Classifying Estimated Corresponding Points by Delaunay Triangulation

Elizabeth Vargas Vargas

Bachelor Thesis

Universidad del Valle

Escuela de Ingeniería de Sistemas y Computación

Santiago de Cali, June 15, 2012

Classifying Estimated Corresponding Points by Delaunay Triangulation

Elizabeth Vargas Vargas

Bachelor Thesis

To obtain the degree of

Ingeniera de Sistemas

Advisor: Maria Patricia Trujillo Uribe, Ph.D.

Universidad del Valle

Escuela de Ingeniería de Sistemas y Computación

Santiago de Cali, June 15, 2012

Acknowledgements

I would like to express my gratitude to the following people:

To my parents, my sister and my grandmother, because they have been supporting me and taking care of me throughout the duration of my entire studies.

To my advisor, Dr. Maria Trujillo for guiding me during this time and for giving me her trust.

To Ivan Mauricio Cabezas, for sharing all his knowledge and his collaboration along this year.

To my colleagues from the "Laboratorio de Multimedia y Visión", for listening all my ideas, helping me with the presentation and corrections of this work.

Contents

List of Tables

List of Figures

Abstract

Stereo vision has become in one of the most important research topics nowadays. There is a large number of applications in the industry, that goes from searching robots to self-driving cars. These applications work with an error tolerance, that could be improved in order to find more applications in the industry, where accuracy is the most important factor.

The problem has not been solved, given that there are still many points that are incorrectly classified as correspondences. Therefore, there are still many publications about the topic, with the purpose of finding solutions free of error.

Stereoscopic vision means vision with two eyes, which implies that one feature in the 3-Dimensional space is projected onto two planes. The challenge on stereo vision is reconstruct the 3D point, by finding correspondence points in two projections, one from a left plane and one for a right one. The major problem is misclassification, because there are many feature points that may have some similarity.

An algorithm for classifying correspondences based on the Dalaunay triangulation is presented. It is motivated for large misclassified correspondence points that affect the 3D reconstruction.

Chapter 1
Introduction

Finding corresponding points is useful in a very large number of human activities which benefit directly from its non-intrusive character and also from the widespread use cameras have acquired nowadays [7]. In the aerospace business, it is known the use of searching robots that are able to explore the terrain on mars surface for example [16]. Additionally, companies like Google invest millions in research related to self-driving cars [6], focusing on obstacle avoidance. In the health-care sector, image registration is everyday more frequent in the diagnose stage [15], centering on wound depth estimation, internal organs inspection and modeling. Application areas also include entertainment (interactive games), industry (quality control), military (target tracking), robotics (navigation), space (automatic planetary rover in unknown environments) and training (teleoperation) [7].

With all these applications in mind, it is known that the stereo vision problem is important, and finding true correspondences from both images worth it. However, the problem is not still solved, because it is not always an easy task for some kind of images. For this reason, there are written every year thousand of papers with different approaches.

This final career project presents a new approach based on the Delaunay triangulation for classifying a set of correspondence points. Its performance and accurate are evaluated.

1.1 Problem Statement

Stereo Vision refers to the ability to infer information on the 3D structure, such as distance of a scene to the camera using two or more images taken from different viewpoints. The calibration, the correspondence and the reconstruction problems have to be solved in order to infer the 3D structure of a scene. Of these, the most important and the most difficult task is the correspondence problem and it has motivated an immense amount of research [7].

The correspondence problem consists in determining which point on the left image corresponds to which point on the right one [33]. To do that, there have been employed different techniques in order to find the corresponding points given two views. When the correspondence problem is addressed, corresponding points may be inaccurately estimated: falsely matched or badly located. Moreover, small inaccuracies in the estimated corresponding points may have a large impact on the results of the 3D reconstruction [33].

The standard definition of computational vision is that it is inverse optic. The direct problem - the problem of classical optics or of computer graphics - is to calculate the images of three-dimensional objects [13]. Computational vision is confronted with inverse problems of recovering surfaces from images. Much information is lost during the imaging process that projects a three-dimensional world into two-dimensional arrays (images). As a consequence, vision must rely on natural constraints, that is, general assumptions about the physical world, to derive an unambigu-

ous output. This is typical of many inverse problems in mathematics and physics. A problem is well posed when its solution (a) exists, (b) is unique, and (c) depends continuously on the initial data [13]. It implies, that a small variation in the input (true disparity map), produces a large impact on the output (estimated depth).

A great complication in correctly establishing correspondence is the well-known false-target problem [7].Matching points can (in principle) be found at the intersection of the rays passing through these points and the associated pupil centers (or pinholes, see Figure 1.1). Thus, when a single image feature is observed at any given time, stereo vision is easy [4]. Assuming that all six projections appear similar, it is not feasible to choose the correct pairs from all the possible ones, as many consistent matching interpretations are plausible. For instance, in Figure 1.1, ignoring transparency of objects, there are three physically consistent interpretations. In general, the false-target problem refers to the fact that a given feature point in one image may match equally well a number of points in the other image. Such ambiguities can not be resolved, unless further information concerning the projections is provided [7].

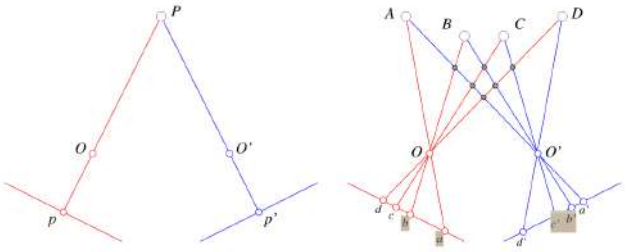

Figure 1.1: Binocular fusion problem

Another important error source are occlusions, which have parts of the scene visible in one image, but hidden in the other image. This is caused by cases where between a surface part of an object and the camera, appears a part of another or the same object, but also because a part is out of the field of one camera. As a result, the stereo matcher may get confused and assign a spurious match. Such a mismatch may also have repercussions in the matching of other, non-occluded points and affect neighbouring correspondences [7].

False matches are caused mainly by photometric differences between the images, i.e. illumination and contrast. Because the cameras view an object from different positions, the reflectivity properties of its surfaces often give projections of different intensities. Also, if during the capturing of the two images, the lighting conditions have changed, intensity variations of conjugate points are very likely. Structural properties of the scene, such as repetitive patterns (e.g. a brick wall), texture or large areas of constant intensity (e.g. surfaces with complete lack of markings and constant shading) increase ambiguity in the matching process [7].

The problem is still open to proposals that improve the quality of the correspondences by eliminating inaccuracies. To overcome the mentioned problems, this final career project analyses the possibilities of using the Delaunay triangulation to classify the correspondences. The question that need to be considered is whether the use of this approach . classifies correctly a set of initial corresponding points as “good/bad matches” or not.

1.2 Proposed Solution

Delaunay triangulation has been widely used by many algorithms in different contexts including block matching. Its uniqueness property makes it a useful tool for finding non-recurring solutions.

The proposed solution uses two different approaches, in order to classify a set of corresponding points as correctly or incorrectly estimated. An overview of the algorithm is presented in 4.1.

The first approach is about triangulating the feature points of each image, and then compare them in order to classify the correspondences. A matching is considered correct if a set of points shape the same triangles in both images. The evaluation approach is described in 4.5. The second approach consists in adding constrains to the triangulation based on the image edges. The criteria established for building the constrains are explained in 4.4.

An algorithm in C++ was implemented and tested with several dataset in order to evaluate its performance. Results are presented on the section 4.7.

1.3 Project Scope

The purpose of the project is to evaluate the pertinence of the use of the Delaunay triangulation to classify a set of corresponding points. For this reason, the final product is not a software, otherwise an evaluation of the accuracy of the proposed algorithm. Two approaches will be implemented, in order to achieve the objective:

- A classification based on the Delaunay triangulation
- A classification based on a constrained Delaunay triangulation

1.4 Objectives

1.4.1 General Objective

Develop an approach to classify a set of corresponding points using the constrained Delaunay triangulation.

1.4.2 Specific Objectives

- Select a robust algorithm to estimate corresponding points.
- Select an effective method to detect edges in order to provide them to the constrained Delaunay triangulation.
- Propose a strategy for classifying a set of corresponding points using the Delaunay triangulation.
- Evaluate the effectiveness of the results, in order to find the best approach to obtain a set of corresponding points free of errors.

Chapter 2

Theoretical Background

2.1 Digital Images Representation

Images measure the quantity of light that infringes on a photosensitive device [33]. A image is characterized by a two-dimensional array, where each element in the array is called a pixel (picture element) [14]. Digital images are represented by a numerical matrix, E, with N rows and M columns. $E(i, j)$ represents the image value at pixel (i, j) (i-th row and j-th column), and encodes the intensity recorded by the photosensors of the CCD (Charged Coupled Device) array contributing to that pixel. $E(i, j)$ is an integer in the range $[0, 255]$ [33].

99	71	61	51	49
93	74	53	56	48
101	69	57	53	54
107	82	64	63	59
114	93	76	69	72
117	108	94	92	97
116	114	109	106	105
115	113	109	114	111
110	113	111	109	106
103	107	106	108	109

Figure 2.1: Gray scale image with a selected region and its correspondence matrix representation

The brightness of an image point can be represented by one byte, or 256 gray levels (typically 0 is black, 255 white). This is an adequate resolution of monochromatic (or gray-level) images and is suitable for many vision tasks. Colour images require three monochromatic component images (red, green, blue) and therefore three numbers [33]. As a result, a colour image is be represented with a three dimensional matrix where each channel is represented by one matrix.

Figure 2.2: Color image and its three components: Red, Green, and Blue.

2.2 Image Pre-Processing

In order to remove the noise presenting in the images, or to obtain a particular set of features, there are some operations applied to the images before we apply the feature detection on them. This section describes the pre-processing techniques using in the algorithm.

2.2.1 Convolution

The procedure of constructing a new array, the same size as the image and afterwards filling each location of this new array with a weighted sum of the pixel values from the locations surrounding the corresponding location in the image, using the same set of weights each time is known as **linear filtering** [4].

The pattern of weights used for a linear filter is usually referred to as the kernel of the filter. The process of applying the filter is usually referred to as **convolution** [4].

Given a filter kernel H, the convolution of the kernel with image F is an image R. The i, j'th component of R are given by:

$$R_{ij} = \sum_{u,v} H_{i-u,j-v} R_{u,v} \tag{2.1}$$

2.2.2 Thresholding

The input to a thresholding operation is typically a grayscale image and the output is a binary image representing the segmentation. The segmentation is determined by a single parameter known as the intensity threshold. In a single pass, each pixel in the image is compared with this threshold. If the pixel's intensity is higher than the threshold, the pixel is set to white in the output. If it is less than the threshold, it is set to black [23].

2.2.3 Morphological Operations

Morphological operators take a binary image and a structuring element as input and combine them using a set operator (intersection, union, inclusion, complement). They process objects in the input image based on characteristics of its shape, which are encoded in the structuring element. The morphological operations used in our algorithm were:

- **Erosion:** The mathematical definition of erosion for binary images is the following: Considering that X is the set of Euclidean coordinates corresponding to the input binary image, and that K is the set of coordinates for the structuring element. Let Kx represents the translation of K so that its origin is at x. Then the erosion of X by K is simply the set of all points x such that Kx is a subset of X [21].
- **Dilation:** The mathematical definition of dilation for binary images is: Considering that X is the set of Euclidean coordinates corresponding to the input binary image, and that K is the set of coordinates for the structuring element. Let Kx represents the translation of K so that its origin is at x. Then the dilation of X by K is simply the set of all points x such that the intersection of Kx with X is non-empty [20].
- **Opening:** Is defined as an erosion followed by a dilation using the same structuring element for both operations [22].
- **Closing:** Is defined as a dilation followed by an erosion using the same structuring element for both operations [19]

2.3 Image Features

2.3.1 Interest Points

FAST Corners Detector

Corners are positions in the image containing rich visual information and can be found reproducibly in different images of the same object. Recently, corners have also been used in object recognition (where they are usually referred to as interest points): the regions surrounding interest points contain discriminative information about which object classes are present [17].

FAST (Features from Accelerated Segment Test) is a high quality corner detector implemented using machine learning. It is many times faster than other existing corner detectors. The FAST detector also provides high levels of repeatability under large aspect changes and for different kind of features [25].

SIFT Detector

The Scale-Invariant Feature Transform was first described by David Lowe's in the ICCV 1999 conference paper [11] which also gives some more information on the applications to object recognition. This method "transform an image into a large collection of local feature vectors, each of which is invariant to image translation, scaling, and rotation, and partially invariant to illumination changes and affine or 3D projections"

The first phase identifies key locations in scale space by looking for locations that are maxima or minima of a difference-of-Gaussian function. A feature vector is generated by using each point. The vector describes the local image region sampled relative to its scale-space coordinate frame. The resulting feature vectors are called SIFT keys. In Lowe's implementation, each image generates on the order of 1000 SIFT keys. This process that requires less than 1 second of computation time [11].

2.3.2 Edges

Edges are points in the image where brightness changes particularly sharply are often called. Edge points are associated with the boundaries of objects and other kinds of meaningful changes" [4]. In edge detection the goal is to return a binary image where a non-zero value denotes the presence of an edge in the image. They optionally also return other information such as the orientation and scale associated with the edge [17].

Figure 2.3: Egg image and the detected edges

There are various reasons for being interested in edges. "The contours of potentially scene elements like solid objects, marks on surfaces, and shadows, all generate intensity edges. Moreover, image lines, curves and contours, which are often the basic elements for stereopsis, calibration, motion analysis and recognition, are detected from chains of edges points. Finally, lines drawings are common and suggestive images for humans" [33].

Edge detection in computer vision is typically done in three steps:

- **Noise Smoothing:** Suppress as much of the image noise as possible, without destroying the true edges. In the absence of specific information assume the noise white and Gaussian.
- **Edge Enhancement:** Design a filter responding to edges; that is, the filter's output is a large at edge pixels and low elsewhere, so that edges can be located as the local maxima in the filter's output.
- **Edge Localization:** Decide which local maxima in the filter's output are edges and which are just caused by noise. This involves thinning wide edges to 1-pixel width *(nonmaximum suppression)* and establishing the minimum value to declare a local maxima an edge *(thresholding)*. [33]

Canny Edge Detector

Canny operator was designed to be an optimal edge detector. It takes as input a gray scale image, and produces as output an image showing the positions of tracked intensity discontinuities [24].

"The Canny operator works in a multi-stage process"[24]. The steps followed by the operator are:

1. first of all the image is smoothed by Gaussian convolution,
2. then a simple 2-D first derivative operator is applied to the smoothed image to highlight regions of the image with high first spatial derivatives,
3. edges give rise to ridges in the gradient magnitude image,
4. the algorithm then tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output.

The values of the parameters, the width of the Gaussian kernel used in the smoothing phase, and an upper and a lower threshold used by the tracker, could not be already estimated. They must be obtained according to the nature of the images, that is, during the implementation process, in order to obtain a high accuracy level on the detected borders.

2.4 A Brief Review of Stereo Vision

Stereo vision refers to the ability to infer information on the 3-D structure and distance of a scene from two or more images taken from different viewpoints. From a computational point of view, a stereo system must solve two problems. The first, known as correspondence, consist in determining which item in the left eye corresponds to which item in the right one. The difficult part here is that some parts of the scene are visible by one view only. Therefore, a stereo system must also be able to determine the image parts that should not be matched. The second problem that a stereo system must solve is reconstruction. The disparities between correspondence items of all the image points form the so-called disparity map, which can be displayed as an image. If the geometry of the stereo system is known, the disparity map can be converted to a 3-D map of the viewed scene [33].

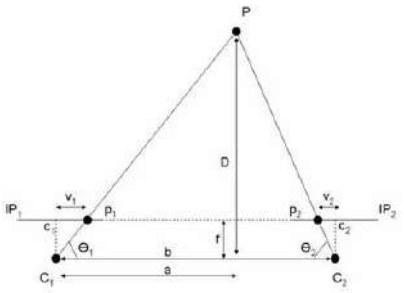

Figure 2.4: Stereo system

The correspondence problem is then view as a search problem: "given an element in the left image, we search for the corresponding elements in the right image" [33]. This requires two decisions:

- which image element to match and,
- which similarity measure to adopt.

The calculations associated with stereo algorithms are often consider simpler when the images of interest have been rectified, i.e., replaced by two projectively equivalent pictures with a common image plane parallel to the baseline joining the two optical centers.

2.5 Delaunay Triangulation

"Let P be a set of points in the plane, and let $\Im$ be a triangulation of P. Then $\Im$ is a Delaunay triangulation of P if and only if the circumcircle of any triangle of $\Im$ does not contain a point of P in its interior"[3].

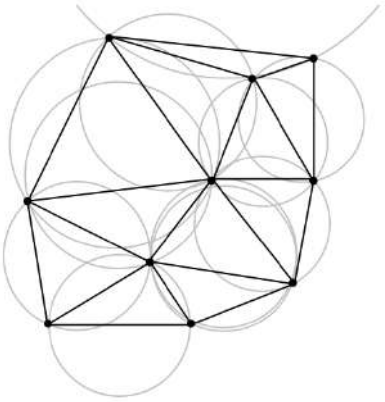

Figure 2.5: Delaunay Circumcircles

The Delaunay triangulation has the following characteristics:

- "Any Delaunay triangulation of P maximizes the minimum angle over all triangulations of P"[3].
- "The expected number of triangles created by the algorithm is at most $9n + 1$ bein n the number of points"[3].

- "The Delaunay triangulation of a ser P of n points in the plane can be computed in $O(n \log n)$ expected time, using $O(n)$ expected storage"[3].
- It is unique.

A constrained Delaunay triangulation is a triangulation with constrained edges which tries to be as much Delaunay as possible. The constrained edges are not necessarily Delaunay edges, therefore the triangles of a constrained Delaunay triangulation do not necessarily satisfy the empty circle property but they fulfill a lighter constrained empty circle property. Therefore, a triangulation is considered constrained Delaunay if and only if the circumscribing circle of any facet encloses no vertex visible from the interior of the facet [18].

Chapter 3
State of the Art

Along the chapter there are summarized some publications about the correspondence problem and the Delaunay triangulation employment. Their advantages and disadvantages are discussed, and it is provided information on computational complexity, practical realizations and applications.

3.1 Stereo Matching

Stereo vision is one of the most important topics on computer vision, therefore there is a lot of literature on this subject. On the current section, are going to presented the most important classification of the algorithms, and then are summarized the most important articles about it.

The first division of the stereo methods is based on the type of output disparity map (Figure 3.1). Dense disparity maps are the most desirable, because in them is determined for all or for almost all the pixels the disparity values. On the contrary, sparse disparity maps have disparity values determined just for selected image points. In most situations they are faster, but have limited applications since there are missing values that has to be interpolated [2].

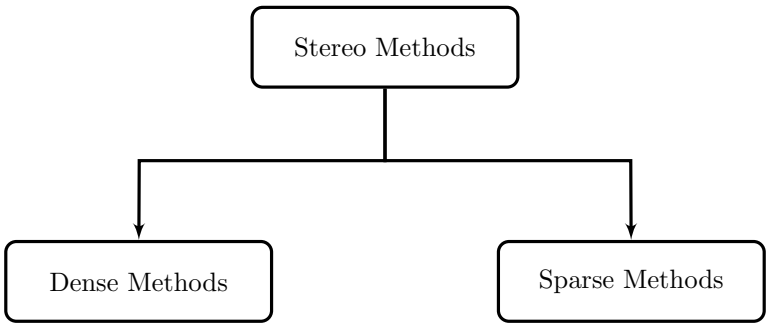

Figure 3.1: Diversity of stereo methods according to the output disparity map

Other possible classification is derived from the format of the signal taken for computation of match values (Figure 3.2). There are methods where it comes directly from the intensity values, whereas others transform intensity into other domains or compute some characteristic features which are then used for matching [2].

A final classification could be determined by dividing the methods into two groups: local and global methods (Figure 3.3). The first ones compute disparity values based only on the local information around certain position of pixels, while global methods use all cost values in the optimization process to determine disparity and occlusions [2].

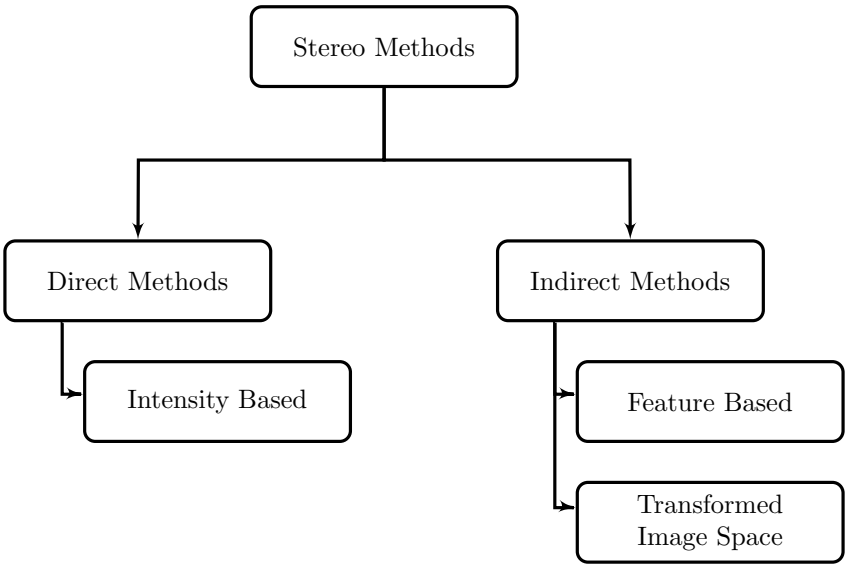

Figure 3.2: Diversity of stereo methods – division into direct methods that are based on bare intensities and indirect ones which operate on transformed space

In the following subsections are presented some of the most characteristic matching methods, as well as publications about their use.

3.1.1 Belief Propagation

The stereo problem is formulated in the probabilistic way in this approach, by means of Markov random fields. The maximum a posteriori estimation is obtained by applying a Bayesian belief propagation (BP) algorithm. BP executes a kind of message passing. This message is meant as a probability that a receiver (a node in MRF) should exhibit disparity which is congruent with all information already passed to it by a sender. The nodes are separated into high-confidence and low-confidence ones. The entropy of a message from high-confidence nodes to low-confidence nodes is smaller than in the opposite direction [2].

One implementation of this approach is presented in [31] where the stereo matching problem is formulated as a Markov network and is solved using Bayesian belief propagation. This approach gives a promising way for ill-posed problems because it treats it as an inference problem or finding the "best guess" solution. For stereo matching, the idea is to infer scene structure S given images I. The output from the Bayesian approach is not only a single solution but also a posterior probability distribution $P(S|I)$. The stereo Markov network consists of three couple Markov random fields that model the following:

- A smooth field for depth/disparity (D), defined on the image lattice of the reference view.
- A line process for depth discontinuity (L), located on the dual of the image lattice and represents explicitly the presence or absence of depth discontinuities in the reference view.
- A binary process for occlusion (O), to indicate occlusion regions in the reference view.

Using Bayes' rule, the joint posterior probability over (D), (L), and (O) given a pair of stereo images $I = I_L, I_R$, where I_L, I_R is the left (reference) and right images, respectively, is:

$$P(D, L, O|I) = \frac{P(I|D, L, O)P(D, L, O)}{P(I)} \tag{3.1}$$

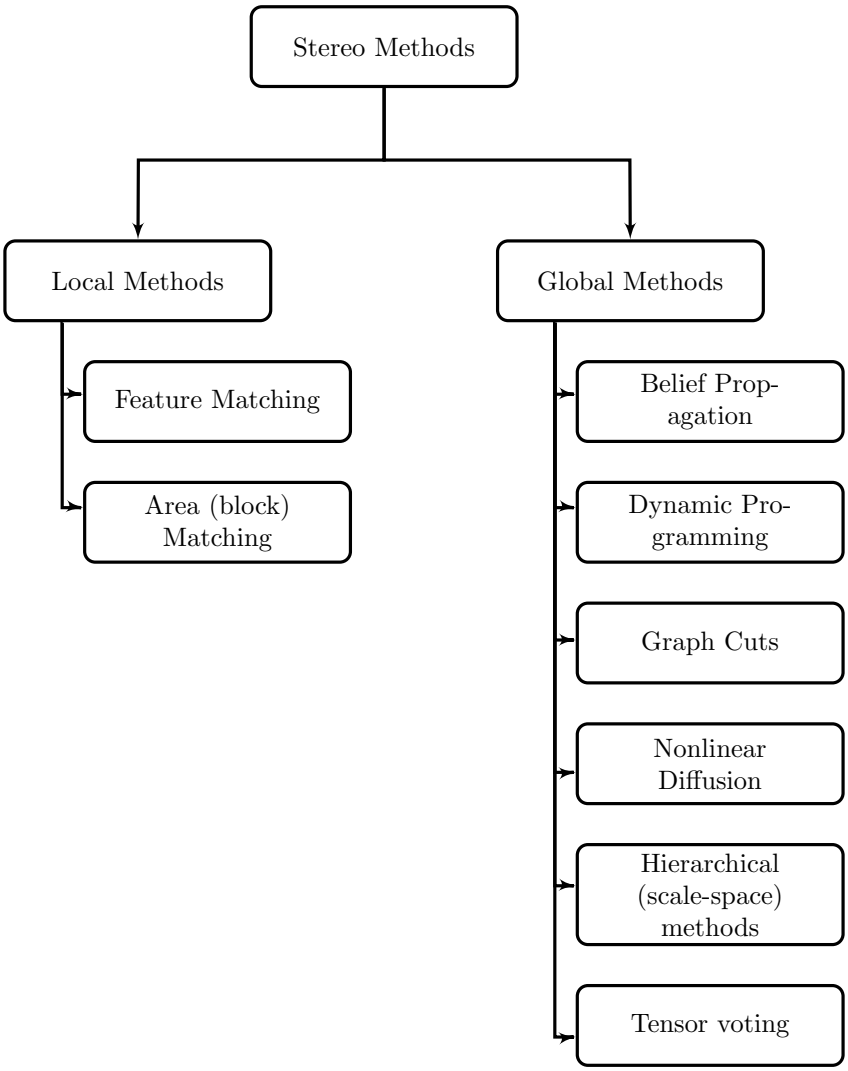

Figure 3.3: Diversity of stereo methods – division into local and global methods

After eliminating the line process and the binary process by introducing two robust functions, the belief propagation algorithm is applied to obtain the maximum aposteriori (MAP) estimation in the Markov network.

There are two contributions in this article. First, stereo matching is formulated using three MRF's and subsequently estimate the optimal solution by a Bayesian Belief Propagation algorithm. Second, it is proposed a probabilistic framework to integrate additional information into the stereo algorithm.

3.1.2 Graph Cut

Methods from this group formulated a solution to the stereo problem as an energy functional. This implies a computation of a maximum flow in graphs [2].

An application of this method can be found in [1]. This paper describes a dense stereo matching algorithm for epipolar rectified images. The method applies colour segmentation on the reference image. The basic assumptions are that disparity varies smoothly inside a segment, while disparity

boundaries coincide with the segment borders. Disparity is modelled inside a segment by a planar equation. Initial disparity segments are clustered to form a set of disparity layers, which are planar surfaces that are likely to occur in the scene. Assignments of segments to disparity layers are then then divided by minimization of a global cost function. This cost function is based on the observation that occlusions cannot be dealt with in the domain of segments.

In the first step, it is applied colour segmentation to the reference image. Given that the basic assumption states that the disparity values inside a colour segment vary smoothly, it is important that a segment does not overlap a disparity discontinuity. It is therefore safer to use oversegmention. The algorithm starts then by computing an initial disparity map using window-based correlation. Once the initial disparity models are known, the objective is to identify those models that represent the dominant depth surfaces of the scene, which are refered as layers.

Knowing the set of disparity layers, the job of the assignment step is to estimate which parts of the images are covered by which layers as well as to identify occlusions. The problem was seen as a labelling problem, that requires a cost function to fulfill its purpose. Therefore a novel cost function is proposed. It is defined on two levels, one representing the segments and the other corresponding to pixels:

$$C(f) = T_{data}(f) + T_{occlusion}(f) + T_{segment}(f) + T_{missmatch}(f) + T_{smoothness}(f) \tag{3.2}$$

The basic idea is that a pixel has to be assigned to the same disparity layer as its segment, but can as well be occluded. The cost function is then effectively minimized via graph-cuts.

The major contribution of the technique to the field of stereo computation lies in that it is shown how region-based matching can be modelled in a graph-cut approach with explicit treatment of occlusions. One important advantage of this algorithm, is that because of the assumptions, the algorithm is capable of handling large untextured regions, estimating precise depth boundaries and propagating disparity information to occluded regions, which are challenging tasks for conventional stereo methods. In the experimental results is shown that the method produces good-quality results, especially in regions of low texture and close to disparity boundaries.

3.1.3 Dynamic Programming

"The main idea of the methods from this group lies in division of the 2D search problem into a series of separate 1D search problems on each pair of epipolar lines" [2].

An example of the use of this method can be found in [32]. The paper proposes a stereo matching algorithm which employs an adaptive multi-directional dynamic programming (DP) scheme using edge orientations. A new energy function is defined in order to consider the discontinuity of disparity and occlusions, which is minimized by the multi-directional DP scheme. Chain codes are introduced to find the accurate edge orientations which provide the DP scheme with optimal multidirectional paths. The proposed algorithm eliminates the streaking problem of conventional DP based algorithms, and estimates more accurate disparity information in boundary areas.

Given an image with size N x M and the disparity range is $0 \sim$ D, exits a 3D (N x M x D) DSI (disparity space image) for dynamic programming schemes. Disparities are usually estimated by finding the shortest path in the DSI. Here was chosen a a path in DSI that has the same direction as edge orientation, and find the shortest path in the DSI using an energy function. It consists of data term, smoothness one for homogeneous area, discontinuity one for edge area, and occlusion one as:

$$E(d_p) = O_p D(p, d_p) + E_s(d_p) + E_d(d_p) + (1 - O_p) E_o(d_p, d_n) \tag{3.3}$$

where O_p is the binary indicator of occlusion. d_p is the disparity of pixel p, d_n are neighboring disparities with respect to edge information.

The edge information used is the boundary area of objects. In this article, it is presented a new scheme to obtain edge orientations using chain codes. Edge pixels are labelled using 8 directional chain codes and classify them into three types such as single, link, and branch. The chain codes helps to find exact starting and ending pixels on the edges.

The Dynamic Programming based stereo matching algorithms have presented better performances than the usual block matching based methods because this one can exploit the correlations of neighboring disparities.

3.1.4 Cooperative Algorithm

In [34] is presented a cooperative stereo algorithm that uses global constrains to find a dense map. Uniqueness and continuity are assumed. In disparity space it is built a three-dimensional array of match values, where each element of the array corresponds to a pixel in the reference image and a disparity, relative to another image. It is constructed an update function of match values for use with real images. This function generates continuous and unique values by diffusing support among neighboring match values and by inhibiting values along similar lines of sight. The initial match values that were obtained by pixel-wise correlation, are used to retain details during each iteration. Occluded areas are identified after the match values have converged.

The cooperative algorithm can be summarized as follows:

1. Prepare a 3D array, (r, c, d): (r, c) for each pixel in the reference image and d for the range of disparity.
2. Set initial match values L_0 using a function of image intensities, such as normalized correlation or squared differences.
3. Iteratively update match values L_n using (5), until the match values converge.
4. For each pixel (r, c), find the element (r, c, d) with the maximum match value.
5. If the maximum match value is higher than a threshold, output the disparity d, otherwise classify it as occluded.

To probe the effectiveness of the algorithm, the authors provide experimental data from many synthetic and real scenes. The resulting disparity maps are smooth and occlusions are detected. The resulting disparity maps calculated by using real stereo images with ground-truth disparities are used for quantitative comparison with other methods. It was also made a comparison with the multi-baseline method and the multi-baseline plus adaptive window method.

3.2 Delaunay Triangulation

Delaunay triangulation has been widely used as a tool for feature correspondence, region corresponding and block matching. It is the main tool for the proposed algorithm, therefore it worth to show the state of the art in utilization to the correspondence problem. Along this section are presented some articles about its applications and the obtained results.

One of the most influential articles for this project is “Estimation of large-amplitude motion and disparity fields: Application to intermediate view reconstruction” [9]. In the paper is described

a method for establishing dense correspondence between two images in a video sequence or in a stereo pair by combining feature matching with Delaunay triangulation. In the proposed approach:

1. feature points are found using a simple intensity corner detector
2. correspondence pairs between two images are found by maximizing cross-correlation over a small window
3. the Delaunay triangulation is applied to the resulting points, and a dense vector field is computed by planar interpolation over Delaunay triangles
4. the Delaunay triangles are divided whenever the displacement vectors within a triangle do not allow good intensity match

The approach has several advantages: it permits to estimate large displacements and subsequently take into account motion/disparity discontinuities and is at the same time very efficient computationally compared to a typical multi-resolution block matching. Initial results are very promising; the method produces high-quality intermediate views for natural (complex) stereoscopic image sequences. However, the results depend on the presence of texture in the images; the method works well in sequences with strong textures.

Chapter 4

Proposed Approach

The flowchart presents the main overview of the proposed algorithm that is going to be described in detail along the chapter. According to the classification presented on section 3.1, the suggested approach could be classified as a sparse, indirect and local method.

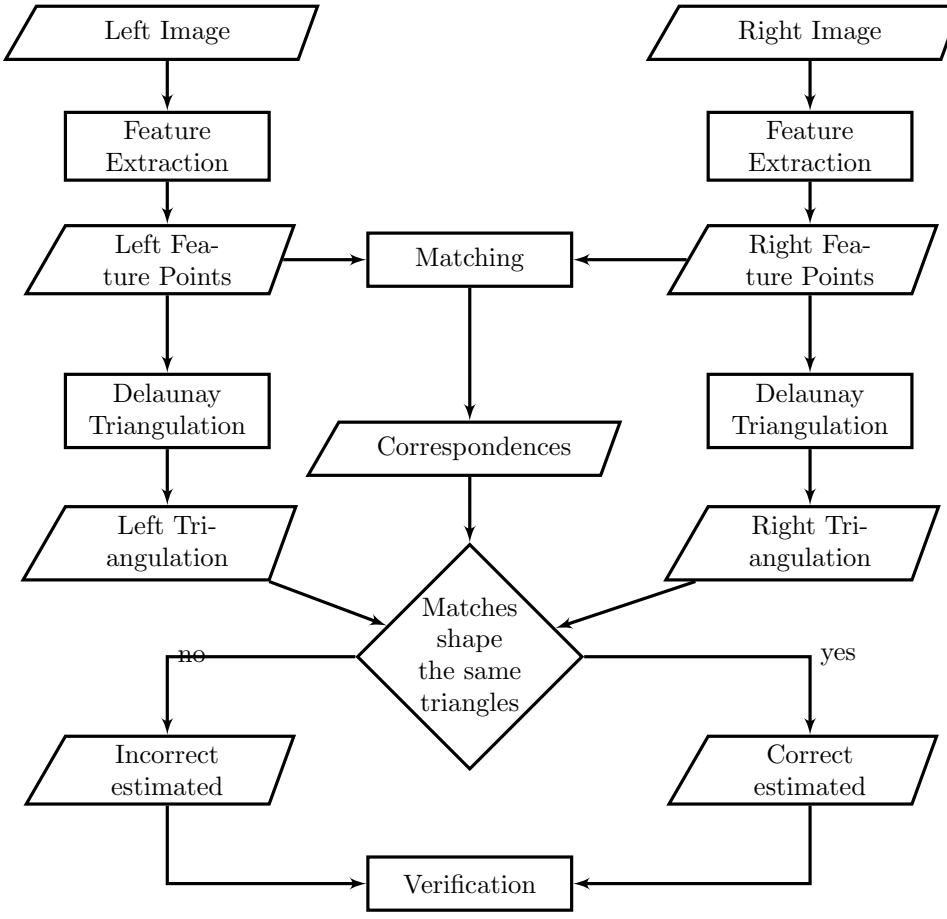

Figure 4.1: Overview of the proposed approach

Figure 4.1 shows a survey of the proposed algorithm. The steps that are going to be presented on the following sections could be summarized as follows:

1. The algorithms input is a couple of images, that represent a left and a right view from the same scene. A detailed description of the input images could be found on the section 4.1.

2. A initial set of feature points is computed and matched using two different approaches: SIFT keypoints and FAST corners. This part of the procedure is described in the sections 4.2 and 4.3.

3. Using the keypoints that were matched, it is built a Delaunay triangulation for each stereo image based. In one of the algorithms approaches are included some constrains, that are explained in detail in section 4.4.

4. Based on the triangulation created on the previous stage, the matches are classified with the criteria illustrated in 4.5.

5. The result is compared with a "gold standard" represented in the ground-truth disparity map as described in 4.6.

The algorithm was implemented in C++ using g++ compiler version 4.4.3. Additionally, it was used the library OpenCV (Open Source Computer Vision) version 2.3.1 for the image processing and Delaunay Triangulation. For the Constrained Delaunay Triangulation algorithm was used the library (Computational Geometry Algorithms Library) version 4.0. The algorithm for corners detection and matching was written in MATLAB version 7.12.

4.1 Stereo Images

The datasets used to test the algorithm were provided from Middlebury Stereo Datasets available on [28]. In total were used 8 of them.

Figure 4.2: Employed datasets

The first dataset, published in [27] consists of high-resolution stereo sequences with complex geometry and pixel-accurate ground-truth disparity data. The ground-truth disparities are acquired "using a novel technique that employs structured lighting and does not require the calibration of the light projectors". Because we are testing a two-view stereo algorithm, the two reference views were used. The images are rectified, which means that all image motion is entirely horizontal. There were used the quarter-size (450 x 375) versions.

The second dataset was published in [26, 8] "Each dataset consists of 7 views taken under three different illuminations and with three different exposures. Disparity maps are provided for views 1 and 5. The images are rectified and radial distortion has been removed". There were used the dataset in third-size (width: 447 or 463, height: 370).

4.2 Feature Points

The proposed approach is classified as a local method that uses feature matching, for this reason, it was necessary to determine a initial set of features that will be matched. Therefore, we opted for the use of two algorithms from the state of the art: SIFT (Scale Invariant Feature Transform) and FAST (Features from Accelerated Segment Test).

4.2.1 Scale Invariant Feature Transform

In order to classify estimated point correspondence by Delaunay triangulation, it is necessary to define a set of initial corresponding points, that are going to be classified. For each one of the both images on the datasets, the resulting keypoints were calculated when applying the SIFT algorithm. Using Lowe's implementation provided in [10], the images presents not only the keypoints, but also the magnitude and the direction of the vector of features. Keypoints are presented with subpixel precision. In Figure 4.3 are presented both teddy images, with the corresponding keypoints.

Figure 4.3: Example of one pair of images after applying the SIFT algorithm.

The most important characteristics from this approach are presented from the author in [12] and could be summarized as follows:

- The features are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint.
- They are well localized in both the spatial and frequency domains, reducing the probability of disruption by occlusion, clutter, or noise.
- The algorithm generates large numbers of features that densely cover the image over the full range of scales and locations.
- A typical image of size 500x500 pixels will give rise to about 2000 stable features (however this number depends on both image content and choices for various parameters)

4.2.2 Features from Accelerated Segment Test

In order to detect corners, it was used the Computer Vision System Toolbox provided by MATLAB. The function *vision.CornerDetector* provides the performance for this task, allowing the change in some parameters. In our particular situation, the detection was implemented by using a method of local intensity comparison: FAST [25]. It detects around 1000 corners using the datasets that were presented already. Figure 4.4 presents the corners detected for the Teddy stereo par.

In [25] the most important characteristics of the corners algorithm are presented. The advantages are that it is many times faster than other existing corners detectors and it has high levels of repeatability under large aspect changes and for different kind of features. However, it is not robust to high level of noise, it can not respond to 1 pixel wide lines at certain angles, when the quantization of the circle misses the line and it is depended on a threshold.

Figure 4.4: Example of one pair of images after applying the FAST algorithm.

4.3 Initial Correspondences

In order to provide a initial set of correspondences, there are used two different algorithms, one for each set of features. For the SIFT algorithm, it is also provide an algorithm for matching SIFT keypoints in [10] that is used to generate the initial set of correspondences in our implementation. For the FAST corners, the correspondences are obtained by using a function provided by MATLAB.

4.3.1 Scale Invariant Feature Transform

According to David Lowe in [12], the matching criteria for the the best candidate match for each keypoint is found by identifying its nearest neighbour in the database of keypoints from training image. The nearest neighbour is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector. Even though, many features from an image will not have any correct match in the training image because they arise from background clutter or were not detected in the training image.

For the Teddy dataset, the resulting image is presented in the Figure 4.6

In [12] are presented the most remarkable features from the matching algorithm:

- The keypoint descriptors are highly distinctive, which allows a single feature to find its correct match with good probability in a large database of features.
- In a cluttered image, many features from the background will not have any correct match in the database, giving rise to many false matches in addition to the correct ones.
- The correct matches can be filtered from the full set of matches by identifying subsets of keypoints that agree on the object and its location, scale, and orientation in the new image. The probability that several features will agree on these parameters by chance is much lower than the probability that any individual feature match will be in error.

Figure 4.5: Example of one pair of images after matching

- Each cluster of 3 or more features that agree on an object and its pose is then subject to further detailed verification. First, a least-squared estimate is made for an affine approximation to the object pose. Any other image features consistent with this pose are identified, and outliers are discarded. Finally, a detailed computation is made of the probability that a particular set of features indicates the presence of an object, given the accuracy of fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence.

4.3.2 Features from Accelerated Segment Test

In order to match the detected FAST corners, it was used a method provided by the *Computer Vision System Toolbox* from MATLAB. First of all, the feature representation from the corners are extracted by using the *extractFeatures* method. The resulting extracted points from both images are matched by using the function *matchFeatures*. The metric specified for this function was SDD (Sum of squared differences) and the match threshold was 0.2. Additionally, given that the input images are rectified, it was checked the condition that the correspondence points have the same x coordinate. The number of matches obtained for each stereo par is variable, but it is around 150 and 300.

Figure 4.6: Example of one pair of images after matching the FAST corners

4.4 Constrains

The proposed approach assumed that image edges could provide information to the Delaunay triangulation indicating the way in which the feature points are related.

The first constrain focus on grouping the points such that triangulation is applied to group of points and not to all of them. This criteria is presented on the Figure 4.7 where the red and green circles are the pixels of interest and the gray region represents the pixels labeled as edges. Considering that there is an edge between both points, the criteria establish that the points must be triangulated in different groups.

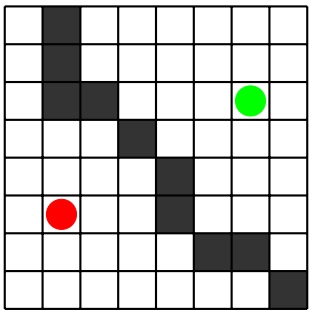

Figure 4.7: Triangles constrain

In order to do that, it process the image, in order to obtain a binary image where the main contours could be identified. First of all a threshold to the image by using the Otsu algorithm was applied. Afterwards it was applied the close and open morphological filters. Figure 4.8 illustrates the resulting images on each stage for the Teddy dataset.

Figure 4.8: (a) Binary image after applying Otsu threshold. (b) Resulting image from the closing operation. (c) Resulting image after the opening filter

The openCV function for contours extraction is finally applied to the resulting image. Because of the big number of unnecessary contours on an image, it was applied a classification to them: contours with more than 1000 points, as well as contours with less than 100 were eliminated. Afterwards, the keypoints are assigned to one contour, in order to triangulate just the points belonging to the same contour. For each keypoint, it is calculated the distance to each contour, by using the function *pointPolygonTest* implemented in OpenCV, which retrieves the minimum Manhattan distance from the point to the contour (a positive value, if the point is inside the contour, and a negative value if is not). The point assigned to the contour with the minimum distance (or to the contour where is inside). Figure 4.9 illustrates the estimated contours and posterior classification by presenting an approximation of each contour by using a bounding box.

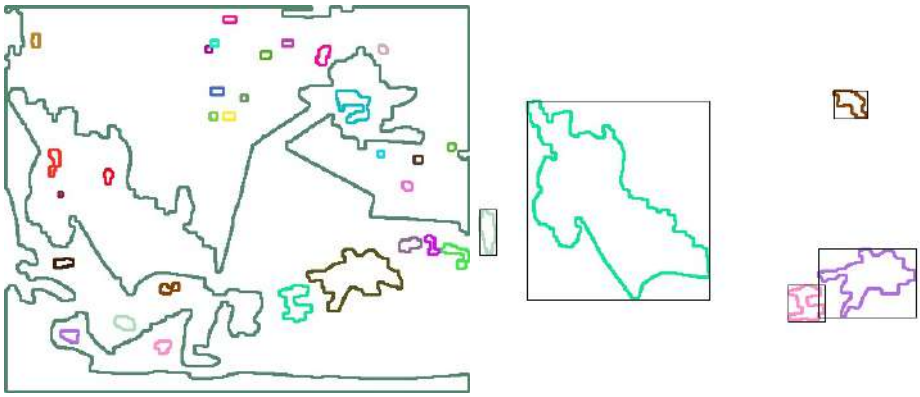

Figure 4.9: Polygons approximation

The second constrain is based on the idea that points belonging to the same contours, must have an edge in the triangulation. Therefore the detailed edges are using in this situation. Figure 4.10 shows this idea by presenting two keypoints, blue and purple, that belong to the same contour. The dark gray pixels represents the image edges and the light gray represents contour pixels, that is, pixels that are not labeled on the image, but that illustrates that edges, despite they are not continuous, belong to the same contour. The orange pixel, even thought it belongs to an edge, it does not belong to the same contour that the blue and purple pixels, therefore, it must be an edge between the blue and the purple pixel but not with the orange one. Given the big number of edges presented on the image, and the number of points located in each contour, it is also required that the points are not separated for more than 5 pixels between each other horizontally as well as vertically.

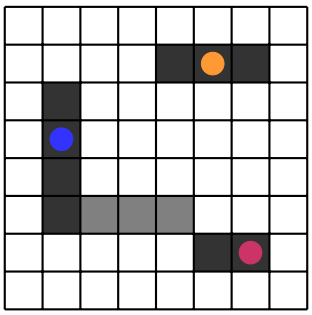

Figure 4.10: Edges constrain

4.5 Classification

The correspondence obtained in the matching stage is classified by using the Delaunay triangulation. It is assumed, than given two correspondences points, they must be "correctly estimated" (i.e. they represents same feature point in both images) if and only if they shape the same triangle in both stereo images, implying that they have the same adjacent vertices in both images. We assume that two vertices are considered the same, if they belong to the initial correspondence points set. The matches that do not satisfy this condition, are label as "incorrectly estimated".

Considering a triangulation as a undirected graph, we say that correspondences points are considered "correctly estimated" if and only if their graphs are isomorphic.

Figure 4.11: Example of one pair of stereo images after apply the Delaunay Triangulation

Definition 1 *Being G and G', with a set of vertices and edges (V, A) and (V', A') respectively. We say that a bijective function ϕ: $V \to V'$ is a graph isomorphism if:*

$$w, v \in A \Leftrightarrow \phi(v), \phi(w) \in A' \tag{4.1}$$

That is, if ϕ preserves the adjacency between vertices [5].

In our algorithm, the bijective function ϕ is represented by the initial map of correspondences.

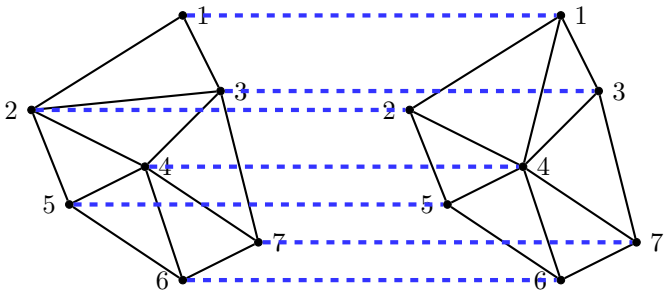

Figure 4.12: Triangulation subset

Figure 4.11 presents a sample triangulation after applying Delaunay Triangulation. It is important to emphasize that the triangulation was done in a first instance with all the points provided by the SIFT algorithm, however those points added noise to the triangulation, so it was necessary to discard them. The triangulation was then done only with the matching points, obtaining better results.

The subset of triangles highlighted with red are going to be used as a sample to explain in detail the classification criteria. Figure 4.12 represents the graphs from left and right triangulations respectively. Vertices are labelled with the numbers 1 to 7 and the one with the same number represents a correspondence, which is also illustrated with a dashed line.

Lets consider the vertex number 5, its correspondences represented in the Figure 4.13 and its adjacent vertex 2, 4 and 6. According to the evaluation criteria, the matching between the vertex number 5 would be labeled as "correctly estimated", given that they have exactly the same adjacent vertex on the left and right triangulations.

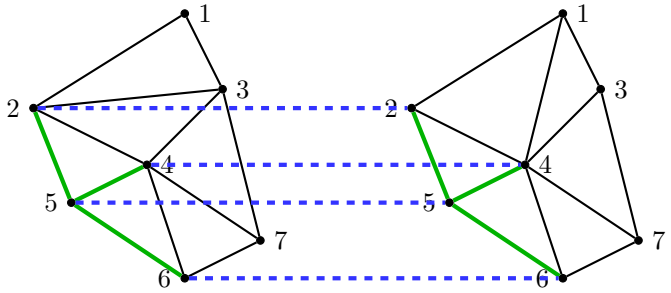

Figure 4.13: Classification criteria for the vertex number five

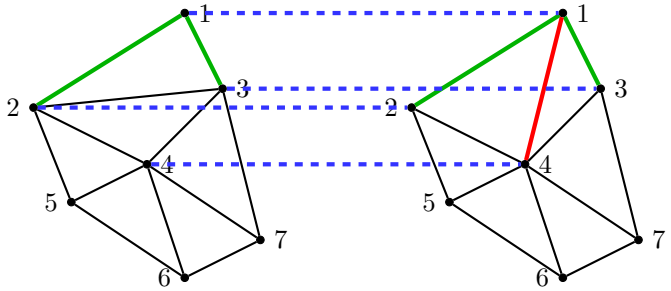

Figure 4.14: Classification criteria for the vertex number one

Now consider the vertex number 1, its correspondences represented on the Figure 4.14 and its adjacent vertices 2 and 3. In the triangulation of the right stereo image, the correspondence vertex has three adjacent vertices: 2, 3 and 4. Even though that they have two common vertices (2 and 3), the matching is labeled as "incorrectly estimated" because the vertices are not exactly the same on both images, therefore they do not shape the same triangles.

The last case that could be presented is illustrated by Figure 4.15 where the initial correspondences has been changed. In this situation, the vertex number 7 is going to be considered, which has edges with the vertex number 3, 4 and 6. In the right triangulation, the corresponding vertex has also three adjacent edges. However, the vertex number 3 on the left stereo image is not an adjacent vertex on the right image, because in its position is vertex number 2 and they are not correspondent. Therefore, the matching between the vertices number 7 is labeled as "incorrectly estimated".

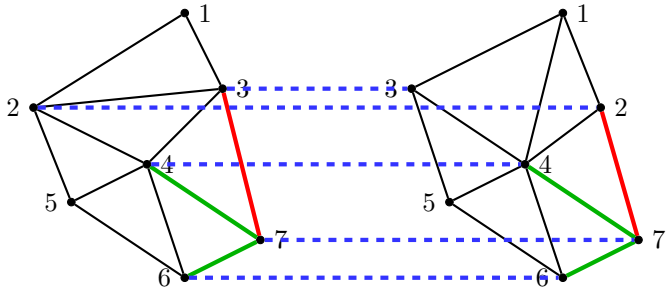

Figure 4.15: Classification criteria for the vertex number seven

The explained criteria is used for the constrained classification as well. The only difference between both triangulations is that for unconstrained triangulation the triangulations are subdivided

by groups, however it does not affect the proposed evaluation. The kind of triangulation obtained is illustrated on the Figure .

Figure 4.16: Triangulation obtained for the teddy stereo par after applying the first constrain

4.6 Verification

It is necessary to compare the obtained results with a "gold standard" in order to evaluate its accuracy, therefore it is used the ground-truth of the image for this purpose. In combination with the stereo images, each dataset provides the true disparity map.

In the dataset published in [27] "Disparities are encoded using a scale factor 4 for gray levels 1 .. 255, while gray level 0 means unknown disparity. Therefore, the encoded disparity range is 0.25 .. 63.75 pixels" [29]. In the same way, in the dataset published in [26, 8] " In the third-size versions, the intensity values of the disparity maps need to be divided by 3" [30]. Figure 4.17 shows an example of the ground-truth disparity map from the Teddy stereo par.

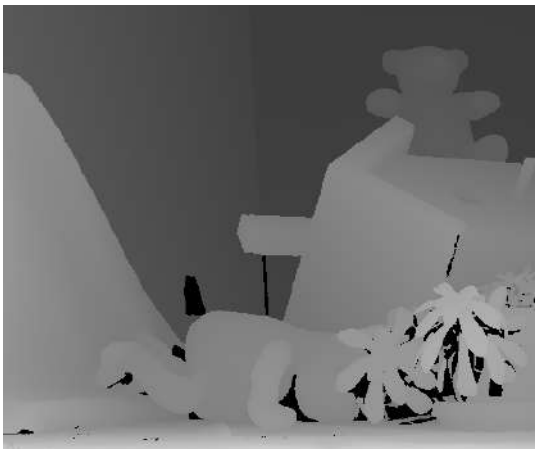

Figure 4.17: Ground Truth image for the Teddy stereo par

For each correspondence calculated in the initial matching, it is calculated the disparity value represented by the absolute difference of both x coordinates. This value, is compared with the

ground-truth by using the absolute error equation. The error for the element (i, j) is calculated using the following formula:

$$error_{i,j} = |groundtruth_{i,j} - \Delta x_{i,j}| \tag{4.2}$$

$groundtruth_{i,j}$ represents the value obtained from the ground-truth disparity map and $\Delta x_{i,j}$ is the absolute difference between the x coordinate from both correspondence points. For error value bigger than **1**, the correspondence is labeled as "bad match", otherwise is considered a "good match".

4.7 Results

After testing the proposed algorithm with the datasets, it was made some analysis about the quality of the obtained results as well as the algorithm behaviour for the different inputs. A "confusion matrix" is generated for both features. The algorithm classifies the correspondence as "incorrectly estimated" and "correctly estimated" according to the criteria explained in 4.5, meanwhile the comparison with the ground-truth arrange them as "good match" or "bad match". The indicators from this matrix represents the following information:

- **True positives (tp):** The algorithm classifies the match as "incorrectly estimated", and the comparison with the ground-truth classifies it as "good match".
- **False positives (fp):** The algorithm classifies the match as "incorrectly estimated", but the comparison with the ground-truth classifies it as a "good match".
- **False negatives (fn):** The algorithm classifies the match as "correctly estimated", but the comparison with the ground-truth classifies it as a "bad match".
- **True negatives (tn):** The algorithm classifies the match as "correctly estimated", and the comparison with the ground-truth classifies it as a "bad match".

Based on those indicators, it is calculated the following. In a perfect algorithm performance, all of this values would be 100%:

- **Sensitivity:** Is the probability that a matching is a "incorrectly estimated", given that is a "bad match".
- **Specificity:** Is the probability that a matching is a "correctly estimated", given that is a "good match".
- **Positive Predictive Value (PPV):** Percentage of "incorrectly estimated" that were identified well, from the total of "incorrectly estimated".
- **Negative Predictive Value (NPV):** Percentage of "correctly estimated" that were identified well, from the total of "correctly estimated".

4.7.1 Scale Invariant Feature Transform

The SIFT algorithm, in agreement with the information provided by the author, generates between 500 and 1000 keypoints for each image of the stereo par, and from those points, around 200 are matched. The percentage of "bad matches" in the initial set of correspondences varies between 12% and 36%. The less number of "bad matches" was obtained by the dataset **books**, while the worst one by the dataset **teddy**. The information of the algorithm is presented on the Table 4.1.

After applying the proposed algorithm with no constrains to the points generated with SIFT, Tables 4.2 and A.1 were created (The Tables with the first group of indicators can be found on the Appendix A). There is presented the information in a "confusion matrix". The most remarkable

Table 4.1: SIFT algorithm performance

Dataset	Points Left	Points Right	Matches	Bad Matches	% Bad Matches
Art	1167	1089	205	33	16,10
Books	823	904	401	51	12,72
Cones	1163	1147	467	61	13,06
Dolls	1462	1406	561	115	20,50
Laundry	1123	1124	263	95	36,12
Moebius	867	920	318	56	17,61
Reindeer	592	553	244	37	15,16
Teddy	844	881	328	133	40,55

result is the sensitivity of the dataset **art** because is the highest one. However, the value is just 80%, which is still low for the algorithm performance. In general, the unconstrained triangulation presents in this set of matching a poor result, because the values of sensitivity are not higher than 80%, which means, that there are many "bad matches" that are not considered "incorrectly estimateds". On the other hand, the specificity value, even though is not so high, presents an acceptable value, given that the existence of "good matches" that are consider "incorrectly estimateds" is not so critical in our algorithm.

Table 4.2: Unconstrained classification Confusion matrix I

Dataset	Sensitivity(%)	Specificity(%)	PPV(%)	NPV(%)
Art	81,82	51,16	24,32	93,62
Books	68,63	70,86	25,55	93,94
Cones	50,82	63,30	17,22	89,55
Dolls	62,61	54,26	26,09	84,91
Laundry	70,53	52,38	45,58	75,86
Moebius	51,79	54,96	19,73	84,21
Reindeer	64,86	41,55	16,55	86,87
Teddy	57,63	55,02	21,94	85,55

The first constrain was added to the triangulation, in order to evaluate its pertinence. Afterwards a new "confusion matrix" was built, which is presented in the Tables 4.3 and A.2. In general, the sensitivity decreased for most datasets while the specificity increased. The reason, is that there are some edges "missing" in comparison with the unconstrained triangulation, and some of them were not supposed to be there, decreasing the number of false positives (which increases the specificity) and some had to be there, increasing the number of false negatives (which decreases the sensitivity).

The second constrain was applied to the triangulation and the results are presented on the Tables 4.4 and A.3. The constrain presents a really bad performance for half of the datasets, because not only the sensitivity but also the specificity. However, for the others improves the sensitivity in comparison with the previous versions of the algorithm, decreasing the specificity at the same time.

The algorithm, not only in the unconstrained version, but also in the constrained one, tends to present a huge number of false positives. The reason, is that one pair labelled as a "incorrectly estimateds" makes that the pairs around them (the adjacent vertices in the triangulation graph) are also labelled the same because of the missing edges. It means, that there is an error propagation, which decrease the specificity. In order to improve this, it was proposed another evaluation method for both algorithms, which is more flexible.

Table 4.3: Triangles constrained classification Confusion matrix I

Dataset	**Sensitivity(%)**	**Specificity(%)**	**PPV(%)**	**NPV(%)**
Art	75,76	55,23	24,51	92,23
Books	68,63	70,86	25,55	93,94
Cones	42,62	71,67	18,44	89,26
Dolls	58,26	60,54	27,57	84,91
Laundry	64,21	54,76	44,53	73,02
Moebius	46,43	62,21	20,80	84,46
Reindeer	49,00	40,91	79,03	15,00
Teddy	64,41	55,76	24,20	87,72

Table 4.4: Edges constrained classification Confusion matrix I

Dataset	**Sensitivity(%)**	**Specificity(%)**	**PPV(%)**	**NPV(%)**
Art	60,61	29,65	14,18	79,69
Books	41,18	42,00	9,38	83,05
Cones	32,79	39,90	7,58	79,80
Dolls	76,52	10,31	18,03	63,01
Laundry	87,37	20,24	38,25	73,91
Moebius	67,86	27,48	16,67	80,00
Reindeer	67,57	18,84	12,95	76,47
Teddy	67,80	19,70	15,63	73,61

First of all, the proposed evaluation was that, being n the number of edges from a matched point, evaluate that the correspondent points have $n-1$ equal vertices, if and only if $n > 3$. However, this evaluation decreased so much the sensitivity value for the Teddy dataset, where it was initially tested. Therefore, the value of n was increased, until the sensitivity and specificity values were acceptable. The final condition was that $n > 8$ and it was tested for all the datasets. This results are presented on the Tables 4.5 and A.4. In general, the specificity value increased, but the sensitivity value decreased considerably for some datasets like **art**.

4.7.2 Features from Accelerated Segment Test

The FAST algorithm provides nearly 1000 keypoints on each stereo par. However, only a small portion of those points, around 300 is matched. The percentages of the bad matches is higher in comparison with the SIFT algorithm, and it is remarkable for the datasets **books** and **laundry** that have 49,17% and 66,21%. The complete information is summarized in the Table 4.6.

The results after the application of the unconstrained triangulation are presented on the Tables 4.7 and B.1 (The Tables with the first group of indicators can be found on the Appendix B) . The values of sensitivity are higher than the ones for the SIFT points. However, the specificity is at the same time lower than with the other set of keypoints. For the datasets **books** and **laundry** is specially high the sensitivity value: 93,29% and 97,42% respectively. It could be inferred then, that the algorithm has a better performance, when the initial set of correspondences has around 50% of "bad matches".

Afterwards, the first constrain was applied, and the results were similar to the case of the SIFT points. A table with the results could be find in 4.8 and B.2. While the specificity value incremented for most datasets (for books there was no change), the sensitivity decremented for some of them, and for others keep equal.

The second constrain was applied, and as a result, a large number of matches were labelled

Table 4.5: Unconstrained classification with a "relaxed condition" Confusion matrix I

Dataset	Sensitivity(%)	Specificity(%)	PPV(%)	NPV(%)
Art	63,64	55,81	21,65	88,89
Books	58,82	76,29	26,55	92,71
Cones	45,90	70,69	19,05	89,69
Dolls	57,39	58,74	26,40	84,24
Laundry	63,16	57,14	45,45	73,28
Moebius	50,00	60,31	21,21	84,95
Reindeer	59,46	47,83	16,92	86,84
Teddy	52,54	62,08	23,31	85,64

Table 4.6: FAST algorithm performance

Dataset	Points Left	Points Right	Matches	Bad Matches	% Bad Matches
Art	985	986	164	72	43,90
Books	958	955	303	149	**49,17**
Cones	947	946	268	78	29,10
Dolls	950	948	280	85	30,36
Laundry	960	966	293	194	**66,21**
Moebius	963	970	213	63	29,58
Reindeer	948	955	201	58	28,86
Teddy	950	947	316	123	38,92

as "incorrectly estimated", it implies, that the sensitivity increased for most of the datasets, but at the same time the specificity decrease to values around 10%. Tables 4.9 and B.3 presents the obtained results.

The new evaluation criteria explained in the last section was also tested in the FAST algorithm. For this kind of features the results were better than for the SIFT feature points. For the half of the datasets the sensitivity keeps equal, meanwhile the specificity raised around two or three percent. Tables 4.10 and B.4 presents the results.

4.8 SIFT vs FAST

In the process of the initial set of points selection, the first choice was to use the Sift Invariant Feature Transform, as a set of beginning feature points. However, after testing the proposed algorithm, the result was not the expected, even tough many datasets and constrains were used. Therefore, it was decided to change the initial set of features for FAST corners. As was presented in the last section, the result got better.

The SIFT feature points have been used in the literature in many applications, providing excellent results, and for this reason it was the first choice considered for the proposed algorithm. However, the points generated by it, generates points that are inside the objects and not in the edges, like FAST corners do. For the triangulation building, it produces vertex triangulated with "interior" feature points, but not with remarkable features like "corners", which are more relevant in the stereo correspondences. Therefore, a big number of false positives is generated, decreasing the sensitivity and hence the algorithm performance.

Table 4.7: Unconstrained classification Confusion matrix I

Dataset	Sensitivity(%)	Specificity(%)	PPV(%)	NPV(%)
Art	70,83	25,00	42,50	52,27
Books	**93,29**	23,38	54,09	78,26
Cones	70,51	46,32	35,03	79,28
Dolls	80,00	44,62	38,64	83,65
Laundry	**97,42**	25,25	71,86	83,33
Moebius	84,13	43,33	38,41	86,67
Reindeer	87,93	27,97	33,12	85,11
Teddy	78,86	34,20	43,30	71,74

Table 4.8: Triangles constrained classification Confusion matrix I

Dataset	Sensitivity(%)	Specificity(%)	PPV(%)	NPV(%)
Art	63,89	31,52	42,20	52,73
Books	93,29	23,38	54,09	78,26
Cones	71,79	51,05	37,58	81,51
Dolls	67,06	62,05	43,51	81,21
Laundry	95,88	39,39	75,61	82,98
Moebius	68,25	49,33	36,13	78,72
Reindeer	77,59	35,66	32,85	79,69
Teddy	78,86	36,79	44,29	73,20

4.9 Triangulation

Independent of the kind of input set of initial feature points, the quality of the classification was tested. 500 points were randomly chosen from the left image of the teddy dataset and based on the ground-truth disparity map, the right corresponding points were chosen. The results are presented on the Tables 4.11 and C.1. It is remarkable, that according to the analysis presented in the last sections, the triangulation does not present good performance, when the number of initial "bad matches" is small.

According to the results obtained in the previous section, it was assumed that the performance of the algorithm is related to the initial set of chosen points. One important reason for this assumption is that Delaunay Triangulation depends from the distance of a vertex to its neighbours. Therefore, it was calculated the distance from each vertex to its adjacent vertices, and the results are presented in the Tables 4.12 and 4.13.

In Table 4.12 can be observed that the distances between the points and the vertices that are labelled as "Incorrect classified" are always larger than distances between the one labelled as "Correctly estimated". A similar behaviour is presented for the FAST corners, however, the values are very close between each other for the datasets where the best algorithm performance is achieved. A new hypothesis was referred, based on this fact. The factor that could influence this change is the deformation suffered from the objects given by the position of the cameras. It causes that points that are visually correspondent, are labeled as "incorrectly estimated".

4.10 Algorithm Performance

The performance of the algorithm according to its runtime was evaluated. The program was executed on a machine with the processor AMD Athlon(tm) X2 Dual-Core QL-65 (2.1GHz, 1MB L2 Cache) and 3GB of RAM memory. The time measurement was done by using the command **time**

Table 4.9: Edges constrained classification Confusion matrix I

Dataset	**Sensitivity(%)**	**Specificity(%)**	**PPV(%)**	**NPV(%)**
Art	90,28	9,78	43,92	56,25
Books	91,28	13,64	50,56	61,76
Cones	82,05	11,58	27,59	61,11
Dolls	92,94	9,74	30,98	76,00
Laundry	93,81	24,24	70,82	66,67
Moebius	88,89	4,67	28,14	50,00
Reindeer	87,93	15,38	29,65	75,86
Teddy	83,74	11,92	37,73	53,49

Table 4.10: Constrained classification with a "relaxed condition" Confusion matrix I

Dataset	**Sensitivity(%)**	**Specificity(%)**	**PPV(%)**	**NPV(%)**
Art	69,44	28,26	43,10	54,17
Books	92,62	24,68	54,33	77,55
Cones	70,5	50,53	36,91	80,67
Dolls	77,65	47,18	39,05	82,88
Laundry	97,42	27,27	72,41	84,38
Moebius	82,54	46,00	39,10	86,25
Reindeer	87,93	31,47	34,23	86,54
Teddy	78,86	38,34	44,91	74,00

from Ubuntu version 10.04. Results are presented on the Table 4.14 and 4.15, where the runtime is expressed in minutes (m) and seconds (s).

Table 4.11: Triangles constrained classification Confusion matrix I

Dataset	Sensitivity(%)	Specificity(%)	PPV(%)	NPV(%)
Art	-	52,00	0,00	100,00
Books	0,00	52,10	0,00	99,58
Cones	50,00	52,01	0,38	99,58
Dolls	66,67	52,72	0,76	99,58
Laundry	75,00	53,63	1,12	99,57
Moebius	60,00	53,74	1,12	99,13
Reindeer	50,00	52,83	1,14	98,73
Teddy	71,43	53,75	1,85	99,13

Table 4.12: Distance Vertices SIFT points

Dataset	Incorrect Left	Incorrect Right	Correct Left	Correct Right
Art	41,5014	40,5149	27,1347	27,4006
Books	36.5893	35.1679	19.0444	19.0759
Cones	32,396	32.0308	23.3873	23.2421
Dolls	25.9198	25.6324	22.6723	22.5702
Laundry	36.4967	37.7258	25.1747	25.8179
Moebius	35.8318	35.1869	23.7902	23.7454
Reindeer	34.4418	34.6294	23.8409	23.9347
Teddy	35,0916	36,369	25,316	25,4032

Table 4.13: Distance Vertices FAST

Dataset	Incorrect Left	Incorrect Right	Correct Left	Correct Right
Art	45.9654	45.8604	33.3795	33.0938
Books	**27.8345**	**27.5849**	**37.7912**	**37.5622**
Cones	34.2913	35.0061	31.9205	31.8214
Dolls	36.0264	37.5633	31.0209	30.9548
Laundry	**30.4673**	**29.6956**	**36.8481**	**35.6139**
Moebius	45.3322	38.7913	31.9513	31.9211
Reindeer	**37.0303**	**35.7755**	**36.3115**	**35.6**
Teddy	31,043	30.2301	27,998	27.6946

Table 4.14: Program runtime with the SIFT points

Dataset	Unconstrained	Triangles Constrain	Edges Constrain
Art	0m3.522s	0m0.703s	0m30.183s
Books	0m22.649s	0m24.073s	1m34.225s
Cones	0m38.740s	0m2.529s	2m41.956s
Dolls	1m9.766s	0m2.835s	4m38.767s
Laundry	0m7.223s	0m0.591s	0m49.141s
Moebius	0m11.806s	0m0.653s	1m6.720s
Reindeer	0m5.610s	0m0.728s	0m24.481s
Teddy	0m12.565s	0m2.242s	1m10.704s

Table 4.15: Program runtime with FAST

Dataset	**Unconstrained**	**Triangles Constrain**	**Edges Constrain**
Art	0m1.970s	0m0.523s	0m18.123s
Books	0m11.913s	0m10.655s	0m54.179s
Cones	0m8.151s	0m8.234s	0m53.973s
Dolls	0m9.181s	0m0.546s	1m7.226s
Laundry	0m9.850s	0m0.671s	0m59.391s
Moebius	0m4.663s	0m0.469s	0m31.748s
Reindeer	0m4.027s	0m0.550s	0m17.340s
Teddy	0m3.894s	0m1.470s	1m6.373s

Chapter 5
Conclusions

5.1 Conclusions

- The use of all the recognized keypoints for the triangulation, adds noise to it, incrementing the number of matches that are incorrectly classified. Therefore, the analysis was done by only triangulating the matches points.
- The classification algorithm presents better performance for the FAST corners than for the SIFT keypoints, in terms of sensitivity, which is for our analysis the most representative measurement of quality. On the other hand, the specificity is higher for the SIFT algorithm, given that less points are recognizing as "bad matches", and therefore, according to the evaluation criteria, the number of false positives decrease. It means, that the more "bad matches" that are correctly classified, the more "good matches" that are incorrectly classified, because of the evaluation criteria used, where one bad match affects directly the neighbours points.
- The classification algorithm exhibit the best performance in the FAST corners feature points, where the percentage of initial bad matches is bigger than approximately 50%. It means, that the algorithm presents high sensitivity (between 93% and 97%) when the initial matching is bad.
- The use of constrains contribute to the increment of specificity, given that not all the matching points are triangulated together, reducing the number of false positives. However, it also decrements the sensitivity, because for some points that are "bad matches", there are missing edges, generating a false negative.
- The relaxed evaluation condition presented worked better for the FAST corners features, because of the locations of the points. For the SIFT algorithm, there are many internal points, for this reason, the new evaluation criteria only decrease the sensitivity. In the FAST corners situation, the criteria presents the same sensitivity and higher specificity for the half of the datasets, while for the others the specificity increase, but the sensitivity decrease.

5.2 Future Research

The project opens the possibility to continue the research about the use of the Delaunay Triangulation as a classification tool for a set of correspondence points. The future research could be then be oriented to:

- Propose a new kind of constrains, different that edges, like colour, texture, etc. that contribute to create a triangulation, that improve the classification quality, increasing the number of classified as "incorrect bad matches and decreasing the number of classified as "correct" good matches.

- Identify a new evaluation criteria, that increase the specificity, but at the same time the sensitivity, improving the classification performance.

Bibliography

[1] M. Bleyer and M. Gelautz. Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions. *Signal Processing: Image Communication*, 22(2):127–143, 2007.

[2] J. Paul Siebert Boguslaw Cyganek. *An Introduction to 3D Computer Vision Techniques and Algorithms*, volume 10. Wiley, 1 edition, February 2009.

[3] M. De Berg, O. Cheong, and M. Van Kreveld. *Computational geometry: algorithms and applications*. Springer-Verlag New York Inc, 2008.

[4] D.A. Forsyth and J. Ponce. *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 1 edition, August 2002.

[5] Pablo Fernández Gallardo. Notas de matemática discreta. `http://www.uam.es/personal_pdi/ciencias/gallardo/capitulo8a.pdf`, March 2010.

[6] ©2012 Google. German scientists unveil self-driving car. `http://www.google.com/hostednews/afp/article/ALeqM5izCRPtPhsc2k6kCY2_-YOeUs7r_w?docId=CNG.c668be9320e3376a10d767deb0b0649f.a91`, October 2010. Visited June 2012.

[7] John Y. Goulermas. *Evolutionary Techniques for the Stereo Correspondence Problem*. PhD thesis, University of Manchester, 2000.

[8] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. pages 1–8, 2007.

[9] M. Kardouchi, J. Konrad, and C. Vazquez. Estimation of large-amplitude motion and disparity fields: Application to intermediate view reconstruction. *Proc. SPIE Visual Communications and Image Process*, 4310:340–351, 2001.

[10] David Lowe. Demo software: Sift keypoint detector. `http://www.cs.ubc.ca/~lowe/keypoints/siftDemoV4.zip`, July 2005. Visited on May 2012.

[11] D.G. Lowe. Object recognition from local scale-invariant features. *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2(60):1150–1157, 1999.

[12] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[13] J. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. *Journal of the American Statistical Association*, pages 76–89, 1987.

[14] David Marshall. 2d image input. `http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MARSHALL/node5.html`, 1997. Visited May 2012.

[15] ©2004-2012 Koninklijke Philips Electronics N.V. Syntegra. `http://www.healthcare.philips.com/in_en/products/ros/products/syntegra/`, June 2012. Visited June 2012.

[16] Jet Propulsion Laboratory. California Institute of Technology. Mars exploration rovers. http://marsrovers.jpl.nasa.gov/home/index.html, June 2012. Visited June 2012.

[17] S.J.D. Prince. *Computer vision: models, learning, and inference*. Cambridge University Press, 2011.

[18] CGAL Open Source Project. 2d triangulations. http://www.cgal.org/Manual/latest/doc_html/cgal_manual/Triangulation_2/Chapter_main.html, March 2012. Visited June 2012.

[19] A. Walker R. Fisher, S. Perkins and E. Wolfart. Closing. http://homepages.inf.ed.ac.uk/rbf/HIPR2/close.htm, 2003. Visited June 2012.

[20] A. Walker R. Fisher, S. Perkins and E. Wolfart. Dilation. http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm, 2003. Visited June 2012.

[21] A. Walker R. Fisher, S. Perkins and E. Wolfart. Erosion. http://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm, 2003. Visited June 2012.

[22] A. Walker R. Fisher, S. Perkins and E. Wolfart. Opening. http://homepages.inf.ed.ac.uk/rbf/HIPR2/open.htm, 2003. Visited June 2012.

[23] A. Walker R. Fisher, S. Perkins and E. Wolfart. Thresholding. http://homepages.inf.ed.ac.uk/rbf/HIPR2/threshld.htm, 2003. Visited June 2012.

[24] Ashley Walker Robert Fisher, Simon Perkins and Erik Wolfart. Canny edge detector. http://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm, 2003. Visited November 2011.

[25] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. *Computer Vision–ECCV 2006*, pages 430–443, 2006.

[26] D. Scharstein and C. Pal. Learning conditional random fields for stereo. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1–8, 2007.

[27] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings*, 1:I–195, 2003.

[28] Daniel Scharstein. Middlebury stereo datasets. http://vision.middlebury.edu/stereo/data/. Visited October, 2011.

[29] Daniel Scharstein. 2003 stereo datasets with ground truth. http://vision.middlebury.edu/stereo/data/scenes2003/, November 2007. Visited June 2012.

[30] Daniel Scharstein. 2005 stereo datasets with ground truth. http://vision.middlebury.edu/stereo/data/scenes2005/, July 2007. Visited June 2012.

[31] J. Sun, N.N. Zheng, and H.Y. Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 787–800, 2003.

[32] M.C. Sung, S.H. Lee, and N.I. Cho. Stereo matching using multi-directional dynamic programming and edge orientations. *IEEE International Conference on Image Processing, ICIP*, 1:I–233, 2007.

[33] E. Trucco and A. Verri. *Introductory techniques for 3-D computer vision*, volume 93. Prentice Hall, 1998.

[34] C.L. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):675–684, 2000.

Appendices

Appendix A

Scale Invariant Feature Transform

A.1 Unconstrained Classification

Table A.1: Unconstrained classification Confusion matrix II

Dataset	Bad Matches	Good Matches	tp	fp	fn	tn
Art	111	94	27	84	6	88
Books	137	264	35	102	16	248
Cones	180	287	31	149	30	257
Dolls	276	285	72	204	43	242
Laundry	147	116	67	80	28	88
Moebius	147	171	29	118	27	144
Reindeer	145	99	24	121	13	86
Teddy	155	173	34	121	25	148

A.2 Constrained Classification: Triangles

Table A.2: Triangles constrained classification Confusion matrix II

Dataset	Bad Matches	Good Matches	tp	fp	fn	tn
Art	102	103	25	77	8	95
Books	137	264	35	102	16	248
Cones	141	326	26	115	35	291
Dolls	243	318	67	176	48	270
Laundry	137	126	61	76	34	92
Moebius	125	193	26	99	30	163
Reindeer	124	120	98	26	102	18
Teddy	157	171	38	119	21	150

A.3 Constrained Classification: Edges

Table A.3: Edges constrained classification Confusion matrix II

Dataset	Bad Matches	Good Matches	tp	fp	fn	tn
Art	141	64	20	121	13	51
Books	224	177	21	203	30	147
Cones	264	203	20	244	41	162
Dolls	488	73	88	400	27	46
Laundry	217	46	83	134	12	34
Moebius	228	90	38	190	18	72
Reindeer	193	51	25	168	12	39
Teddy	256	72	40	216	19	53

A.4 Relaxed Evaluation Condition

Table A.4: Constrained classification with a "relaxed condition" Confusion matrix II

Dataset	Bad Matches	Good Matches	tp	fp	fn	tn
Art	97	108	21	76	12	96
Books	113	288	30	83	21	267
Cones	147	320	28	119	33	287
Dolls	250	311	66	184	49	262
Laundry	132	131	60	72	35	96
Moebius	132	186	28	104	28	158
Reindeer	130	114	22	108	15	99
Teddy	155	173	31	102	28	167

Appendix B

Features from Accelerated Segment Test

B.1 Unconstrained Classification

Table B.1: Unconstrained classification Confusion matrix II

Dataset	Bad Matches	Good Matches	tp	fp	fn	tn
Art	120	44	51	69	21	23
Books	257	46	139	118	10	36
Cones	157	111	55	102	23	88
Dolls	176	104	68	108	17	87
Laundry	263	30	189	74	5	25
Moebius	138	75	53	85	10	65
Reindeer	154	47	51	103	7	40
Teddy	224	92	97	127	26	66

B.2 Constrained Classification: Triangles

Table B.2: Triangles constrained classification Confusion matrix I

Dataset	Bad Matches	Good Matches	tp	fp	fn	tn
Art	109	55	46	63	26	29
Books	257	46	139	118	10	36
Cones	149	119	56	93	22	97
Dolls	131	149	57	74	28	121
Laundry	246	47	186	60	8	39
Moebius	119	94	43	76	20	74
Reindeer	137	64	45	92	13	51
Teddy	219	97	97	122	26	71

B.3 Constrained Classification: Edges

Table B.3: Edges constrained classification Confusion matrix II

Dataset	Bad Matches	Good Matches	tp	fp	fn	tn
Art	148	16	65	83	7	9
Books	269	34	136	133	13	21
Cones	232	36	64	168	14	22
Dolls	255	25	79	176	6	19
Laundry	257	36	182	75	12	24
Moebius	199	14	56	143	7	7
Reindeer	172	29	51	121	7	22
Teddy	273	43	103	170	20	23

B.4 Relaxed Evaluation Condition

Table B.4: Constrained classification with a "relaxed condition" Confusion matrix I

Dataset	Bad Matches	Good Matches	tp	fp	fn	tn
Art	116	48	50	66	22	26
Books	149	154	138	116	11	38
Cones	149	119	55	94	23	96
Dolls	169	111	66	103	19	92
Laundry	261	32	189	72	5	27
Moebius	133	80	52	81	11	69
Reindeer	149	52	51	98	7	45
Teddy	216	100	97	119	26	74

Appendix C

Triangulation Performance

C.1 Unconstrained Classification

Table C.1: Unconstrained classification for 500 random points Confusion matrix II

Dataset	Theoretical Bad Matches	Found Bad Matches	tp	fp	fn	tn
Art	0	260	0	260	0	240
Books	1	260	0	260	1	239
Cones	2	260	1	259	1	239
Dolls	3	264	2	262	1	235
Laundry	4	269	3	266	1	230
Moebius	5	269	3	266	2	229
Reindeer	6	264	3	261	3	233
Teddy	7	270	5	265	2	228