(R_i \cup R_j, T_{ij}) \leq NFA((R_i, T_i), (R_j, T_j))$$

The NFA of the joint region (the left term) is computed as described before:

$$NFA(R_i \cup R_j, T_{ij}) = 2(N+1)\bar{C}N_{transf}P(R_i \cup R_j)$$

The NFA of having both regions separately (the right term) is:

$$NFA((R_i, T_i), (R_j, T_j)) = 2(N+1)\bar{C}N^2_{transf}P((R_i, T_i), (R_j, T_j))$$

The final region-merging criterion is:

$$P(R_i \cup R_j, T_{ij}) \leq N_{transf}P((R_i, T_i), (R_j, T_j)) \tag{4.14}$$

The term N_{transf} in the final merging criterion can be interpreted as the cost of having a more complex model. Without this term, the region criterion would have never been satisfied:
Let R_i, R_j be two regions, both with n_i and n_j points. Let k_i and k_j be the number of points that satisfy Eq 4.4 at each region. When we consider the joint region $R_{ij} = R_i \cup R_j$, the number of points is the sum of the points at both regions: $n_{ij} = n_i + n_j$. Nevertheless, the number of valid points in the joint region is always less than or equal to the sum of the valid points at each region: $k_{ij} \leq k_i + k_j$. Thus:

$$B(n_{ij}, k_{ij}, p) = B(n_i + n_j, k_{ij}, p) \geq B(n_i + n_j, k_i + k_j, p)$$

and the merging condition is never reached. This proves that without the term N_{transf}, no merging would be done.

Estimation of the number of transformations

It only remains to estimate the number of transformations we can test at each region. This can be done by considering that each transformation is defined by three points. Suppose that we know that the range of the disparity values is $[d_{min}, d_{max}]$. Then, these points have values in that range and if we define a discretization step s, then we have the following number of possible points:

$$\frac{d_{max} - d_{min}}{s}$$

which leads to the following number of transformations:

$$N_{transf} = \left(\frac{d_{max} - d_{min}}{s}\right)^3$$

Note that the discretization step s is in fact the precision we want to obtain at the final disparity map.

Chapter 5

A continuous formulation of the number of false alarms

The definition of the ϵ-meaningful event "transformation T fits region R" proved in the previous chapter is discrete, since the random variables were defined as Bernoulli random variables: if the point satisfies Eq. 4.4 then the random variable takes the value 1. Otherwise, the value is 0.
In this chapter we present a different approach to define this event, using a set of continuous random variables defined between 0 and 1 instead of the Bernoulli ones. This redefinition of the event enables us to incorporate more information in the computation of the number of false alarms, which in turns leads to a more accurate model.
In section 5.1 we present this new event and we derive the density probability function associated to it. In order to obtain this density function, we need to determine the density function associated to the robust function estimator used to obtain the transformation, which is done in Section 5.2. Section 5.3 presents a reformulation of the number of false alarms given in the previous chapter, using the new event presented in Section 5.1. Finally, in Section 5.4 we approximate the number of false alarms using one of the Hoeffding Inequalities.

5.1 Redefinition of the fitting event

Recall Definition 5 of the number of false alarms of a region R and a transformation T given in Chapter 4:

$$NFA(R,T) = N_{tests}B(k,n,p)$$

where n is the number of valid points at region R, k is the number of points $\mathbf{x}$ that satisfy:

$$e_{\mathbf{x}} = |T(\mathbf{x}) - d(\mathbf{x})| < \alpha$$

for a given threshold α, and p the probability that a point $\mathbf{x}$ satisfies $e_{\mathbf{x}} < \alpha$ in the background model. Note that in this context, all points $\mathbf{x}$ that satisfy $e_{\mathbf{x}} < \alpha$ have the same relevance (weight), independently of the error value: we count all points with an error less than α as valid. A better approach would be to incorporate the error measures $e_{\mathbf{x}}$ in the definition of the *a contrario* model: given a region R and a transformation T, instead of considering the event "k points among n have an error less than α", we consider the event "the transformation error is less than θ". So, if we define a random variable X associated to the points $\mathbf{x}$, the probability of this new event can be defined as

$$P(\frac{1}{|R|}\sum_{X\in R}|T(X) - d(X)| \leq \theta) \tag{5.1}$$

Consider now a random variable $E_{\mathbf{x}}$ associated to the error measures $e_{\mathbf{x}}$, and assume that we know the distribution of this random variable (we will discuss this in the next subsection). Let f_e be the density distribution of the random variable $E_{\mathbf{x}}$. To be consistent with the robust method used to estimate the transformation, we use the same robust function[1] ρ to measure the error. The probability of Eq 5.1 can be expressed in terms of the ρ function:

$$P(\frac{1}{|R|}\sum_{X\in R}\rho(T(X) - d(X)) \leq \theta) = P(\frac{1}{|R|}\sum_{X\in R}\rho(E_{\mathbf{x}}) \leq \theta) = P(\sum_{X\in R}\rho(E_{\mathbf{x}}) \leq |R|\theta) \tag{5.2}$$

To define a valid *a contrario* model, we need to find the distribution of Eq 5.2. If we consider the variables $E_{\mathbf{x}}$ to be independent, then $\rho(E_{\mathbf{x}})$ is also independent, which leads to a summation of independent random variables. We can apply the following proposition to obtain the final distribution [28]:

Proposition 3 *Let X and Y be two independent random variables, with density functions f_X and f_Y respectively. Then $Z = X + Y$ has as density function f_Z, the convolution product of f_X and f_Y.*

[1] Recall the basic properties of robust functions: it must be symmetric, positive-definite, with a unique minimum at 0, and in general is chosen to be less increasing than square (L^2 norm).

Now, if we note the distribution function of $\rho(E_{\mathbf{x}})$ as f_ρ, then the distribution function of the probability described before is

$$f_{\sum} = f_\rho * f_\rho * ... * f_\rho = f_\rho^{n*}(x) \tag{5.3}$$

where the number of terms to be convolved is n, the number of valid points at region R. In order to obtain the distribution of Eq 5.2, we have to find the distribution f_ρ first, which is done in Section 5.2.

5.1.1 The Background model

The number of false alarms requires to define a valid *a contrario* or background model. Let's consider that the random variables $D_{\mathbf{x}}$ associated to the disparity values $d(\mathbf{x})$ are uniformly distributed in the interval defined by the minimum and maximum disparity values: $D_{\mathbf{x}} \sim Uni([d_{min}, d_{max}])$. Given a transformation T, if we define a new random variable $E_{\mathbf{x}} = T - D_{\mathbf{x}}$ associated to the error measures, then for each value of T we obtain again a uniform distribution, now defined in the interval $[d_{min} - T, d_{max} - T]$. The problem is that this distribution depends on the value of T, leading to one distribution per transformation, which is impractical. To avoid this problem, from all possible ρ functions, we consider those which are constant outside a given range. The general form of this kind of functions is:

$$\rho(x) = \begin{cases} g(x) & \text{if } |x| \leq \sigma \\ k & \text{otherwise} \end{cases} \tag{5.4}$$

where $g(x)$ is some symmetric, definite positive function with only one zero at zero and k is defined in order to make function ρ continue at $\pm\sigma$. Given the distribution $E_{\mathbf{x}}$, if we evaluate ρ at $E_{\mathbf{x}}$ values, we obtain a new distribution with the following characteristics:

1. For the values of $T \in [d_{min} + \sigma, d_{max} - \sigma]$, the random variable $E_{\mathbf{x}}$ is bounded by $[-\sigma,\sigma]$ and is uniformly distributed in this interval: $E_{\mathbf{x}} \sim [-\sigma, \sigma]$. So, the random variable $\rho(E_{\mathbf{x}})$ is bounded by $[0, k)$.

2. For the values of $T > |\sigma|$, $\rho(E_{\mathbf{x}})$ has a peak in k.

After these considerations, we can now formalize the distribution of $\rho(E_{\mathbf{x}})$.

5.2 Probability Distribution of $\rho(E_{\mathbf{x}})$

In order to obtain the density function of the probability defined in Equation 5.2, we need an expression for the density function of $\rho(E_{\mathbf{x}})$. The following proposition gives the general form of the density function of $\rho(E_{\mathbf{x}})$:

Proposition 4 *Let E be a random variable with density probability function f_e. Given a robust function ρ of the form:*

$$\rho(x) = \begin{cases} g(x) & \textit{if } |x| \leq \sigma \\ k & \textit{otherwise} \end{cases} \tag{5.5}$$

where $g(x)$ is some symmetric, definite positive function with only one zero at zero, the density function of the random variable $\rho(E)$ is

$$f_\rho(\theta) = \tilde{\rho}^{-1}(\theta)'(f_e(-\tilde{\rho}^{-1}(\theta)) + f_e(\tilde{\rho}^{-1}(\theta))) \tag{5.6}$$

for $\theta \in [0, k)$, where $\tilde{\rho}$ is ρ restricted to positive values and

$$f_\rho(\theta) = \int_{-\infty}^{-\sigma} f_e(x)dx + \int_{\sigma}^{\infty} f_e(x)dx \tag{5.7}$$

for $\theta = k$.

Proof. If f_e is the density function of E, the distribution function is:

$$F_e(\theta) = P(E \leq \theta) = \int_{-\infty}^{\theta} f_e(x)dx$$

Then, the distribution function F_ρ of the random variable $\rho(E)$ is:

$$F_\rho(\theta) = P(\rho(E_{\mathbf{x}}) \leq \theta) = \int_S f_e(x)dx \tag{5.8}$$

where the set S is:

$$S = \{x \text{ with } \rho(x) \in [0, \min\{\theta, k\}]\}$$

because function ρ is bounded by $[0, k]$.
Now we define two subsets of S in the following way:

$$S^+ = \{x \geq 0 \text{ with } \rho(x) \in [0, \min\{\theta, k\}]\}$$

$$S^- = \{x < 0 \text{ with } \rho(x) \in [0, \min\{\theta, k\}]\}$$

Recall that function ρ is symmetrical, positive-definite and has a unique minimum at zero. Let's define $\tilde{\rho}$ as:

$$\tilde{\rho}(x) = \begin{cases} \rho(x) & \text{if } |x| \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.9}$$

Let's consider first the case in which $\theta < k$ (which is the same as to consider $|x| \leq \sigma$). We can express the subset S^+ as:

$$S^+ = \{x \in [0, \tilde{\rho}^{-1}(\theta)]\}$$

because $\tilde{\rho}$ is strictly increasing in the interval $[0, \theta]$. Subset S^- can be expressed as:

$$S^- = \{x \in [-\tilde{\rho}^{-1}(\theta), 0]\}$$

because of the symmetry property of function ρ.
As we have $S^+ \cap S^- = \emptyset$ and $S^+ \cup S^- = S$, we can express the integral at Equation 5.8 in two terms:

$$F_\rho(\theta) = \int_{S^-} f_e(x)dx + \int_{S^+} f_e(x)dx \tag{5.10}$$

and finally obtain:

$$F_\rho(\theta) = \int_{-\tilde{\rho}^{-1}(\theta)}^{0} f_e(x)dx + \int_0^{\tilde{\rho}^{-1}(\theta)} f_e(x)dx = \int_{-\tilde{\rho}^{-1}(\theta)}^{\tilde{\rho}^{-1}(\theta)} f_e(x)dx \tag{5.11}$$

The density function f_ρ can be obtained by deriving F_ρ:

$$f_\rho(\theta) = (F_\rho(\theta))' = \left(\int_{-\tilde{\rho}^{-1}(\theta)}^{\tilde{\rho}^{-1}(\theta)} f_e(x)dx \right)'$$

Applying Leibniz integral rule [8]:

$$\frac{\partial}{\partial x} \int_{a(x)}^{b(x)} f(x,t)dt = b(x)'f(x,b(x)) - a(x)'f(x,a(x)) + \int_{a(x)}^{b(x)} \frac{\partial}{\partial x} f(x,t)dt$$

we finally obtain:

$$f_\rho(\theta) = \tilde{\rho}^{-1}(\theta)' f_e(-\tilde{\rho}^{-1}(\theta)) + \tilde{\rho}^{-1}(\theta)' f_e(\tilde{\rho}^{-1}(\theta)) = \tilde{\rho}^{-1}(\theta)'(f_e(-\tilde{\rho}^{-1}(\theta)) + f_e(\tilde{\rho}^{-1}(\theta)))$$

which proves the proposition for $\theta < k$. It remains to obtain the value of density function f_ρ in k. Because F_e is a distribution probability, the summation over the interval $[0, k]$ must be 1. Thus, we

have an accumulation at point k, that correspond to the probability of having an error bigger than σ:

$$\int f_\rho \delta_k = P(|E_x| > \sigma) = \int_{-\infty}^{-\sigma} f_e(x)dx + \int_{\sigma}^{\infty} f_e(x)dx$$

where δ_k is a Dirac Distribution and the integral is a Lebesgue integral [8].

□

As we are working with uniform distributed random variables, the following corollary is of interest:

Corollary 1 *With the same hypothesis and notations of Proposition 5.2, if the random variables E are uniformly distributed over an interval* $[e_{min}, e_{max}]$*, the density function* f_ρ *can be simplified as:*

$$f_\rho(\theta) = \tilde{\rho}^{-1}(\theta)' 2 f_e(\theta) \tag{5.12}$$

for $\theta \in [0, k)$ *and*

$$f_\rho(\theta) = \frac{e_{max} - e_{min} - 2\sigma}{e_{max} - e_{min}} \tag{5.13}$$

for $\theta = k$*.*

Proof. The proof when $0 \leq \theta < k$ is direct as f_e is constant:

$$f_\rho(\theta) = \tilde{\rho}^{-1}(\theta)'(f_e(-\tilde{\rho}^{-1}(\theta)) + f_e(\tilde{\rho}^{-1}(\theta))) = \tilde{\rho}^{-1}(\theta)' 2 f_e(\theta)$$

For $\theta = k$ we have:

$$P(|E_x| > \sigma) = \frac{1}{e_{max} - e_{min}} \left(\int_{e_{min}}^{-\sigma} 1dx + \int_{\sigma}^{e_{max}} 1dx \right) = \frac{e_{max} - e_{min} - 2\sigma}{e_{max} - e_{min}}$$

□

5.2.1 Implementation details

As explained before, the density function $f_{\sum}$ can be obtained by convolving the density function f_ρ derived in the previous section. Corollary 1 states that if the random variables $E_{\mathbf{x}}$ are uniformly distributed in the interval $[-\sigma, \sigma]$, function f_ρ becomes:

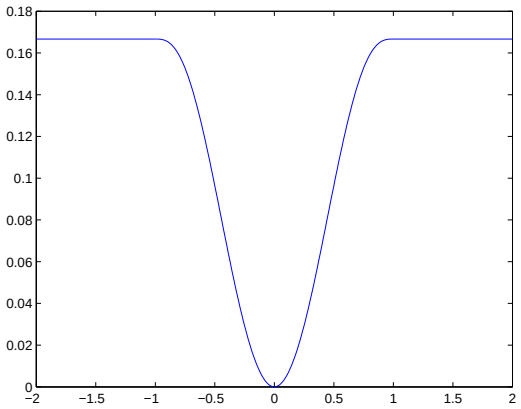
$$f_\rho(\theta) = \tilde{\rho}^{-1}(\theta)' 2 f_{X_e}$$

where $\tilde{\rho}$ is ρ function restricted to positive values. Hence, in order to obtain an analytical expression for f_ρ, $\tilde{\rho}$ must be invertible and its inverse must be differentiable. Besides, if the derivative of its inverse is easily convolable, we can obtain $f_{\sum}$ analytically.

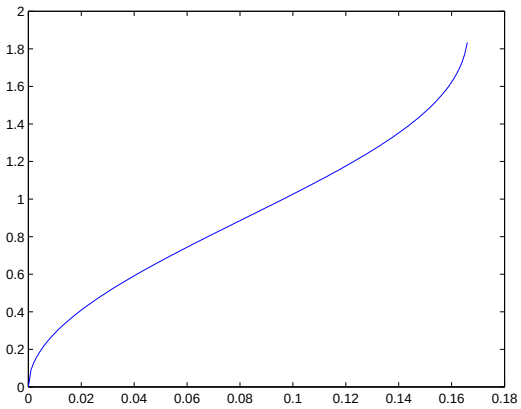
Another approach is to obtain a discrete convolution of f_ρ. If we want to use the discrete convolution we must first sample the interval in a way that guarantees that at the end, we will obtain the values with the required precision. If we consider the random variables $E_{\mathbf{x}}$ to be uniformly distributed in the range $[e_{min}, e_{max}]$, then the random variable $\rho(E_{\mathbf{x}})$ is restricted to the interval $[0, k]$, which leads to the density function we have obtained in the previous section. Given a precision step δ, we sample the interval $[e_{min}, e_{max}]$ with this step and we evaluate function f_ρ at each sample.
One possible function of the type presented in Eq 5.9 is the *Tukey* function:

$$\rho(x) = \begin{cases} \frac{\sigma^2}{6}(1 - (1 - (\frac{x}{\sigma})^2)^3) & \text{if } |x| \leq \sigma \\ \frac{\sigma^2}{6} & \text{otherwise} \end{cases}$$

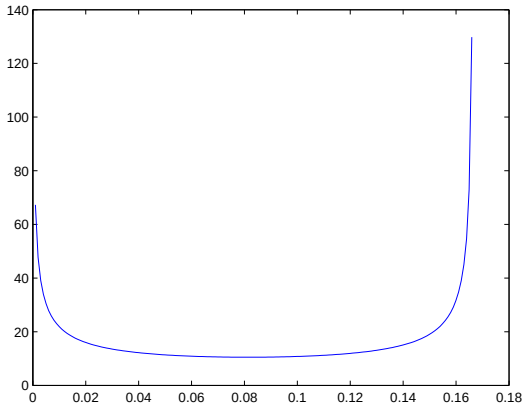
Figure 5.1 shows Tukey function, the intermediate functions to obtain f_ρ and the final f_ρ sampled with a step $\delta = 0.01$.



(a) Tukey Function



(b) Inverse taking into account only positive values



(c) Derivative of the inverse function

Figure 5.1: Robust function and its intermediate ones, used to obtain a discrete convolution of the density function f_ρ

5.3 Reformulation of the number of false alarms

Recall the generic definition of the number of false alarms given at Definition 4:

$$NFA(E) = \mathcal{N} \cdot \mathrm{P}[\mathcal{E} \geq E|H_1]$$

In this case, the observation E corresponds to the error measures θ of Eq 5.2:

$$\theta = \frac{1}{|R|} \sum_{\mathbf{x} \in R} \rho(T(\mathbf{x}) - d(\mathbf{x}))$$

and the random variable $\mathcal{E}$ corresponds to:

$$\mathcal{E} = \frac{1}{|R|} \sum_{\mathbf{x} \in R} \rho(E_{\mathbf{x}})$$

The number of false alarms of a region R and a transformation T can be redefined as follows:

Definition 8 (Number of false alarms) *of a region R and a transformation T is defined as:*

$$NFA(T, R) = N_{tests} P[\frac{1}{|R|} \sum_{\mathbf{x} \in R} \rho(E_i) \geq \theta|H_1] \tag{5.14}$$

where the probability of Eq 5.14 is the complement of the one defined in Eq 5.2:

$$P[\frac{1}{|R|} \sum_{\mathbf{x} \in R} \rho(E_i) \geq \theta] = 1 - P[\frac{1}{|R|} \sum_{\mathbf{x} \in R} \rho(E_i) \leq \theta]$$

which in turns has $f_{\sum}$ as density function.

We obtain the final expression of the number of false alarms:

Definition 9 (Number of false alarms) *of a region R and a transformation T is defined as:*

$$NFA(T, R) = N_{tests}(1 - P[\frac{1}{|R|} \sum_{\mathbf{x} \in R} \rho(E_i) \leq \theta|H_1]) \tag{5.15}$$

where probability P has $f_{\sum}$ as density function.

5.4 An approximation using Hoeffding inequalities

Although we can obtain numerically the distribution explained before, we can also approximate it using one of the Hoeffding inequalities [32], which is much more easy to compute. We will use the following theorem

Theorem 1 (Hoeffding) *Let $Y^1, ..., Y^N$ be independent random variables with $\mu^i = \mathbb{E}[Y^i] \in (0, 1)$ and $P[0 \leq Y^i \leq 1] = 1$, $i = 1, ..., N$. Let $\mu = (\mu_1 + ... + \mu_N)/N$. Then, for $0 < \eta < 1 - \mu$ and $\hat{Y} = (Y^1 + ... + Y^N)/N$,*

$$\mathbb{P}[\hat{Y} - \mu \geq \eta] \leq e^{Nw(\eta,\mu)}$$

where

$$w(\eta, \mu) = (\mu + \eta) \ln(\frac{\mu}{\mu + \eta}) + (1 - \mu - \eta) \ln(\frac{1 - \mu}{1 - \mu - \eta})$$

This theorem is very useful to obtain an upper bound of the number of false alarms in Eq (5.15). The idea to use this theorem to approximate the probability of a continuous set of random variables was first introduced in [46], to the case of motion estimation.
As explained before, the number of false alarms is the expectation of some observed events in the *a contrario* model. Given a transformation T and a region R, we obtain the error of the transformation and the data in R:

$$\theta = \frac{1}{N}\sum_{i=1}^{N}\rho(e_i) = \frac{1}{N}\sum_{i=1}^{N}\rho(T(x_i) - d(x_i)) \tag{5.16}$$

We can use this error to define the event to analyze. If we define the event C_R as "transformation T fits the data in R", and if we normalize the values of θ between 0 and 1, we can model the event C_R as $1-\theta$. The number of false alarms of the fit with error θ is:

$$NFA(R,T) = N_{test}P[C_R \geq 1-\theta]$$

where P the probability of the *a contrario* model.

Going back to the theorem, if we define our Y_i as $1-\rho(E_i)$ where values $\rho(E_i)$ are also normalized between 0 and 1, then the probability becomes

$$\mathbb{P}[\frac{1}{N}\sum_{i=1}^{N}Y_i \geq 1-\theta] = \mathbb{P}[\frac{1}{N}\sum_{i=1}^{N}Y_i - \mu \geq \eta] \leq e^{Nw(\eta,\mu)}$$

Finally, it only remains to obtain μ, which is the mean of $\sum Y_i$ in the *a contrario* model:

$$\mu = \frac{1}{N}\sum Y_i = \frac{1}{N}\sum(1-\rho(E_i)) = 1-\mu_\rho$$

The mean of the random variables $\rho(E_i)$ is

$$\mu_\rho = \int \rho(x) f_e(x) dx \tag{5.17}$$

where f_e is the density function of the random variable E. In the previous section we have assumed that the random variables E_i were uniformly distributed over $[-\sigma,\sigma]$, then $f = \frac{1}{2\sigma}$ and the mean became

$$\mu_\rho = \frac{1}{2\sigma}\int \rho(x)dx$$

Thus, the NFA can be written as:

$$NFA(R,T) = N_{test}e^{Nw(\mu_\rho-\theta,1-\mu_\rho)}$$

where μ_ρ is the mean of the random variables $\rho(E_i)$ defined in Eq 5.17, and θ is the observation (the error) defined in Eq. 5.16.
In order to apply the theorem the hypothesis must been satisfied:

$$\mu = 1-\mu_\rho \in (0,1)$$

If the function ρ is constant outside a defined range as discussed before, then we can normalize $\rho(e_i)$ to be always between 0 and 1, and then μ_ρ is also normalized in $[0,1]$. This means that $\mu \in (0,1)$.
The other hypothesis is

$$0 < \eta < 1-\mu$$

$$\Rightarrow 0 < \theta - \mu < 1-\mu$$

$$\Rightarrow 0 < 1-\theta_e-(1-\mu_\rho) < 1-(1-\mu_\rho)$$

The final condition is:

$$\Rightarrow 0 < \theta_e < \mu_\rho$$

Although we would obtain this value analytically, we can also compute it using a discretization method. Suppose we define a step τ for the discretization. We discretize the range $[-\sigma, \sigma]$ in M bins $-\sigma + \tau i$ with $i \in 0..M$. Then, the mean can be obtained as:

$$\mu_\rho = \frac{1}{M} \sum_{i=0}^{M} \rho(e_i)$$

5.4.1 A reformulation of the merging criterion

Once we have defined this new approach to compute the Number of False Alarms of a transformation and a region, we can analyze again the merging criterion defined in Section 4.3. Recall Eq. 4.6:

$$NFA(R_i \cup R_j, T_{ij}) \leq NFA((R_i, T_i), (R_j, T_j))$$

that we have shown that can be written it as:

$$P(R_i \cup R_j, T_{ij}) \leq N_{Trans} * P((R_i, T_i), (R_j, T_j))$$

The first term is computed directly from the definition presented before. Now, for the second term, as both regions have the same *a contrario* model, we can apply again the Theorem 1, taking into account the error at each observation. That is, ours Y_i would be the errors computed with T_i for the points in R_i, and the errors with T_j for the points in R_j. The number of points N would be $n_i + n_j$, and μ would be the same because of the *a contrario* model. We finally have:

$$e^{(n_i+n_j)w(\mu_e - \theta_{ij}, 1-\mu_e)} \leq N_{trans} e^{(n_i+n_j)w(\mu_e - (\theta_i+\theta_j), 1-\mu_e)}$$

where θ_{ij} is the error computed with T_{ij} in the points of region R_{ij} and θ_i, θ_j are the errors computed in T_i for the points in R_i and T_j for the points in R_j respectively. This is indeed, the same idea that we have already applied to the case of the probabilities computed using the binomial tail at Section 4.3.

Chapter 6

A discussion on the initial segmentation

The region-merging algorithm introduced before has two inputs: a non-dense disparity map and an initial segmentation of the reference image. Indeed, a segmentation based on the reference image is a suitable approximation of the objects present in the scene, because of the lambertian hypothesis [1]. There exist several methods to obtain the initial segmentation; in fact, the segmentation of an image in connected regions is one of the most active topics in computer vision. See [42, 47] for a complete review.

In this chapter we analyze which is the relation between the initial segmentation and the resulting interpolation of the disparity map. We analyze two variants of the classical Mumford and Shah functional[43]. The first one is an implementation using a piecewise constant function approximation [38]. The second one is similar to the first one but the boundaries are restricted to be level lines of the image [6, 46]. The third segmentation we have analyzed is a different approach based on geometrical characteristics of the image. It is based on the detection of the segments present in the image and the polygons defined by them. This segmentation is more suitable for urban scenes since most of the objects presented in such images can be represented by a set of polygonal objects.

6.1 Segmentations based on Mumford-Shah functional

In this section we review two different segmentation approaches, both based on a simplified version of the Mumford and Shah functional [43]:

$$E(B, \tilde{u}) = \int_{\Omega \setminus B} (\tilde{u} - u)^2 + \lambda l(B) \tag{6.1}$$

where B is the set of boundaries between regions, $\Omega \subset \mathbb{R}^2$ the image domain, $\tilde{u}$ the piecewise constant function and $u : \Omega \to \mathbb{R}$ the image to approximate.
The first approach is based on a region growing method presented in [38], that enables us to define the number of regions we want to have in the final segmentation. Given the original reference image, we segment it into intensity constant regions.

The second approach [6, 46] is based on the same functional, but the set of boundaries B is restricted to be the level lines of the image. Recall that from mathematical morphology, an image can be described by its upper or lower level sets [13, 54], where the level lines are the boundaries between these level sets. As the boundaries of the objects of an image are generally composed by pieces of level lines, it is a good idea to restrict the boundaries to this set.

6.2 A segmentation based on polygons

Let's now make use of an important characteristic of images taken from urban scenes: they are generally composed by geometrical structures well delimited such as roofs, streets, etc. These geometrical structures can be described as a set of polygons, each one formed by tree or more segments (a triangle, a rectangle, etc). Thus, a segmentation based on polygons seems to be a suitable segmentation of an urban scene. The segmentation algorithm based on polygons can be summarized as follows:

1. Find all the meaningful segments of the image.
2. Keep only the maximal meaningful segments.
3. Given a minimum and a maximum number of sides, construct all possible polygons using the segments obtained before and label them with a different label for each polygon.
4. Find all the intersections of the set of polygons obtained before and label them with a different label for each intersection.
5. For each pixel of the reference image, find the smallest polygon that contains it and assign its label to the pixel.

[1] the surface luminance is the same regardless of the angle of view

6.2.1 Meaningful segments

The detection of the meaningful segments of an image was introduced in [24, 22] and improved lately in [3, 57] and is based on the computational gestalt theory explained at chapter 4.

Definition 10 (Number of false alarms of a segment) *Let A be a segment of length l with k points having their direction aligned with the direction of A. We define the number of false alarms of A as*

$$NFA(A) = NFA(l,k) = N^4\mathbb{P}[S_l \geq k] = N^4 \sum_{i=k}^{l} C_k^l p^k (1-p)^{l-k} \tag{6.2}$$

where p is the probability of having a direction in the background model and N is the size of the image (considered square).

As we are working with discrete values, we can say that two points have the same direction if their difference is less than a given threshold n (for instance 16). If we define the *a contrario* model to be uniformly distributed, the probability of two points to have the same direction n is $p = 1/n$.

The ϵ-meaningful segment is defined as:

Definition 11 (ϵ-meaningful segment) . *Given a segment A, we say that the segment is ϵ-meaningful if*

$$NFA(A) \leq \epsilon \tag{6.3}$$

6.2.2 Maximal meaningful segments

It is clear from Definition 11 that longer or shorter segments can still be ϵ-meaningful. However, our perception process keep only the most significant of all these meaningful segments. This leads to the following definition [25]:

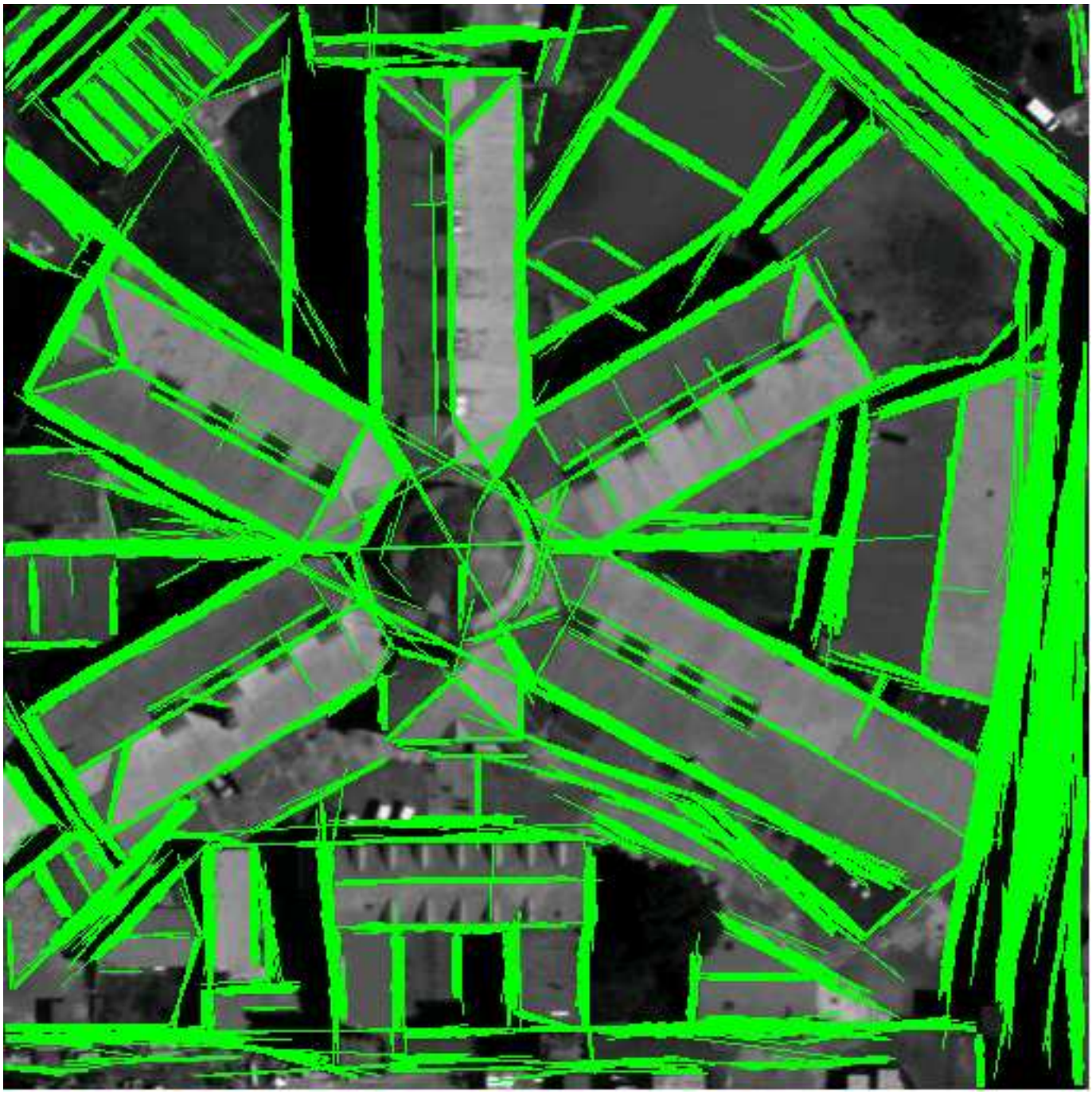
Definition 12 (Maximal meaningful segment) *We say that a segment A is* maximal meaningful *if it is meaningful and:*

1. *It does not contain any more meaningful segments.*
2. *It is not contained in a more meaningful segment.*

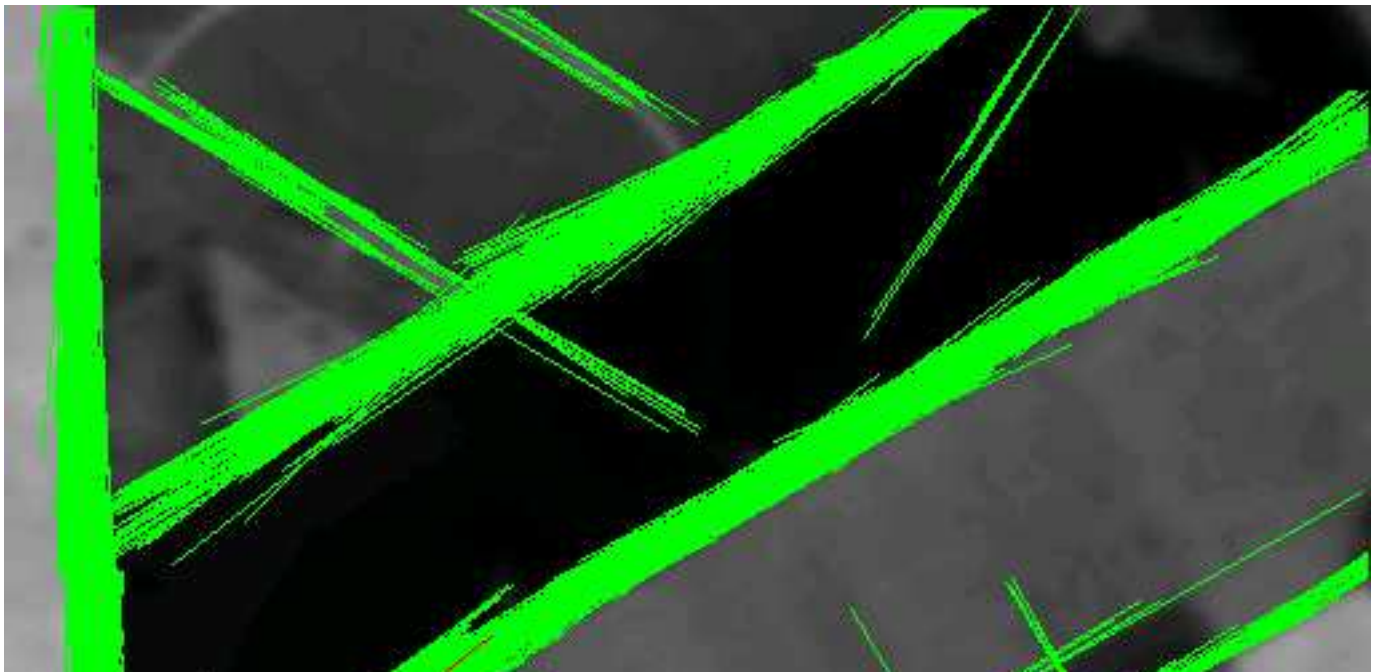
Although this definition removes the ambiguity when we have segments completely contained in each other, we still may have many meaningful segments for the same border. Indeed, correctly sampled images are at least slightly blurred. In that case, the borders present in the images are bigger than one pixel, which leads to the detection of several meaningful segments near them as we can see at the top row of Figure 6.1.
In order to obtain the best segment among all the maximal meaningful ones, we use an exclusion principle similar to the one presented in Section 4.5, that states that a point must belong only to one segment [25] and can be summarized as follows: we assign each pixel x to the segment A if the NFA(A) is the lowest among all the segments that also have pixel x (we consider that a pixel belongs to a segment if the distance between the point and the segment is less than some given radius r).
Now, we compute again the NFA of each segment A but now we define as valid points only the assigned points. This leads to an increase in the NFA, but if it is still less than ϵ, we consider the segment to be a *maximal EP meaningful segment*. The bottom row of Figure 6.1 shows the set of maximal meaningful segments after applying the exclusion principle explained so far.

6.2.3 Building the polygons

In [56], a detection of meaningful polygons is analyzed. It starts by constructing all the possible polygons from the set of maximal meaningful segments, using a given range in the number of sides (for instance 3 to 6). To construct these polygons, we find all the intersections between the set of



(a) All 1-meaningful segments



Input		100 %
SEGM	Error	Valid Regions
MS	0.266145	95.83 %
LI	0.254563	79.88 %
Polygonal	0.217938	88.54 %

Table 6.1: Error Table using different initial segmentations, obtained by different techniques, computed on a initial disparity map with precision 0.25. They were computed only at the regions validated by the region-merging algorithm. The column Valid Regions shows the percentage of valid points in the image.

segments. The segments are in fact enlarged in both directions by a factor given as parameter to the algorithm. This is done to find the intersection of segments that are close enough to form a polygon but do not intersect, as we can see at the roof details presented in Figure 6.1(d).
In [56], after obtaining the set of all possible polygons, a criterion based on the definition of an *a contrario* model is performed to obtain only the meaningful ones. For our purpose, which is to obtain an over segmentation of the image, the resulting seg of polygons is not quite adequate. Indeed, if the objects are not well contrasted, they are not detected as meaningful polygons. Thus, we follow another approach which is to use all the original polygons to make a partition of the image: For each one of the polygons, we find all the intersections between them and we define a new polygon for each of these intersections. Then, for each pixel in the image, we look for the smallest polygon that contains it and we assign its label to the pixel. We proceed in the same way for all the pixels in the image, obtaining at the end an image of labels. Following this process we finally end with a partition of the image into disjoint regions, composed by the polygons and all their intersections, as it is shown in Figure 6.2(d).

6.3 Evaluation of the different segmentations

In this section we analyze the influence of the initial segmentation in the region-merging algorithm. We have tested the region-merging algorithm using the segmentation techniques presented so far. We present the notation that we have used for each technique in the figures and tables of this section:

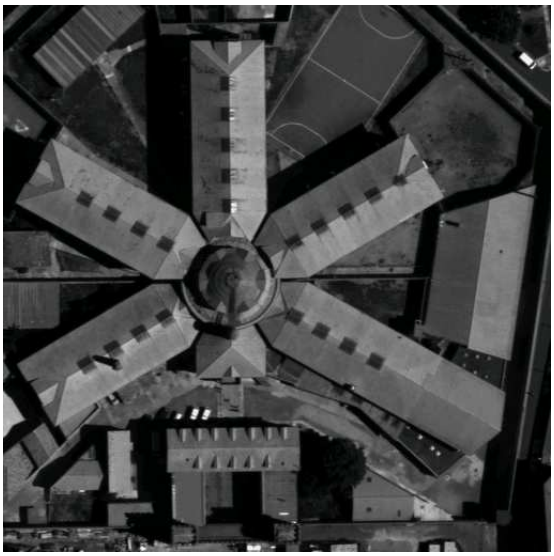
1. **MS** is the segmentation based on the Mumford and Shah functional with piecewise constant approximation presented in Section 6.1:

2. **LI** is the segmentation based on the same same Mumford and Shah functional but where the set of boundaries B is restricted to the level sets of the image, also presented in Section 6.1.

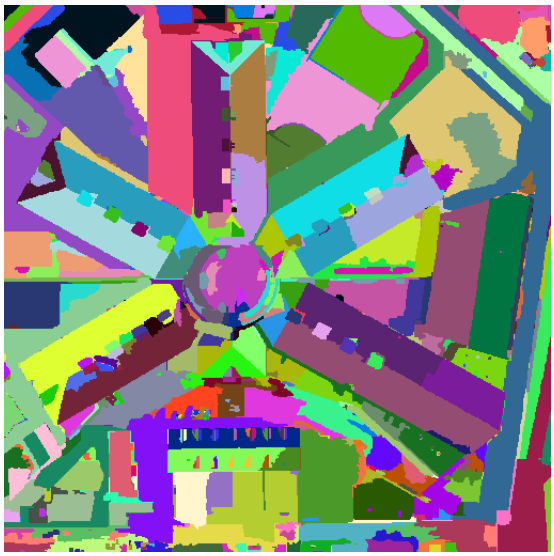
3. **PL** is the segmentation obtained from the polygons build from meaningful segments presented in Section 6.2.

Figure 6.2 shows the original reference image and a segmentation obtained from each of the techniques explained before. In Table 6.1 we list the error measure computed against the ground truth. As expected, the best results are obtained using the polygonal segmentation. This means that the polygonal segmentation leads to a better interpolation in the validated regions. However, the **LI** segmentation leads to a better result in terms of denoising the original valid points.

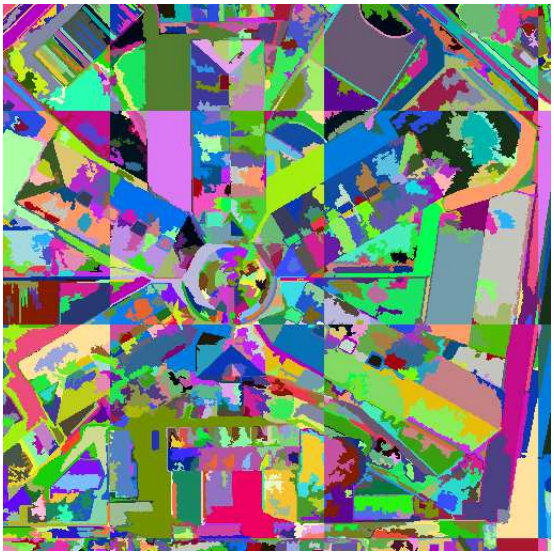
We also analyze the performance of the algorithm in relation to the number of initial regions. Table 6.2 shows the error measures using **MS** segmentation technique. Although the segmentation using 2500 regions gives better results than the one with 1000, the performance drop down for a segmentation of 5000 regions. The reason for this degradation is that in order to estimate the disparity map, we use only valid disparity values. If we oversegment the image, many of the small regions would not have a minimum number of valid points to estimate the transformation. In our implementation, we just merge these regions by its gray level using a simplified approach where we merge the two regions whith the most similar gray level. In order to improve the results, it is necessary to incorporate the



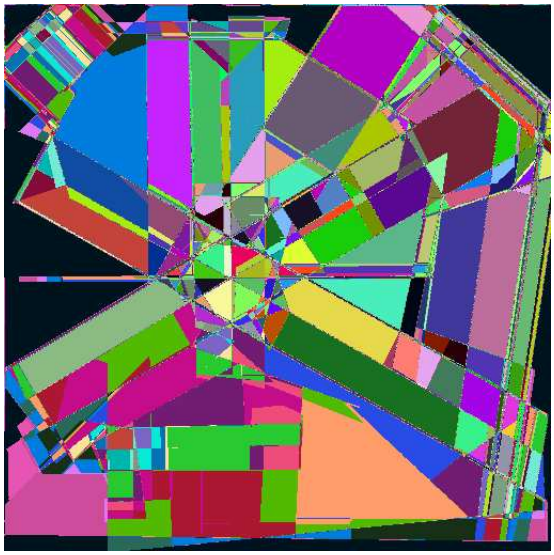
(a) Original reference image



(b) Mumford & Shah Segmentation with 1000 regions



(c) L. Igual segmentation with 3560 regions



(d) Polygonal segmentation

Figure 6.2: Initial segmentations

Input		100%
Nb Regions	Error	Valid Regions
1000	0.261089	92.65 %
2500	0.231525	87.14 %
5000	0.260127	84.68 %

Table 6.2: Region-Merging performance with different number of regions in the initial segmentations, using always the MS piecewise segmentation approach to generate them.

gray level information to the merging algorithm, defining a valid merging criterion for this case. In this way, we compute the final disparity map incorporating all the information of the scene that we have: the gray level reference image, and the disparity values.

Chapter 7

Experimental Results

FALTA:
1. RESULTADOS DE RAME SIMULADOS

In this chapter we present the different experiments we have done to analyze the performance of the region-merging algorithm. We start by introducing at Section 7.1 the data sets and the error measures used in our experiments. Section 7.2 summarizes the most important results obtained with the method presented in this work. We show a comparison, both quantitative and qualitative, between this method and the other existing sub-pixel ones presented in Section 3.3. The other sections are devoted to analyze different aspects of the region merging algorithm: Section 7.3 presents an analysis of the algorithm in relation to the merging criterion (the discrete or the continuous one). Section 7.4 shows the results obtained using disparity maps obtained from different methods.

7.1 Datasets and error measures

7.1.1 Datasets

We have done all the experiments using two data sets. The first one is a real pair of aerial images of the same scene with a b/h factor (baseline / altitude) of 0.045 and is shown at Figure 7.1. From now on, we will refer to this dataset as **REAL**. We also have the ground truth for this pair of images (Figure 7.1(c)). In order to analyze the performance of the methods presented here, this set has a drawback: the images were taken with a difference of more than 20 minutes. With this delay, some objects in the scene have moved between one image and the other (the shadows, cars, etc). Recall that in this work we are interested in images taken almost simultaneously, so there is no interest in analyzing the results of the methods in the shadows. To avoid this problem, we have manually built a mask of valid regions, leaving the big shadows that appear in the images out of the mask. This mask is shown in Figure 7.1(d).
Even if we use the mask that leaves out the big shadows, small shadows can still be taken into account because the mask does not cover them. To avoid this problem, we have simulated a second dataset (called **SIM**) from the original images and the ground truth information using the method described in the Appendix B.

7.1.2 Error measures

As we have the ground truth for our image set, we can perform a quantitative analysis using an error measure based on a modified L_2 distance, that considers the possibility of a small geometric deformation of the ground truth:

$$d(e, \varepsilon, M) = \sum_{x \in M} \frac{1}{|M|} \min_{|x'-x| \leq r} (e(x')a + b - \varepsilon(x))^2 \tag{7.1}$$

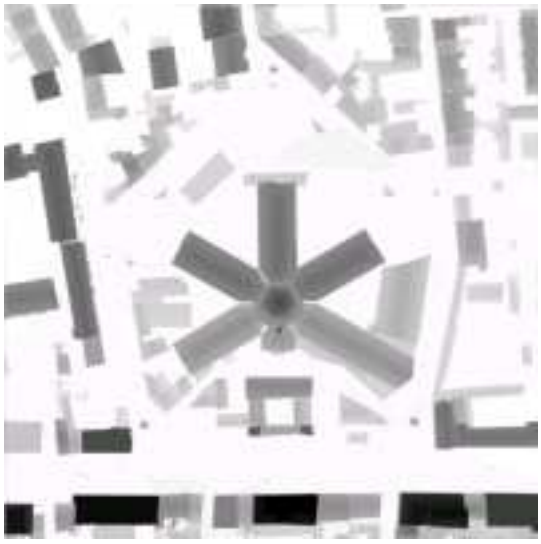
The distance takes the valid points map M and computes the optimal scaling parameters a and b to minimize the error, additionally it considers the minimum error inside a neighborhood of $r = 1$ to avoid small registration errors principally over the borders.

Additionally, as one of the outputs of the MARC algorithm is a mask of valid points, the comparison with the ground truth is performed over the masks of valid points. This analysis gives an idea of the improvement in the original disparity measures, since no interpolated data is being taken into account. Finally, in the algorithms with a validation method, the error is also computed over the valid regions and compared with the other ones. We summarize the notation for each of the measures presented in the tables and figures of this chapter:

1. **VALID** - Is the set of valid points returned by the method that generates the disparity map to be approximated.
2. **SHADOW** - Is the mask where we keep the big shadows out.
3. **MERGE** - Is the set of valid regions returned by the region-merging algorithm.

(a) Reference Image

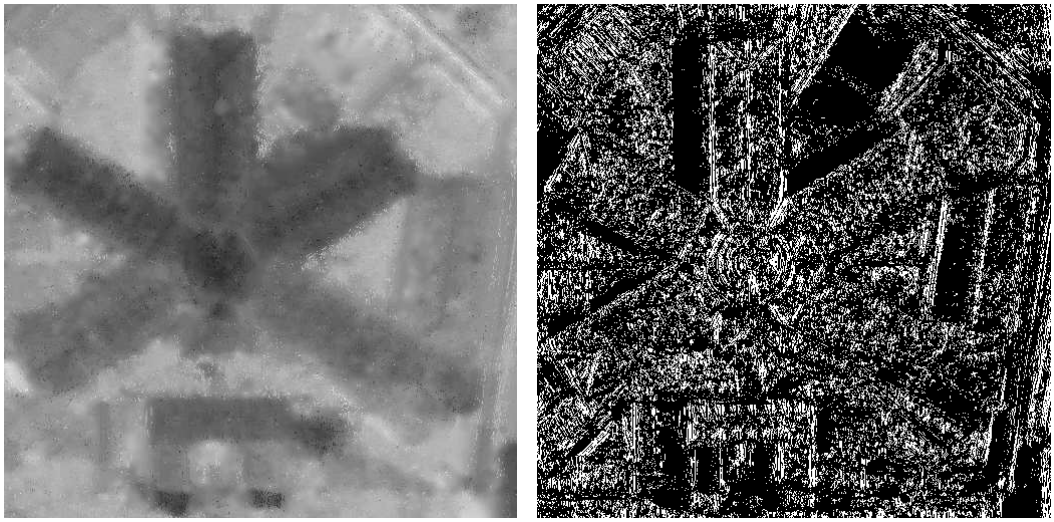
(b) Secondary Image



(c) Ground Truth

(d) Mask of valid regions (the white ones).

Figure 7.1: **REAL**: The set of real images, taken with a baseline/altitude relation of 0.045, the Ground truth and the mask of valid points used to compute the error.



(a) Not dense disparity map obtained by MARC (b) Mask of valid points returned by MARC (the white ones)

Figure 7.2: Disparity map obtained by MARC and the corresponding mask of valid points. A crop of 512x512 from the original 1000x1000

7.2 Summary of Results

In this section, we show the results obtained using the different methods presented before. We adopt the following nomenclature to simplify the presentation of the results:

v0: Original MARC version: a multi-resolution, multi-scale cross correlation with barycentric correction (Section 3.3.1).

v21: Minimal surface and global data fit: $S_3(\varepsilon) + \lambda D_1(\varepsilon)$ (Section 3.3.1).

RAME: Region-based Affine Motion Estimation (Section 3.3.2)

MERGE-NFA: Region-Merging algorithm with merging criterion based on NFA (Chapter 4).

COMB: A combination between the disparity map computed with **MERGE-NFA** and **v21**, where we keep the disparity of the merging process in the validated regions, and we use the disparity of **v21** in the others.

Precision: 0.25 REAL	VALID ∩ SHADOW ∩ MERGE 5.54 %	VALID ∩ SHADOW 5.77 %	SHADOW 23.95 %	MERGE 95.54 %
Input	0.226030	0.226767	0.238051	0.360394
MARC v1	0.170697	0.171363	0.215615	0.342045
MARC v21	0.165686	0.165396	0.175263	0.290857
RAME	**0.140566**	**0.142189**	**0.165362**	0.289553
MERGE-NFA	0.153548	0.176193	0.188532	**0.269563**
COMB	0.153548	0.153653	0.169192	0.269572

Table 7.1: Comparative results using as the initial disparity map the one generated by MARC V1 before interpolation. The map was generated with the real set of images and the initial segmentation based on Mumford and Shah (Sec. 6.1). The resulting map is shown in Figure 7.5.

Table 7.1 shows the results of the different methods. Figure 7.2 shows the initial disparity map generated by MARC and its corresponding mask of valid points[1]. The error measures are listed in

[1]Recall that MARC generates a non dense disparity map

Precision: 0.25 REAL	VALID ∩ SHADOW ∩ MERGE 14.48 %	VALID ∩ SHADOW 18.48 %	SHADOW 77.01 %	MERGE 68.93 %
Input		0.210837	0.272751	
MARC v1	0.148391	0.186594	0.246648	0.210860
MARC v21	0.125083	0.164830	0.182789	0.136278
RAME	0.109499	**0.139619**	**0.160053**	0.122410
MERGE-NFA	**0.105658**	0.248457	0.268886	**0.120245**
COMB	0.105658	0.150634	0.170584	0.120245

Table 7.2: The same experiment of Table 7.1 but using as initial input the crop of 512x512 shown at Figure 7.2. Note that in the case of the **RAME** algorithm, the result is computed directly from the pair of images

Precision: 0.25 SIM	ALL 100.00 %	VALID 41.08 %	VALID ∩ MERGE 40.76 %	MERGE 98.74 %
Input	0.119538	0.042609	0.042021	0.114802
MARC v1	0.076431	**0.042708**	**0.041841**	0.074795
MARC v21	0.090244	0.056693	0.054389	0.084797
RAME	0.060982	0.057251	0.055831	0.058265
MERGE-NFA	**0.074865**	0.056597	0.052916	**0.068179**
COMB	**0.069782**	0.053645	0.052916	0.068179

Table 7.3: The same comparative results as Table 7.2 but using the pair of simulated images.

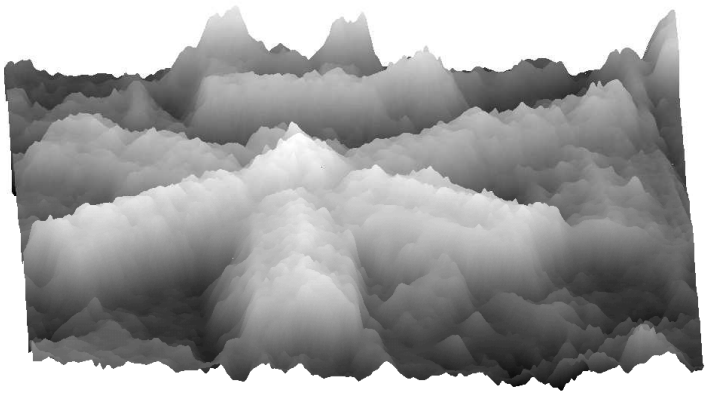
the Input row of Table 7.1. This input is used in all the methods presented (MARC V1, MARC v21, MERGE-NFA and COMB) except for the RAME method, which computes the disparity map directly from the pair of original images.

Note that in order to analyze the performance of the region-merging algorithm, the relevant error measures are those computed over the regions validated by the merging process. In Figure 7.5(b) we can see these valid regions, which are about 93% of the total image. This is because, as we have said before, that the merging algorithm assumes an affine transformation at each region. Although this is a good approximation for plane regions, it is not a valid model for other kind of objects. Indeed, the validation process enable us to determine which regions can be modeled by affine transformations and which regions cannot. For the valid regions, the error drops down from 0.29 to 0.26. In fact, if we consider the error of the disparity map from which the region merging is applied, this error drops from 0.36 to 0.26. The best error measure is obtained by a combination of both methods: the affine region model obtained from the merging process to the valid regions, and the Marc v21 for those which are not valid. This composition is shown in Figure 7.5(d).

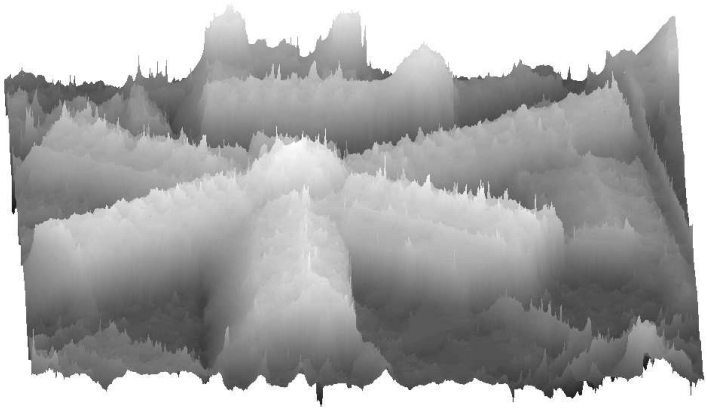
Recall that if we compare the results of Table 7.1, the RAME method gives the best results for the valid points, shadow and valid regions. Nevertheless, when looking in the valid regions, the merging process gives the best results. One thing we can analyze is the results of the merging process when we use as the input disparity maps generated from different methods. This is done in Section 7.4.

Although we have a good idea of the performance of the algorithms when seeing the error Table 7.1, we can also have a qualitative idea of the performance when looking at the reconstructed 3D scene, as shown in Figures 7.3 and 7.4. We can see that the combination between the region-merging and the variational approach as we explained it before, gives a better result.

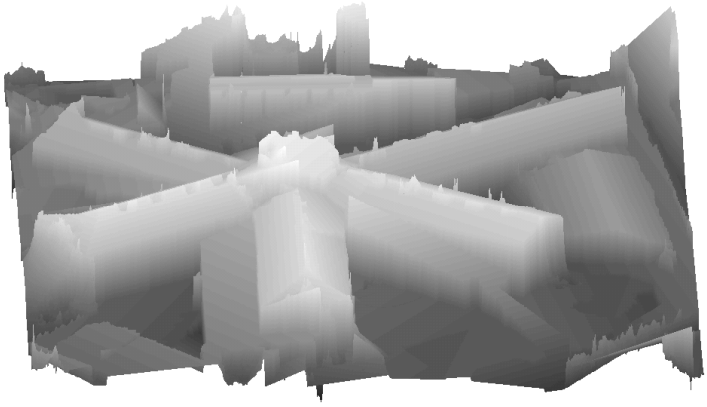
Going back to Table 7.1, the results obtained by the **RAME** method are better than the ones obtained by MARC and its variants (V1 and V21). This is an expected result because this method is contrast invariant, and we know that there are differences in contrast between both images due to the shadows movement. Table 7.3 presents the same experiment but using the pair of simulated images. In this case, the best results are obtained by **MARC** which is also expected since in this



(a) 3D scene generated by the disparity map obtained from MARC V1

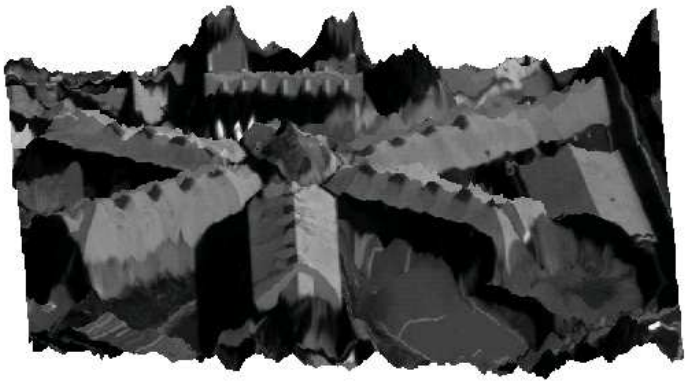


(b) 3D scene generated by the disparity obtained from MARC V21

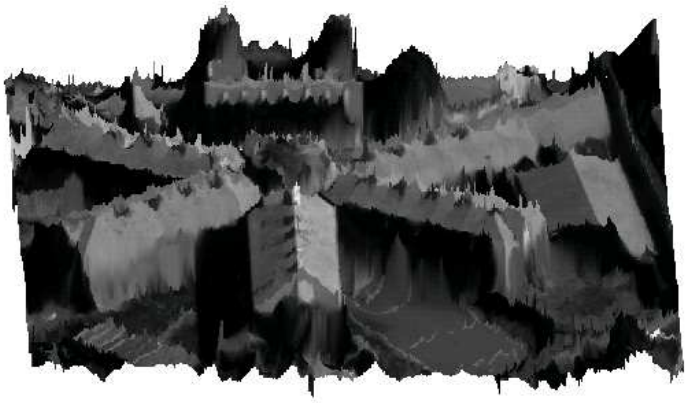


(c) 3D scene generated by the disparity obtained from region-merging

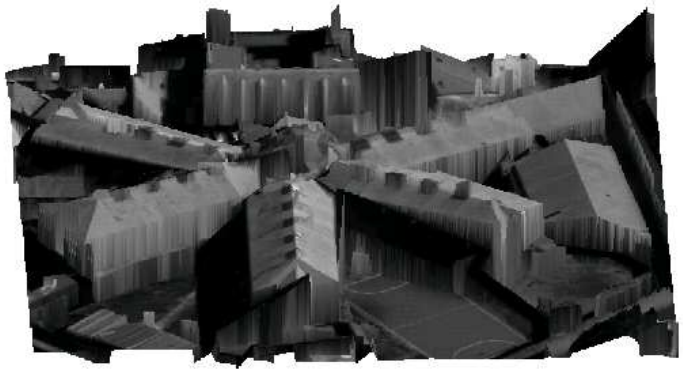
Figure 7.3: A spatial visualization of different disparity maps.



(a) 3D scene generated by the disparity map obtained from MARC V1

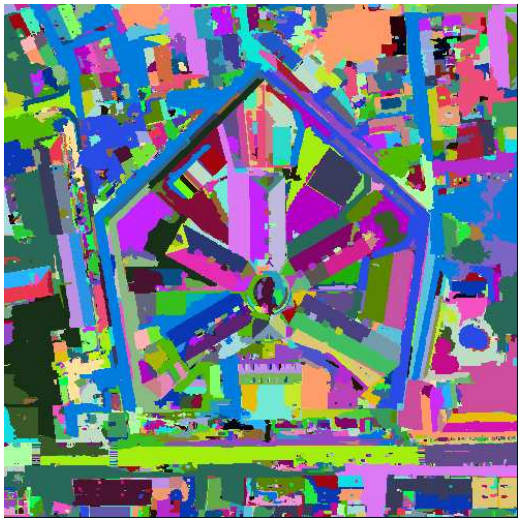


(b) 3D scene generated by the disparity obtained from MARC V21

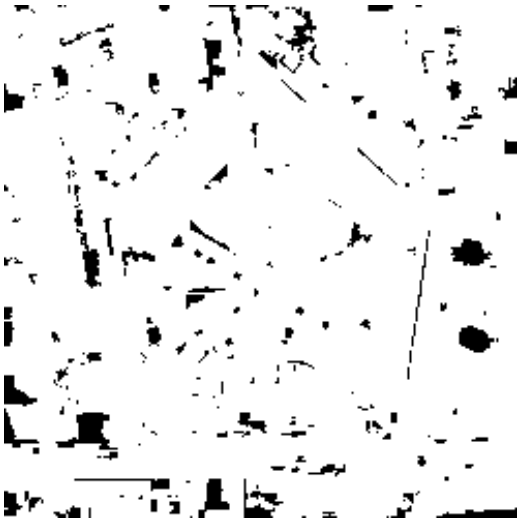


(c) 3D scene generated by the disparity obtained from region-merging

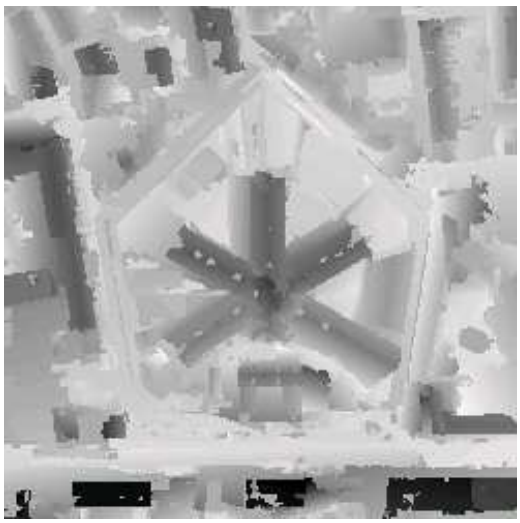
Figure 7.4: The same disparity maps presented in Figure 7.3 using the original gray level image as the texture.



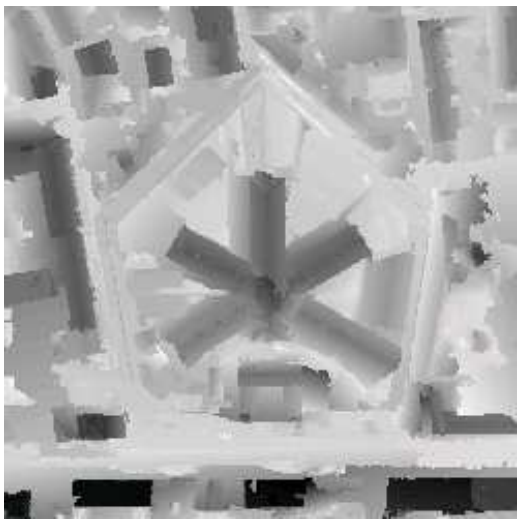
(a) Segmentation based on a piecewise Mumford and Shaw functional. 4500 regions.



(b) Regions validated by the merging procedure.



(c) Result from the merging process



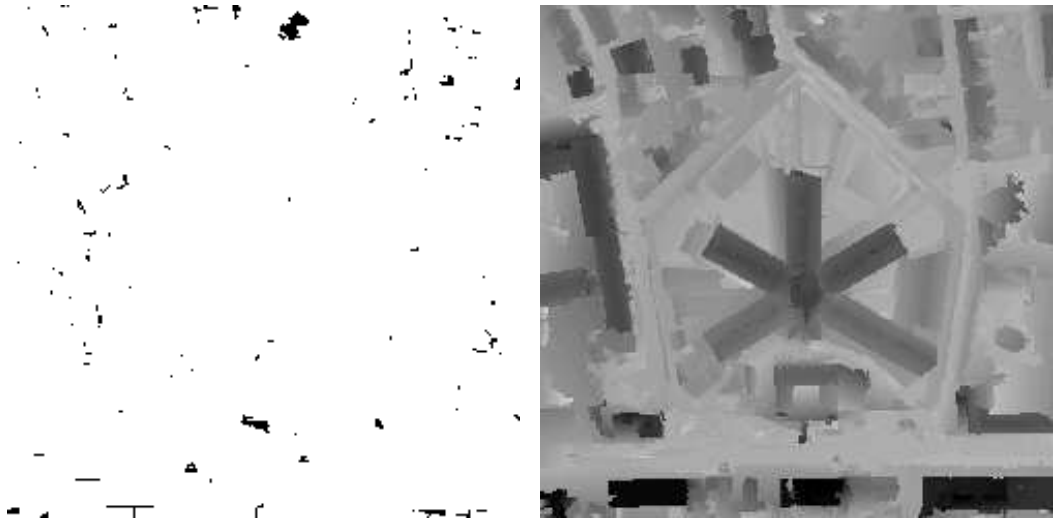
(d) A composition between the merging result and marc v21

Figure 7.5: Results of the merging process. The disparity map in (d) is a combination between the disparity map computed by the merging process and the one obtained by Marc v21. We keep the information from the merging process in the regions validated by this process. In the not valid regions, we use the information from Marc v21

case, contrast invariance is not necessary, and **MARC** uses more information (the intensity values) than the **RAME**.

7.3 Comparison between different merging criteria

In this section we analyze the performance of the algorithm considering the two different merging criteria presented before: the discrete and the continuous one. The last one is at present only implemented in its Hoeffding approximation (Sec 5.4). For this test, we take into account a crop of the original images a factor of two (that is, images of 512 by 512 pixels). Obviously, both methods are valid only in the valid regions. In order to compare the results, we must analyze them with the same mask. Table 7.5 shows the results of the different merging criteria. MERGE1 corresponds to the discrete merging criterion whereas MERGE2 to the continuous one. As expected, the continuous implementation gives better results, since it uses more information to obtain the NFA of each region.



(a) Regions validated by the merging procedure. (b) Result from the merging process

Figure 7.6: Results of the merging process using as input the disparity values obtained by RAME.

As we are doing an approximation with affine transformations and a region-merging procedure, we also analyze whether the improvement obtained in the final result depends on the region-merging approach or it is due to the affine approximation. For doing so, we have computed the errors with and without merge, in both cases doing an affine approximation at each region. The results are listed at Table 7.6. Although the improvement is not very significative (1 %), we obtain better results when we apply the merging criterion and, which is more important, the final scene is well segmented in coherent regions.

Precision: 0.25 REAL	ALL 100 %	VALID ∩ SHADOW 18.48 %	SHADOW 77.01 %	MERGE1 92.65 %	MERGE2 59.07 %
Input	0.315763	0.210837	0.272751	0.310297	0.208683
MARC V21	0.263286	0.164830	0.182789	0.259091	0.119339
MERGE1	0.296696	0.216466	0.238449	0.261089	0.135424
MERGE2	0.327764	0.288394	0.300664	0.320845	0.099923
COMB1	0.264584	0.162011	0.187463	0.261089	0.108643
COMB2	0.257716	0.150352	0.170784	0.252964	0.099923

Table 7.4: Comparative results using different merging criteria. The initial disparity map is the one generated by MARC V1 before interpolation. The map was generated with the real set of images. The resulting map is shown in Figure 7.5.

7.4 Comparison between different disparity maps

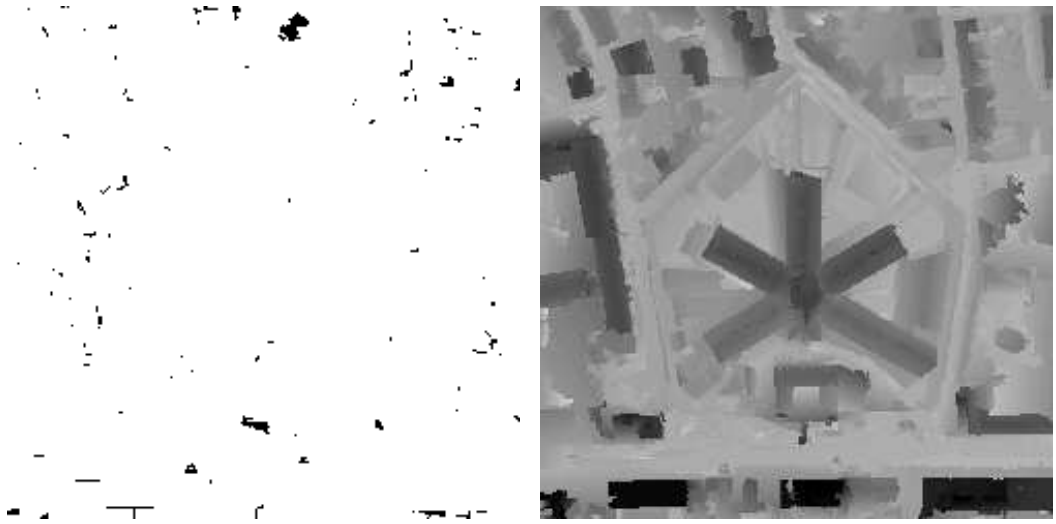
Finally, we are interested in analyze whether the region-merging algorithm can be used as way to improve the results obtained by any of the methods presented so far. That is, given the best results of each method, we try to determine if the region-merging process improves these results. The results of these experiments are shown at Table 7.7

Precision: 0.25 REAL	ALL 100 %	VALID ∩ SHADOW 18.48 %	SHADOW 77.01 %	MERGE1 92.65 %	MERGE2 59.07 %
Input	0.315763	0.210837	0.272751	0.310297	0.208683
MARC V21	0.263286	0.164830	0.182789	0.259091	0.119339
MERGE1	0.296696	0.216466	0.238449	0.261089	0.135424
MERGE2	0.327764	0.288394	0.300664	0.320845	0.099923
COMB1	0.264584	0.162011	0.187463	0.261089	0.108643
COMB2	0.257716	0.150352	0.170784	0.252964	0.099923

Table 7.5: Comparative results using different merging criterion. The initial disparity map is the one generated by MARC V1 before interpolation. The map was generated with the real set of images. The resulting map is shown in Figure 7.5.

Precision: 0.25 REAL	ALL 100 %	VALID ∩ SHADOW 18.48 %	SHADOW 77.01 %	MERGE 59.07 %
Input	0.315763	0.210837	0.272751	0.208683
MARC V21	0.263286	0.164830	0.182789	0.119339
MERGE-NFA2	0.327764	0.288394	0.300664	0.099923
COMB2	0.257716	0.150352	0.170784	0.099923
NOT MERGE	0.261966	0.159683	0.194831	0.111611

Table 7.6: Comparative results using merging and not merging



(a) Regions validated by the merging procedure. (b) Result from the merging process

Figure 7.7: Results of the merging process using as input the disparity values obtained by RAME.

REAL	VAL ∩ SHADOW ∩ MERGE	VAL ∩ SHADOW 18.48 %	SHADOW 77.01 %	MERGE
MARCV1 Input	0.182890 (17.71 %)	0.186594	0.246648	0.297875 (95.05 %)
Output	0.178836	0.211656	0.238472	0.283137
MARCV21 Input	0.164778 (17.93 %)	0.165879	0.177370	0.238776 (95.97 %)
Output	0.155009	0.174768	0.193007	0.228911
RAME Input	0.139346 (18.40 %)	0.139619	0.160053	0.209444 (99.61 %)
Output	0.142463	0.142463	0.163128	0.203138

Table 7.7: Comparative results using different initial disparity maps. The initial segmentation has 1000 regions.

Chapter 8

Conclusions and future work

In this work we have focused on the computation of highly accurate subpixel disparity map. This enables us to take the pictures almost simultaneously, removing most of the problems presented with occlusions and object displacements when the pictures are taken with a big baseline. The experiments have shown that this can be achieved in practice and that such a configuration of small baseline and subpixel accuracy gives good results in the case of urban scenes.

The introduction of a region-merging piecewise affine model, improves the results both quantitative and qualitative when the images are taken from urban scenes. Due to the validation method presented in Chapter 4, we can use an affine model at those regions where this affine model is correct (roofs, streets, ground, etc.), and keep the original disparity values for those regions where the affine model is not suitable (trees, irregular ground, etc.).

We also presented in Chapter 5, a general way to define the number of false alarms of an event using a continuous formulation. We have applied this new formulation to the case of the fitting of a transformation, and we have shown in the experiments that this redefinition leads to a better definition of the meaningfulness of the event. More applications of this formulation can be thought for existing definitions of number of false alarms (segments, boundaries, etc.) and for the definition of new ones.

In conclusion, the present work can be seen as a first step towards a semantic description of the scene. In this case, a piecewise affine model was analyzed, but the general method and the validation approach presented so far enables us to consider other models. Given a model, we can test wether it is suitable for the disparity values we want to approximate validating the fit using a well defined *a contrario* model. This can be done with several models, keeping the one that best explains the given data.

8.1 Future works

During the execution of this project, some practical issues and variations of the general method arises. As we could not deal with all of them during this work, we include them here to future works (although some of them have already been investigated):

Merging criterion for gray scale images. We can consider this as a sub-problem of the original affine region merging algorithm since piecewise constant functions are a subset of the affine transformations. One possible application is to segment gray level images but also approximate piecewise constant disparity maps. We can see in Figures 8.1 and 8.2 some examples of this application. Figures 8.1(c) and 8.2(c) show the result of the piecewise constant region-merging algorithm. In Figures 8.1(d) and 8.2(d) we can see the segmentation obtained by M & S algorithm with the same number of regions. Although the results between both algorithms are quite similars, the region-merging approach present a major drawback, whic is a lack of regularizaton on the borders of the different objects present on the scene. This is an expected problem since no border information is being used. This problem is not present on the affine region-merging approach because this information is obtained from the initial segmentation. How to incorporate the information of the borders on the general region-merging approach is one of the things we can face on future researches.

Generate pairs of simulated images from a knowing ground truth. This will help to the validation of this and others subpixel techniques since there are no much real pairs with such a configuration. What is more, this problem leads to a final degree project in the Facultad de Ingeniería (Faculty of Engineering) which is in course.

Recoding of the merging algorithm. The merging algorithm is not optimized, taking a long time to execute it. This is not practical for production environments, then a recodification of the whole process must be done.

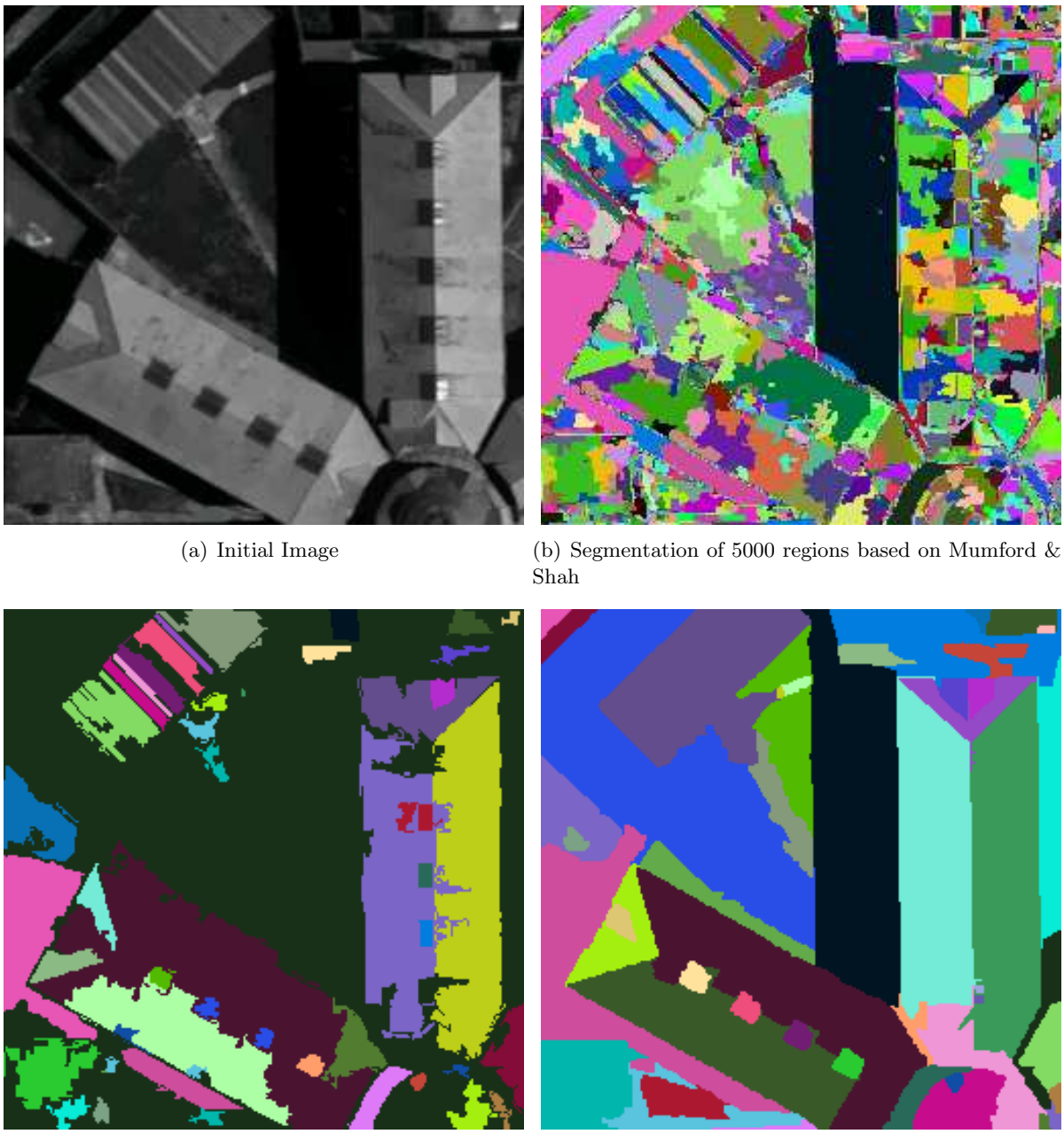
(a) Initial Image

(b) Segmentation of 5000 regions based on Mumford & Shah

(c) Final Segmentation (61 regions)

(d) M & S Segmentation with 61 regions

Figure 8.1: Piecewise constant region-merging process used for segmentation. Note the the main drawback of the region-merging algorithm is the lack of regularization of the borders.



(a) Initial Image

(b) Segmentation of 5000 regions based on Mumford & Shah

(c) Final Segmentation (50 regions)

(d) M & S Segmentation with 50 regions

Figure 8.2: Another experiment of the piecewise constant region-merging process for segmentation

Continuous NFA. The computation of the NFA using the Hoeffding inequalities is an approximation of the real NFA. The implementation of the discrete convolution to obtain the real probability is of interest too. It can be added the study of functions which the derivative of its inverse can be easily convolved, in order to obtain the same result but analytically.

From a more theoretical point of view, the incorporation of more complex models to the description of a real scenes and the validation of them using similar techniques as the one presented in this work seems to be an interesting research to be carried out. If we can develop validation criteria to constant models, to two inclined roof models, and so on, we can test each region of the scene with each one of them and keep the one with the best explanation. The definition of these criteria based on accurate *a contrario* models enables us to compare them directly, without any assumption or *a priori* knowledge.

Appendix A

Analysis of robust estimators

Given a set of values, in our case disparity values within a region, we want to obtain the model that best fits these data values (in our case, the affine model that best fit the disparity values). This estimation can be done in several ways, which is the main subject of this chapter. Here we review the most classical estimation techniques. We start by analyzing the classical Least-Squares. Although this method is easy to be implemented, its performance is degraded in the presence of outliers. This motivates us to investigate other estimation techniques, more robust to the presence of these outliers. For a review on these and other methods see [59, 55].

A.1 Least Squares

Given a region R with n points, we want to find the values (a, b, c) that minimize the following system:

$$\begin{cases} ax_1 + by_1 + c = d_1 \\ ax_2 + by_2 + c = d_2 \\ \vdots \\ ax_n + by_n + c = d_n \end{cases}$$

Written in matrix notation is:

$$\mathbf{A}\mu = \mathbf{b}$$

with

$$\mathbf{A} = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & & \\ x_n & y_n & 1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix} \text{ y } \mu = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

If we consider L^2-norm, we have to minimize the following functional:

$$J(\mu) = \| \mathbf{A}\mu - \mathbf{b} \|_2^2$$

$$\Rightarrow J(\mu) = < \mathbf{A}\mu - \mathbf{b}, \mathbf{A}\mu - \mathbf{b} >$$

$$\Rightarrow J(\mu) = (\mathbf{A}\mu - \mathbf{b})^T(\mathbf{A}\mu - \mathbf{b})$$

$$\Rightarrow J(\mu) = \mu^T\mathbf{A}^T\mathbf{A}\mu - \mu^T\mathbf{A}^T\mathbf{b} - \mathbf{b}^T\mathbf{A}\mu - \mathbf{b}^T\mathbf{b}$$

Deriving J we obtain:

$$J(\mu)' = 2\mathbf{A}^T\mathbf{A}\mu - 2\mathbf{A}^T\mathbf{b}$$

Finally, the vector μ that minimizes J is the solution of the *normal equations*:

$$\mathbf{A}^T\mathbf{A}\mu = \mathbf{A}^T\mathbf{b} \tag{A.1}$$

If $\mathbf{A}^T\mathbf{A}$ is rang complete, we can invert it to obtain

$$\mu = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

Nevertheless, in our case many of the rows are linearly dependent, so the range of the matrix $\mathbf{A}$ is smaller than the matrix dimension. We can use the *Singular Value Decomposition* (SVD) to obtain μ from Eq. A.1 []:

Theorem 2 *If* $\mathbf{A}$ *is a real* $m \times n$ *matrix, then there exist two orthogonal matrices* $\mathbf{U} \in \mathbb{R}^{m \times m}$ *and* $\mathbf{V} \in \mathbb{R}^{n \times n}$ *such that*

$$\mathbf{U^T AV} = diag(\sigma_1, \sigma_2, ..., \sigma_n) \in \mathbb{R}^{m \times n}$$

where $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n \geq 0$

If we decompose matrix $\mathbf{A}$ in its **SVD** decomposition we obtain:

$$\mu = \mathbf{A^{-1}b} = (\mathbf{U\Sigma V^T})^{-1}\mathbf{b} = \sum_i \frac{u_i^T b}{\sigma_i} v_i \tag{A.2}$$

which shows how unstable becomes the solution for small σ. This last equality is obtained in the following way:

$$(\mathbf{U\Sigma V^T})^{-1}\mathbf{b} = (\mathbf{V^T})^{-1}(\mathbf{U\Sigma})^{-1}\mathbf{b} = (\mathbf{V^T})^{-1}(\mathbf{\Sigma})^{-1}\mathbf{U^{-1}b}$$

Then, as $\mathbf{U}$ and $\mathbf{V}$ are orthogonal:

$$(\mathbf{U\Sigma V^T})^{-1}\mathbf{b} = \mathbf{V\Sigma^{-1}U^T b}$$

and finally using the fact that $\mathbf{\Sigma}$ is diagonal, we obtain Eq A.2. If we reduce the dimension of the matrix leaving out the rows where σ_i is very small, we can obtain a more stable system.

A.1.1 Error estimation

Once we have estimated μ^*, we want to find which is the error in the estimation in relation to the real (and unknown) estimation μ (From now on, μ designs the real estimation whereas μ^* designs the estimated one.
Given a region R with n valid points, we note ϵ_i the ground truth at each point $\mathbf{x}_i \in R$ (that is, the real value we want to obtain and that we do not know). We also assume that each observed value d_i has a component of some additive noise r_i with normal distribution $N(0, \sigma)$:

$$\epsilon_i = d_i + r_i$$

If $\mu^* = (a, b, c)^T$, we have for each point x_i of the region:

$$d_i = ax_i + by_i + c$$

or in matrix notation:

$$\mathbf{d} = \mathbf{A}\mu^*$$

So, in one hand, we have the computation of the estimation for the observed data:

$$\mathbf{d} = \mathbf{A}\mu^*$$

and on the other hand, the real disparity values:

$$\epsilon = \mathbf{A}\mu$$

that we can link by

$$\epsilon - \mathbf{d} = \mathbf{r}$$

The previous three equalities enable us to write the following relation:

$$\mathbf{r} = \epsilon - \mathbf{d} = \mathbf{A}\mu - \mathbf{A}\mu^* = \mathbf{A}(\mu - \mu^*)$$

Again, this matrix $\mathbf{A}$ could not be invertible, so we have to obtain its SVD or QR decomposition. To simplify the notation, we consider a QR decomposition of the matrix A (if we have a SVD decomposition $A = USV^T$, we can define $Q = U$ and $R = SV^T$).
We have

$$\mathbf{r} = \mathbf{QR}(\mu - \mu^*)$$
$$\Rightarrow \mu - \mu^* = \mathbf{R^{-1}Q^T r}$$

The covariance matrix of μ is

$$cov(\mu) = E[(\mu - \mu^*)(\mu - \mu^*)^T] = E[\mathbf{R^{-1}Q^T r}(\mathbf{R^{-1}Q^T r})^T] = E[\mathbf{R^{-1}Q^T r r^T Q R^{-T}}]$$

but matrices $\mathbf{Q}$ y $\mathbf{R}$ are constant because they are the decomposition of the data matrix $\mathbf{A}$, so,

$$cov(\mu) = \mathbf{R^{-1}Q^T}E[\mathbf{rr^T}]\mathbf{QR^{-T}}$$

Finally,

$$cov(\mu) = \sigma^2\mathbf{R^{-1}R^{-T}}$$

because we have assumed that the error in the measures have a σ^2 variance.
After obtaining the error at the parameters, we can compute the error made when we interpolate the data using the obtained parameters instead of the "real" ones. Given m points $\mathbf{x}_i$ of a region we have:

$$d_i = \mathbf{x_i}\mu^*$$

which is the computation done using the observed data, and

$$\epsilon_i = \mathbf{x_i}\mu$$

which is the computation done using the real (unknown) data.
So, we can compute the covariance matrix at each point $\epsilon_\mathbf{i}$:

$$cov(\epsilon_\mathbf{i}) = E[(\epsilon_i - d_i)(\epsilon_i - d_i)^T] = E[(\mathbf{x_i}\mu^* - \mathbf{x_i}\mu)(\mathbf{x_i}\mu^* - \mathbf{x_i}\mu)^T]$$

$$\Rightarrow cov(\epsilon_\mathbf{i}) = E[\mathbf{x_i}(\mu^* - \mu)(\mathbf{x_i}(\mu^* - \mu))^T] = E[\mathbf{x_i}(\mu^* - \mu)(\mu^* - \mu)^T(\mathbf{x_i})^T]$$

$$\Rightarrow cov(\epsilon_\mathbf{i}) = \mathbf{x_i}E[(\mu^* - \mu)(\mu^* - \mu)^T](\mathbf{x_i})^T = \mathbf{x_i}cov(\mu)\mathbf{x_i}^T$$

Summing the error at every point, we have an estimation of the variance:

$$\sigma_i^2 = \frac{1}{|R|}\sum_R \mathbf{x_i}cov(\mu)\mathbf{x_i}^T$$

As an example, if we consider a constant model $d_i = c$ instead of the affine one, matrix $\mathbf{A}$ becomes a column of 1, matrix $\mathbf{R}$ has only one value which is $\sqrt{n}$, and as $\mathbf{R}^{-1} = \mathbf{R}^{-T}$, we obtain the classical value of the covariance:

$$cov(\mu) = \frac{\sigma^2}{n}$$

A.1.2 Rescaling the data

The coordinates of each point from which we estimate the affine transformation have as center the top left corner of the image. To use this system coordinates could be a problem in the computation of the parameters, so after starting the computation, we rescale them to the barycenter of each region: $b = \frac{1}{|R|}\sum_{x_i \in R} x_i$, with $b = (x_0, y_0)$ (for simplicity we only displace the center and we do not rotate the axes). The new coordinate system becomes $x_i' = x_i - x_0$, $y_i' = y_i - y_0$. After that, we perform the computation using these new values and then we obtain the final transformation by:

$$T(x_i', y_i') = a(x_i - x_0) + b(y_i - y_0) + c \Rightarrow T(x_i', y_i') = ax_i + by_i + c - ax_0 - by_0$$

$$\Rightarrow T(x_i, y_i) = ax_i + by_i + c'$$

with $c' = c - ax_0 - by_0$

A.1.3 Weighted least squares

With the method explained before, all disparity measures are considered equally reliable. Sometimes however, this is not the case. For example, points near the boundaries of the regions may have more information than points inside them. In order to model this fact we can associate a weight to each measure. This method is known as *weighted least squares*. Similar to least squares, where we want to minimize $(\mathbf{A}\mu - \mathbf{b})^2$, the idea here is to minimize $(\mathbf{WA}\mu - \mathbf{Wb})^2$. That is, to find the best solution

for the equation $\mathbf{WA}\mu^T = \mathbf{Wb}$ where matrix $\mathbf{W}$ is a diagonal matrix.

Both methods are valid if the number of outliers is small. Otherwise, in the general term $\sum(x_i - T(x_i))^2$, the values far from the rest would lead to high error values, and they will have a great impact on the final result.

This can be formalized in the following way:
Consider a more general form:

$$\min_{\mathbf{x}} \sum_i \rho(r_i) \tag{A.3}$$

where ρ is some symmetric positive function with only one minimum at 0 and r_i depends on $\mathbf{x}$ (in fact is the i-th row of the residual $\mathbf{r} = \mathbf{Ax} - \mathbf{b}$). The solution to this minimization is at the point where the derivatives are 0. In our case, vector $\mathbf{x}$ is of the form $(a, b, c)^T$, that leads to the following equation system

$$\frac{\partial \sum_i \rho(r_i)}{\partial a} = 0$$
$$\frac{\partial \sum_i \rho(r_i)}{\partial b} = 0$$
$$\frac{\partial \sum_i \rho(r_i)}{\partial c} = 0$$

applying chain rule,

$$\sum_i \rho(r_i)' \frac{\partial r_i}{\partial a} = 0$$
$$\sum_i \rho(r_i)' \frac{\partial r_i}{\partial b} = 0$$
$$\sum_i \rho(r_i)' \frac{\partial r_i}{\partial c} = 0$$

The derivative function of $\rho(x)$, that we note $\psi(x)$ is known as *influence function* and it measures the influence of a given data on the estimation. For example, in the case of least squares, where $\rho(x) = x^2$, the influence function is $\psi(x) = 2x$.That means that the influence of a given data grows linearly with its value: the greater the error r_i, the larger is its influence on the final estimation.

A.2 Robust Estimators

A.2.1 M-Estimators

M-Estimators [35] is a technique to estimate the general form given in A.3 using different forms of the function ρ. Going back to the general estimation $\min_x \sum_i \rho(r_i)$, we can define from the influence function $\psi(x)$, a *weighted function*

$$w(x) = \frac{\psi(x)}{x}$$

With this new function, equations

$$\sum_i \rho(r_i)' \frac{\partial r_i}{\partial a} = \sum_i \psi(r_i) \frac{\partial r_i}{\partial a} = 0$$
$$\sum_i \rho(r_i)' \frac{\partial r_i}{\partial b} = \sum_i \psi(r_i) \frac{\partial r_i}{\partial b} = 0$$
$$\sum_i \rho(r_i)' \frac{\partial r_i}{\partial c} = \sum_i \psi(r_i) \frac{\partial r_i}{\partial c} = 0$$

can be rewritten as

$$\sum_i w(r_i) r_i \frac{\partial r_i}{\partial a} = 0$$

$$\sum_i w(r_i) r_i \frac{\partial r_i}{\partial b} = 0$$

$$\sum_i w(r_i) r_i \frac{\partial r_i}{\partial c} = 0$$

This system is the same that we obtain if we solve the iterated weighted least squares problem [27]

$$\min_x \sum_i w(r_i^{k-1}) r_i^2, k \geq 1$$

where at each step, we solve a weighted least squares problem using the information obtained in the previous iteration. The first step, $k = 0$ is obtained by a simple least squares problem.

A wide number of weighted functions exist. We did our experiments using the *fair function*, but other ones give similar results.

$$w_i = \frac{1}{1 + |r_i|/c}$$

where c is a constant that we set to 1.3998 to obtain 95% efficiency considering a standard normal distribution (so we have to rescale the error measures: $r_i' = \frac{r_i - \mu}{\sigma^2}$ in order to use the same constant).

A.2.2 Least Median Squares

LMS [48] finds the best parameters by minimizing the non linear problem::

$$\min_i \text{med} r_i^2$$

To solve this problem we search for the solution that best fit at random. Obviously, the search is not exhaustive because of the size of the search space. In general, a Monte Carlo like method is used to perform the search. This method can be summarize as: Given n points x_i:

1. We build m sets of 3 points at random.
2. For each of these sets, we compute the affine transformation T_j.
3. For Each T_j, we compute the error made in the other points and we find the median of the square of these errors:
$$M_j = \text{med}_{\{1..n\}} r_i^2$$
4. The solution to the problem is the transformation with minimal M_j.

This way of building the transformations to be tested does not analyze the entire space of solutions because the transformations are generated from 3 points of the region. This method gives a good approximation but once we have it, we must perform another optimization to obtain better results, by using M-Estimators for example, but with the transformation obtained with LMS as the initial one (remember that in M-Estimators we obtain the first set of parameters by LS directly).

A.2.3 Least Trimmed Squares

The LTS estimator [48] is of the form

$$\min \sum_{i=1}^{k} r_i^2$$

where the r_i are the non decreasing ordered squares residuals. Usually, $h = (n + 1)/2$. We also use a random technique like the one explained before to generate the transformation to be tested but

instead of choosing 3 points we select h. With these h points we compute the transformation using some estimator like LS or the M-Estimator presented before, and we compute the variance error for the entire set of points when using this transformation. The solution will be the one with lower error variance.

A.2.4 RANSAC

This method [29] maximizes the number of points in the region that satisfy the model within a given tolerance. Again, the transformation to be tested is generated randomly. For each transformation we compute the error. In general, we would estimate the variance of the residuals for each transformation, considering only those points which their residuals are smaller than the computed variance. This method is very sensitive to the variance: an over estimation could accept outliers points whereas an subestimation could leave valid points out. We need to estimate a variance robustly to avoid the influence of outliers. In the next section we discuss different techniques to estimate the variance robustly.

A.3 Variance (scale) estimation

One of the most difficult problems when using robust estimators is how to find the correct estimation for the variance, without being influenced by the outliers presented in the data. This is because in general, these methods discard or give less influence to points with an error greater than the some given value related to the variance. We have analyzed some of them based on [55].

A.3.1 Standard Deviation

The standard deviation is obtained by

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - T(x_i))^2}$$

where N is the number of points in the region and T is the obtained transformation.
This estimation is strongly biased by the outliers as we have already seen when analyzing the influence function for the least squares problem. If we use this estimation, the variance would be overestimated.

A.3.2 Median and Median Absolute Deviation

In this case, the summation performed in the computation of the standard deviation is replaced by a rank ordering of the errors. A well known robust median scale estimator is given by:

$$\sigma = 1.4826\left(1 + \frac{5}{n-p}\right)\sqrt{\text{med}_i r_i^2}$$

A variant of this estimation is the Median Absolute Deviation (MAD) [33]:

$$\text{MAD} = 1.4826\ \text{med}_i\{r_i - \text{med}_j r_j\}$$

The factor 1.4826 is added to obtain consistency with the standard deviation for asymptotically normal distributions. Note that this estimation does not need the values to be centered.
This estimation is robust if we have less than 50% of outliers, but it would give very wrong results if the number of outliers is greater than that.

Method	ALL 100.00 %	SHADOW 77.01 %	VALID 23.19 %	VALID ∪ SHADOW 18.48 %
Input	0.315763	0.272751	0.259072	0.210837
LS	0.263253	0.196615	0.208505	0.160747
LS-SVD	0.263035	0.197530	**0.207014**	0.160709
LS-SVD Rescaled	0.262827	0.195871	0.207565	0.159404
M-Est	0.264092	**0.194467**	0.208590	**0.158902**
M-Est SVD Rescaled	**0.261966**	0.194831	0.207382	0.159683
LMS + M-Est	0.264138	0.197671	0.212597	0.165037
LTS + M-Est	0.262323	0.195313	0.208019	0.160479
RANSAC + M-Est	0.264167	0.197623	0.212555	0.164996

Table A.1: Error table of different estimators. The initial disparity map was generated by MARC with a precision of 0.25 and the error values of this initial map are shown in the "Input" row. This table was generated from an initial segmentation of 1000 regions (Figure 6.2(b))

A.3.3 Residual Consensus (RESC) Method

In [58] authors propose to work with the histogram of the residuals. They first build the histograms of the absolute values of the errors and then they compressed it, cutting off the tail, when the number of points in the bin is less than a percentage of the number of points in the first bin. Then, the deviation is defined as:

$$\sigma^2 = \frac{1}{\sum_{i=1}^{v} h_i - 1} \sum_{i=1}^{N} (r_i - \bar{h})^2$$

where v is the number of columns of the histogram and $\bar{h}$ is the mean of all residuals included in the histogram. Note that if we do not cut off the histogram, the deviation is the same as the standard deviation.

A.4 Experiments

In this section we show an analysis of the performance of the algorithm using different estimation techniques. All the estimations have been done with the same disparity map generated by MARC with a precision of 0.25 presented in Figure 7.2 of Chapter 7.

We start analyzing the performance of different estimators without merging. That is, for each given region in the original segmentation, we compute the affine transformation using different estimators. The used segmentation has 1000 regions and have been obtained using the piecewise constant Mumford and Shah method explained in Chapter 6. Table A.2 shows the error of each estimator when compared to the ground truth. First of all, we can see a reduction of the global error when compared to the original data (MARC, first row). There is also an improvement in the results when we change the method to solve the LS problem (both for LS directly and for the M-Estimator). On the other hand, rescaling the coordinates does not improve the results. For the random estimators (LMS, LTS and RANSAC) we find that the results are improved considerably when we adjust the obtained transformations by an M-Estimator. That is, we start the M-Estimator not from a LS result but for one of the random process. Although the best error computed over the entire shadow mask and over the mask and the valid points is M-Estimator, the other one is not very different (0.144 vs 0.148). Besides, if we consider the valid regions, all robust methods have the same performance: 0.139. Table A.3 shows the same analysis using an initial segmentation with more regions (3500).

The second thing we want to analyze is the estimation of the variance since all robust estimator use it in one way or the other to eliminate outliers. Table A.4 shows the results obtained using different variance estimations for the same robust technique. Again, the difference between the best and the worst result is 0.009, being RESC the one which leads to the best result.

Method	ALL 100.00 %	SHADOW 77.01 %	VALID 41.08 %	VALID ∪ SHADOW 31.93 %
Input	0.119538	0.112692	**0.042609**	**0.044797**
LS	0.076565	0.072018	0.059129	0.061561
LS-SVD	0.079399	0.075861	0.062568	0.066221
LS-SVD Rescaled	0.076052	0.071513	0.058229	0.060675
M-Est	**0.073100**	**0.068485**	0.055590	0.057747
M-Est SVD Rescaled	0.077273	0.072963	0.058642	0.060960
LMS + M-Est	0.079626	0.075217	0.062839	0.064920
LTS + M-Est	0.077516	0.073284	0.059101	0.061427
RANSAC + M-Est	0.079629	0.075216	0.062830	0.064905

Table A.2: Error table of different estimators using the simulated pair of images. This table was generated from an initial segmentation of 1000 regions (Figure 6.2(b))

Method	ALL 100.00 %	SHADOW 77.01 %	VALID 23.19 %	VALID ∪ SHADOW 18.48 %
Input	0.315763	0.272751	0.259072	0.210837
LS	0.280633	0.216457	0.226400	0.178726
LS-SVD	**0.278526**	0.219865	**0.222260**	0.179959
LS-SVD Rescaled	0.280663	0.216400	0.226308	**0.178522**
M-Est	0.281650	**0.214947**	0.228415	0.179331
M-Est SVD Rescaled	0.281491	0.215855	0.228653	0.181245
LMS + M-Est	0.289105	0.229117	0.243314	0.199465
LTS + M-Est	0.282562	0.219554	0.231217	0.184635
RANSAC + M-Est	0.289002	0.229294	0.243309	0.199594

Table A.3: Error table of different estimators. The same as Table A.2 but in this case we use a over-segmented image of 3500 regions

Method	ALL 100.00 %	SHADOW 77.01 %	VALID 23.19 %	VALID ∪ SHADOW 18.48 %
Input	0.315763	0.272751	0.259072	0.210837
SD	0.267170	0.192328	0.209036	0.156202
MAD	0.270542	0.192613	0.210534	0.156638
RESC	0.261966	0.194831	0.207382	0.159683
ROB	0.270476	0.192678	0.210538	0.152143
L1	0.268492	0.192655	0.209755	0.156434

Table A.4: Analysis of the different variance estimations using the same robust estimation technique: the M-Estimator SVD Rescaled

A.5 Conclusions

When we compare the different estimation techniques presented before, if we use robust estimators against standard ones like LS, we see an improvement in the final results. Nevertheless, using M-Estimator directly gives similar values of error. The main reason is that in this case we are initializing the M-Estimator with a first set of parameter computed with LS. If we start M-Estimator with a set of parameters computed using one of the random techniques explained before (LTS, LMS, RANSAC) the results are improved. The three random techniques tested, have a similar performance. Maybe RANSAC gives in average, the best results.
However, all the estimation techniques show similar results: in the case of the points computed over the shadow mask the lowest and greatest values range from 0.1944 to 0.1976 for the real set of images. In the case of the simulated ones, if we consider the error in the whole image these values come from 0.073 to 0.079. Note however that the error is not improved in the valid points of MARC. Thus, the robust estimation of an affine transformation does not guarantees the improvement on the valid points. Nevertheless, as we have seen on Chapter 7, the error is improved when we use the region-merging algorithm.
We can conclude that the estimation used to obtain the different transformation is not determinant in the final results, being the M-Estimator with an initialization using RANSAC the best of all.

Appendix B

Data Simulation

The aim of this chapter is how we can obtain a simulated stereo pair based on an exact ground truth information. Even if the ground truth available for our data is very precise the localization of the borders is not perfect (even after the registration, see Figure B.1), and this affects the error measures specially when the correlation algorithm is very sensitive to the information over the borders of the structures.

We are interested in simulating the stereo pairs with very small baseline, almost without occlusions according to the model proposed by Delon and Rougé in [19, 20] [1] where each pixel of the reference image u can be matched to the pixels of the secondary image $\tilde{u}$ over the same epipolar line at a distance ε (the disparity):

$$\tilde{u}(x + \varepsilon(x)) = u(x)$$

This kind of simulation can be obtained by two means, the first one is to consider a 3D terrain model with textures and render the projections corresponding to the two pairs, the second approach (the one documented here) uses an irregular sampling of one image according to the disparity model.

The first option is discarded since generally it is not a requisite for the ray-tracing software to support very small displacements, but more important, there is no guarantee that the simulation is performed taking care of the sampling restrictions (an option that could be simulated at a very high resolution and subsample the result). There exists software to simulate stereo pairs like the one provided by Volker Gerdes (http://www-student.informatik.uni-bonn.de/~gerdes/MRTStereo/index.html) but we have not tested any yet.

B.1 Simulation using irregular sampling

To model the stereo images with $\tilde{u}(x + \varepsilon(x)) = u(x)$ is based on unrealistic assumptions, namely that the terrain can be modeled with a smooth function and that there are no occluded areas in the images. If we name the mapping function $\Phi(x) = x + \varepsilon(x)$, there will be an occlusion in two different situations:

[1] In the proposed model the disparity is proportional to the altitude of the object.

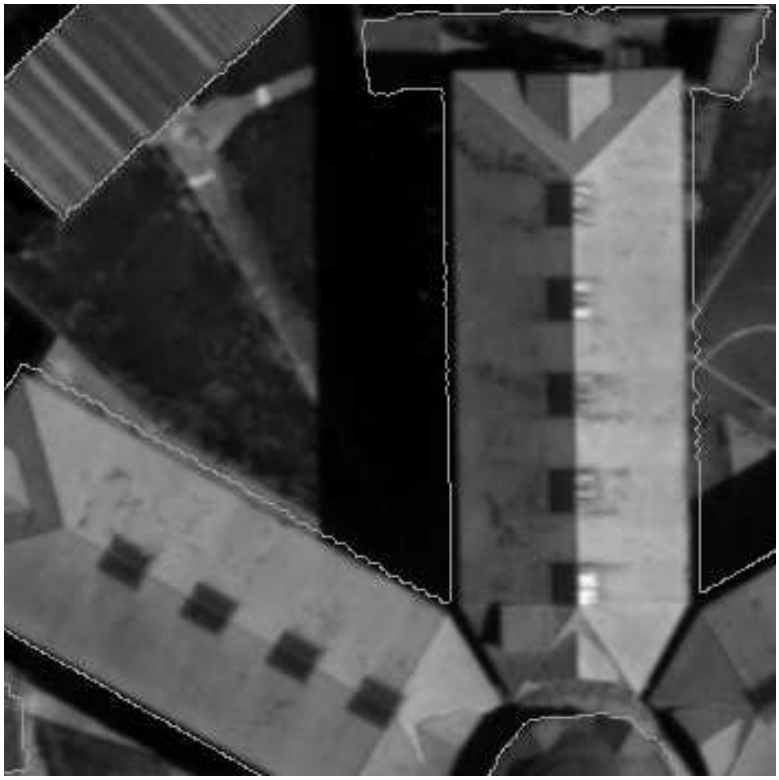


Figure B.1: Errors in the localization of the borders of the ground truth.

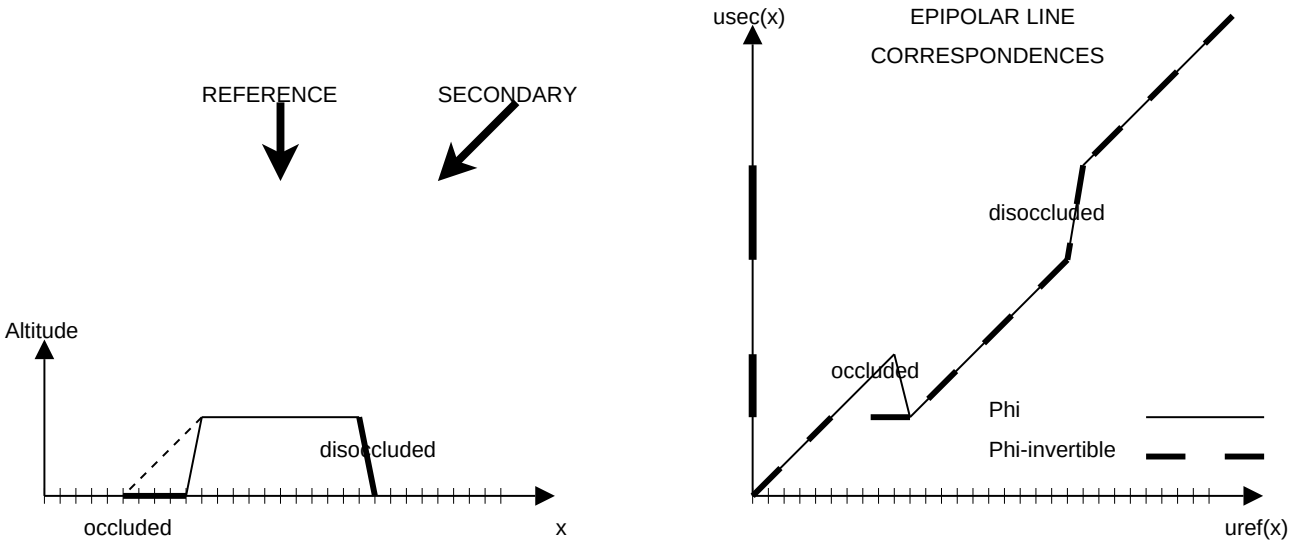


Figure B.2: Problems that appear when inverting a disparity function $\varepsilon(x)$ (left image), the pseudo-inverse of the function $\Phi(x) = x + \varepsilon(x)$ that represents correctly the occlusions is shown in the image of the right.

occlusion If the function $\Phi(x)$ is decreasing then some parts of the reference image u will not be visible in the secondary image $\tilde{u}$. When simulating $\tilde{u}$ from u we need to simulate these occlusions. Recall that function Φ is decreasing in $[a, b]$ if $\Phi(x)' < 0$ for all $x \in (a, b)$, which leads to $\varepsilon'(x) \leq -1$.

disocclusion If $|\Phi'(x)| > 1$ (i.e. if $\varepsilon'(x) \geq 0$) then the sampling rate of the secondary image $\tilde{u}$ will be greater than in the reference image, which means that more detail can be represented in $\tilde{u}$ than in u. When simulating $\tilde{u}$ from u the problem here is how to "disocclude" this information missing in u.

Simulating occlusions In urban terrains (due to the ubiquity of vertical walls) non-decreasing functions are very unlikely, then the function $\Phi(x)$ will not be invertible. On the other hand if we want to obtain an image $\tilde{u}$ such that fulfills the disparity property we may need to compute its values over a regular grid

$$\tilde{u}(x) = u(\Phi^{-1}(x)) \quad \text{with } x \in Z^2$$

which means interpolating u on the irregular (perturbed) grid $\Phi^{-1}(Z^2)$.

These observations lead to the need of inverting the (not always invertible) function $\Phi(x)$, and in order to solve the problem we must compute a pseudo-inverse of this function such that the occluded areas are correctly represented. An occlusion occurs when the same point in the second image (the vertical axis in Figure B.2) correspond to more than one point in the reference image (horizontal axis), the computation of the pseudo-inverse must solve this ambiguity. The solution consists of keeping the highest point value in the areas where the inverse has multiple functional values, as expressed in equation (B.1) and shown in Figure B.2.
That is, from all x that satisfy $\Phi(x) = y$, we define the pseudo-inverse Φ^+ of y to be the maximum of those x:

$$\Phi^+(y) = \arg\max_x(\Phi(x) = y) \tag{B.1}$$

The current implementation of the pseudo-inverse algorithm uses linear interpolation of the existing samples of Φ (a zoomed version) to compute the function. By building a list of intervals $[\Phi(x), \Phi(x+1)]$ where the functional values are known and then obtaining the inverse at integer positions: $y \backslash \Phi(y) = i$ with $i \in Z$, and only keeping the highest y value.

Simulating disocclusion Another problem related to the simulation is the disocclusion. These areas appear as a dragging of the last known value along the disoccluded area. The proposed solution uses a real image (the secondary image) from where the missing information is copied, another option could be to fill the gaps with white noise so that the correlation algorithm will not match the areas. The candidate zones for disocclusion are those where $\varepsilon'(x) > 1$ or equally $\Phi'(x) > 2$. In these areas one or more pixels visible in $\tilde{u}$ are "crunched" between two neighboring pixels in u.

The spectral support

One last detail about the simulation of the images concerns the spectral support of the images. During the irregular sampling of the image some high frequencies could grow outside the actual support of the image, to avoid this we shall extend the support of the reference image before the sampling, and after the sampling we shall apply a prolate filter to reduce the spectrum and subsample without aliasing.

B.2 Simulation algorithm

The final simulation process considering all the details previously discussed is:

1. Perform a 2x zoom of the reference and secondary (u and u_{sec}) images using zero-padding Fourier interpolation, the ground truth (ε) is zoomed using a 1st order spline interpolation. The zoom is needed because the spectral support of the irregular sampled images could grow with respect of the original images.

2. Compute the pseudo-inverse of $\Phi(x) = x+\varepsilon(x)$, as well as the occluded areas and the disoccluded areas. The occluded areas can be identified as points where $\Phi(x) \leq 0$, while the disoccluded are those where $\Phi(x) > 2$. It is important to notice that the disocclusion map must be computed with the coordinates of the simulated image $\tilde{u}$.

3. Resample the reference image according to the irregular grid $\tilde{u} = u(\Phi^{-1}(x))$. This step uses nfft_test developed by Gloria Haro.

4. The computed secondary image $\tilde{u}$ will show artifacts in the disoccluded areas, because that information is not present in the reference image. The gaps of the disoccluded areas are extracted from the REAL secondary image and superimposed over the resampled image.

5. The resulting secondary image $\tilde{u}$ will be a "correct" simulation of the secondary image, the last step consists of reducing the spectrum (with a prolate filter) and subsample both images (u and $\tilde{u}$).

6. Finally a gaussian noise ($\sigma = 3.5$) is added to both images.

Bibliography

[1] http://www.cmla.ens-cachan.fr/cmla/megawave/index.html.

[2] http://www.middlebury.edu/stereo/.

[3] A. Almansa. *Échantillonnage, interpolation et détection. Applications en imagerie satellitaire.* PhD thesis, École Normale Supérieure de Cachan, December 2002.

[4] A. Almansa, J. Delon, G. Facciolo, L. Igual, A. Pardo, J. Preciozzi, and B. rougé. Small baseline stereo for urban digital elevation models using variational and region-merging techniques. Technical report, Facultad de Ingeniería, Universidad de la República, Uruguay, 2006. Informe final. Proyecto PDT SC/OP/17/01.

[5] A. Almansa, A. Desolneux, and Vamech. Vanishing points detection without any a priori information. *IEEE PAMI*, 25(4), april 2003.

[6] C. Ballester, V. Caselles, L. Igual, and L. Garrido. Level lines selection with variational models for segmentation and encoding. To be published in Journal of Mathematical Imaging and Vision, 2005.

[7] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Proceedings of the Seventh International Conference on Computer Vision*, pages 489–495, September 1999.

[8] J-M. Bony. *Cours d'analyse.* Les Éditions de L'École Polytechnique, Janvier 2001. Théorie des distributions et analyse de Fourier.

[9] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in Computational Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):993–1008, 2003.

[10] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. 61(3):211–231, 2005.

[11] N. Camlong. Report cssi/111-1/cor-et-marc-2, description de l'algorithme multiresolution algoritme to refine correlation. Technical report, CNES, 2001.

[12] F. Cao, Y. Gousseau, and P.o Musé F. Sur J.M. Morel. Accurate estimates of false alarm number in shape recognition. Technical report, INRIA, 2004.

[13] V. Caselles, B. Coll, and J.-M. Morel. Topographic maps and local contrast changes in natural images. *Int. Journal Comp. Vision*, 33(1):5–27, 1999.

[14] V. Caselles, L. Garrido, and L. Igual. A Contrast Invariant Approach to Motion Estimation. In *International Conference on Scale Space*. Springer Verlag, 2005.

[15] A. Chambolle. Total variation minimization and a class of binary mrf models. École Polytechnique - Centre de Mathématiques Appliquées, June 2005.

[16] H. Chen, P. Belhumeur, and D. Jacobs. In search of illumination invariants. In *Int. Conf. on Computer Vision and Pattern Recognition*, pages 254–261, 2000.

[17] J. Darbon and M. Sigelle. Image restoration with discrete constrained total variation part i: Fast and exact optimization. *Journal of Mathematical Imaging and Vision*.

[18] J. Darbon and M. Sigelle. Image restoration with discrete constrained total variation part ii: Levelable functions, convex and non-convex cases. *Journal of Mathematical Imaging and Vision*.

[19] J. Delon. *Fine comparison of images and other problems.* PhD thesis, Ecole Normale Superieure de Cachan, 2004.

[20] J. Delon and B. Rougé. Analytic study of the stereoscopic correlation, 2004.

[21] A. Desolneux. *Événements significatifs et applications à l'analyse d'images*. PhD thesis, École Normale Supérieure de Cachan, December 2000.

[22] A. Desolneux, S. Ladjal, L. Moisan, and J.-M. Morel. Dequantizing image orientation. *IEEE Trans. on Image Proc.*, 10:1129–1140, 2002.

[23] A. Desolneux, L. Moisan, and J.-M. Morel. Edge detection by Helmholtz principle. *Journal of Mathematical Imaging and Vision*, 14(3):271–284, 2001.

[24] A. Desolneux, L. Moisan, and J.-M. Morel. Maximal meaningful events and applications to image analysis. *Annals of Statistics*, 31(6):1822–1851, 2003. submitted july 2000.

[25] A. Desolneux, L. Moisan, and J-M. Morel. Gestalt Theory and Image Analysis. A probabilistic Approach, February 2006.

[26] G. Facciolo, A. Almansa, and A. Pardo. Variational approach to interpolate and correct biases in stereo correlation. In *GRETSI*, Louvain-la-Neuve, Belgique, September 2005.

[27] O. Faugeras, Q.-T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images : The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. MIT Press, 2001. FAU o 01:1 1.Ex.

[28] W. Feller. *An Introduction to Probability Theory and Its Applications*. John Wiley & Sons, Inc.

[29] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24:381–395, 1981.

[30] L. Garrido and P. Salembier. Region based analysis of video sequences with a general merging algorithm. European Signal Processing Conference (EUSIPCO), Sept 1998.

[31] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, March 2004.

[32] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[33] P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Commun. Statist.-Theor. Meth.*, A6:813–827, 1997.

[34] B K P Horn and B Schunk. Determining optical flow. 17:185–204, 1981.

[35] P.J. Huber. *Robust Statistics*. John Wiley & Sons, 1981.

[36] S.S. Intille and A.F. Bobick. Incorporating intensity edges in the recovery of occlusion regions. *Proc. Intl Conf. Pattern Recognition*, 1:674–677, 1994.

[37] G. Kanizsa. *La Grammaire du Voir*. arts et sciences. ï¿½itions Diderot, 1997.

[38] G. Koepfler, C. Lopez, and J. M. Morel. A multiscale algorithm for image segmentation by variational method. *SIAM*, 31(1):282–299, February 1994.

[39] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions via graph cuts. pages 508–515.

[40] F. Lecumberry. Cálculo de disparidad y segmentación de objetos en secuencias de video. Master's thesis, Facultad de Ingeniería, Montevideo, Uruguay, Agosto 2005.

[41] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of the 7th IJCAI*, pages 674–679, Vancouver, Canada, 1981.

[42] J.-M. Morel and S. Solimini. *Variational methods in image segmentation.* Birkhäuser, Boston, 1995.

[43] D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 22–26, 1985.

[44] V. Muron. Report cssi/111-1/cor-et-marc-5, manuel utilisateur de la chaine de calcul de d'ecalages entre images par l'algorithme marc. Technical report, CNES, 2003.

[45] P. Musé. *On the definition and recognition of planar shapes in digital images.* PhD thesis, École Normale Supérieure de Cachan, 2004.

[46] L. Igual Mu noz. *Image Segmentation and Compression using The Tree of Shapes of an Image. Motion Estimation.* PhD thesis, Universitat Pompeu Fabra, October 2005.

[47] N.R. Pal and S.K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, September 1993.

[48] P.J.Rousseeuw. Least median of squares regression. *Journal of American Statistical Association*, 79:871–880, 1984.

[49] M. Pollefeys. Visual 3d modeling from images. Technical report, University of North Carolina - Chapel Hill USA.

[50] J. Preciozzi. Urban elevation maps from stereo images by affine region models. Master's thesis. Mémoire DEA, CMLA ENS-Cachan (France).

[51] M. Rodriguez, A. Almansa, and J. Preciozzi. Cálculo de mapas de elevación de zonas urbanas con alta precisión. Proyecto de Grado - En curso, 2006.

[52] S. Roy and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *ICCV*, pages 492–502, 1998.

[53] D. Scharstein and R. Szeliski. A taxomomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, April 2002.

[54] J. Serra. Image analysis and mathematical morphology. *Academic Press*, 1982.

[55] C. V. Stewart. Bias in robust estimation caused by discontinuities and multiple structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(8):818–833, 1997.

[56] R. Grompone von Gioi. Polygons significatives. Memoire DEA, February 2006.

[57] R. Grompone von Gioi and J. Jakubowicz. Multisegment detection. Personal comunication.

[58] X. Yu, T.D. Bui, and A. Krzyzak. Robust estimation for range image segmentation and reconstruction. *IEEE T-PAMI*, 16(5):530–538, 1994.

[59] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *International Journal of Image and Vision Computing*, 15(1):59–76, January 1997.