Faculdade de Engenharia da Universidade do Porto

Global Localization of Vertical Road Signs Using a Car Equipped with a Stereo Vision System and GPS

Miguel Jorge Pereira Cova

Master in Electrical and Computer Engineering, Major Automation

Supervisor: Jaime Cardoso (PhD)

Second Supervisor: Hélder Oliveira (MSc)

Developed at INESC Porto

September 27, 2011

Abstract

Traffic sign detection and recognition has been a topic of study for more than 20 years already. The need to help the drivers being safer in the traffic roads, to insure that the road signs are not disregarded by them, to keep the roads in good conditions or to have a better traffic flow, made traffic sign detection systems eager to be more and more used.

The systems used to detect road signs and their global position still show lack of efficiency and precision. In order to test alternative solutions to this problem, a project was then proposed to improve parts of a mobile road data acquisition system. The goals are to develop a new method to improve the detection phase by reducing the search of vertical road signs; and a method to calculate the global position of vertical road signs.

To fulfill the main objectives of this project, a stereo matching algorithm had to be selected. Five existent implementations were tested and compared with a new stereo matching evaluation. The evaluation is based on the Kendall-tau correlation, in this case, between the obtained disparity map and a manually created ground truth. The results of this evaluation were not conclusive but showed promises for future study.

The reduce search mask was created based in six simple image processing techniques. Those techniques were XZ-transposition, median-filtering, thresholding, XY-transposition, shape selection and region growing. Using those techniques it was showed that for the best results, the search image for road signs can be reduce to around 40% without disregarding any road sign. Other results were showed and compared to the different input parameters of each technique presented.

The global localization method was addressed in two different problems, the distance of the car to the road sign, and the GPS position of the sign. A comparative test was done and the relative localization proved to be consistent with the eye perception of depth. A global localization function was implemented, based in the car GPS coordinates and the relative localization outputted in the previous function, but no testing was done because there was lack of real measures to prove the reliability of the results.

Resumo

Detecção e reconhecimento de sinais de trânsito tem sido um tópico de estudo há mais de 20 anos. A necessidade de ajudar os condutores a ficarem mais seguros na via rodoviária, de assegurar que vejam todos os sinais, de manter as vias rodoviárias em boas condições ou de ter um trânsito menos congestionado, fez com que os sistemas de detecção e reconhecimento de sinais de trânsito sejam cada vez mais requisitados.

Os sistemas usados para detectar sinais de trânsito e a sua posição global ainda apresentam alguma falta de eficiência e precisão. Para testar soluções alternativas a este problema, um projecto foi proposto para melhorar partes de um sistema móvel de aquisição de informação da estrada. Os objectivos são desenvolver um método que melhor a fase da detecção ao reduzir a pesquisa de sinais de trânsito verticais; e um método para calcular a posição global destes sinais.

Para cumprir os principais objectivos deste projecto, um algoritmo de correspondência estéreo tinha de ser usado. Cinco implementações existentes foram testadas e comparadas com um novo método de avaliação de correspondência estéreo. Esta avaliação foi baseada na correlação de Kendall-tau, e neste caso, entre o mapa de disparidade obtido e base de verdade criada manualmente. Os resultados desta avaliação não foram conclusivos mas mostraram algumas promessas para estudos futuros.

A máscara reduzida de pesquisa foi criada a partir de seis simples técnicas de processamento de imagem. Estas técnicas foram: XZ-transposição, filtro-mediana, *thresholding*, XY-transposição, selecção de forma e crescimento de região. Para os melhores resultados estas técnicas mostraram que a imagem de pesquisa dos sinais pode ser reduzida para 40% sem a perda de nenhum sinal de trânsito. Outros resultados foram mostrados e comparados para diferentes parâmetros de cada técnica usada.

O método de localização global foi dividido em dois problemas, a distância do carro ao sinal de trânsito, e a posição GPS do sinal. Foi feito um teste comparativo e a localização relativa provou ser consistente com a percepção do olho da profundidade. A função da localização global foi implementada, baseando-se nas coordenadas GPS do

RESUMO

carro e na localização relativa que resultou da função anterior, mas nenhum teste foi feito porque não haviam medidas reais para provar a fiabilidade dos resultados

Acknowledgements

To Professor Jaime dos Santos Cardoso, for the supervision of this project, useful hints and close follow up.

To Eng. Hélder Filipe Pinto de Oliveira, for the all the support, motivation and special guidance.

To INESC Porto, for all the work conditions and technical support given to this project.

To the VCMI, for the clever insights and comments on the work done and the relaxed work environment.

To all my family, my father that encourages me to work, my mother that makes me healthy, my brothers that make me busy and my grandmothers that make me happy. *(À minha família inteira, o meu pai que me motiva a trabalhar, a minha mãe que me mantém saudável, os meus irmãos que me mantêm ocupado e as minhas avós que me deixam contente.)*

To all my friends, that keep me motivated to work every day even though constantly encourage me to have fun. They know who they are.

"The first principle is that you must not fool yourself - and you are the easiest person to fool"

Richard Feynman

Table of Contents

x

List of Figures

List of Tables

Abbreviations

CC	Camera Calibration
CCD	Charge-coupled Device
FEUP	Faculdade de Engenharia da Universidade do Porto
GPS	Global Positioning System
HSI	Hue-Saturation-Intensity
INESC Porto	*Instituto de Engenharia de Sistemas e Computadores do Porto*
INS	Ineartial Navigation System
JTC	Joint Transform Correlation
LCH	Lightness-Chrome-Hue
MMS	Mobile Mapping System
RGB	Red-Green-Blue
ROI	Region of Interest
SAD	Sum of Absolute Differences
SIFT	Scale-Invariant Features Transform
SURF	Speed-Up Robust Features
SVM	Support Vector Machine
SVS	Stereo Vision System
VCMI	Visual Computing and Machine Intelligence Group
VISAT^{TM}	Video, Inertial and Satellite GPS system

Chapter 1

Introduction

1.1 Context

This project was developed *at Instituto de Engenharia de Sistemas e Computadores do Porto (INESC Porto)* whithin the Visual Computing and Machine Intelligence Group (VCMI), under the supervison of Professor Jaime dos Santos Cardoso and Eng. Hélder Filipe Pinto Oliveira. INESC Porto is a non-profit private association of public utility, dedicated to education, incubation, scientific research and technologic consultancy. INESC Porto produces science and technology that competes and leads national and world markets and develops human resources with scientific and technical qualities that are motivated to be part of the modernization of Portugal.

This project was part of a request of a partner of INESC Porto to develop, under certain conditions, a Stereo Vision System (SVS) to detect road signs. The main project, which this work is part of, is based on the construction of a mobile road data acquisition system that has, among other sensor systems, a SVS and a Global Positioning System (GPS). The SVS is responsible to save the stereo video sequences of the road and process this information to detect road signs and the GPS to give the global position of the car to each recorded stereo frame. The detection, recognition and global localization of road signs is one of the used techniques to obtain road inventory statistics and help the road traffic management.

1.2 Motivation

In the last years many applications of road sign traffic detection and localization systems have been developed. The way to reach autonomous car driving systems has

1

each year been closer and closer but so far only safety systems are being used to help the drivers in the dangerous world of the road traffic environment. Other important systems are the road asset management systems that work daily to help drivers to have better road conditions and good traffic flow. Designing specific algorithms to help small parts of these systems became really important to create a perfect, efficient and errorless road traffic sign detection and recognition systems. This project intends to study and propose other means of image processing to help these systems improve their accuracy. More specifically, the road traffic detection and localization problem is addressed.

My personal motivation to work in this project comes from the interest that I have in the study of image processing techniques. Since I entered the course of Eletrotechnical engineering I had the intention of studying both robotics and image processing but, once I choose to major myself in robotics, the image processing part was forgotten. Later, I was really glad to be offered the opportunity to work and finish my course with a thesis in the area of image processing and fulfill the interest in the both areas of study that I have.

1.3 Description of the Project

This project is part of the project proposed to INESC Porto by one of its partners. The main project is to develop an algorithm that would detect, recognize and calculate the global position of vertical road signs using a SVS and GPS. The stereo cameras are assumed to be already calibrated and so the parameters of them are also already given. The algorithm it is not intended to work in real-time but to process stereo video sequences taken from the system presented above. The car equipped with this system is said to ride at 80 kilometres an hour (km/h) and save internally all the stereo video sequences for off-line processing. This project was divided in two parts: a final detection and recognition part and an initial detection and global localization part. This project is the work of the second part and has the objectives of improving the accuracy and reducing the time of the detection phase and also to develop the algorithm to calculate the global position of the vertical road sings once they are found. The first part of the project depends of the reduced detection phase worked out in this project and detects and recognizes the vertical road signs. Once the sign is found, the function that calculates the positioning of the sign takes that information and uses it to get the final global position of it. Unfortunately the stereo video sequences were not provided in time to test the final part of this work but other methods were used. The combination of the both parts was also not tested till the date of the writing of this document.

1.4   Goals

This project has two main goals to fulfil: It aims to assist the detection phase of the traffic sign systems by reducing the search for vertical road signs; And it also has the objective of designing a method to calculate the global position in GPS coordinates using the disparity map from the pair of frames given by the SVS and the sign already detected in that pair. Both objectives assume the use of stereo disparity maps, so the use of stereo matching algorithms is implied. Instead of creating a stereo matching algorithm, existent algorithms will be used and evaluated to aid the main objectives. Previous search for stereo matching algorithms is to be done along with techniques that can also be of use to reduce the search for road signs and the global position of them.

1.5   Main Contributions

This project introduces a new pre-processing technique to reduce the search for vertical road signs in the detection phase of a traffic sign detection and recognition system. Through simple image processing techniques a reduced search mask is created to aid the detection phase and reduce the effort of the search for these signs. In the way to fulfil the goals of this project a new stereo matching evaluation was approached and some tests showed that this new evaluation might work under certain conditions. Further work on this evaluation also has to be explored to prove it reliability.

1.6   Document Structure

This document describes the work done and the results achieved till the end of this project. Along with this chapter it has more 6 chapters. The chapter 2 is the Background chapter and its purpose is to introduce the reader to some of the previous knowledge needed to read and understand this document. The following chapter 3 is the state-of-art chapter that summarizes, with some examples, the last developments of the area of study under this project. The chapters 4, 5 and 6 are the main chapters and they all represent different stages of development of this project. These three chapters have a methodology section and a results section that addresses to the specific problem under each one of them. The last chapter 7 represents the conclusions of the project and the future work related to it. During the document, the written text will be followed by examples of Figures, tables and equations for the reader be able to follow and understand better all that is said.

Chapter 2

Background

This chapter intends to give the reader the basic knowledge about background mathematics and theory that lies behind image processing regarding stereo utilities and also to familiarize the reader with the vocabulary used within this document.

2.1   Image Formation and Camera Model

"Image formation occurs when a sensor registers a radiation that has interact with physical objects" [1]; it is in this way that Brown and Ballard describe the first moment that you can consider and start to study image formation. This section is based on the information given by Szeliski [2] and Cubber [3]. The image is created by projecting a three dimensional (3D) image into a two dimensional (2D) image plane; that is then modeled by a projection process in which a dimension is lost. The normal way for modeling this process is using central projection, in which a ray coming from a point in space is drawn from a 3D world point through a fixed point in space, the center of projection (also known as pinhole camera model).
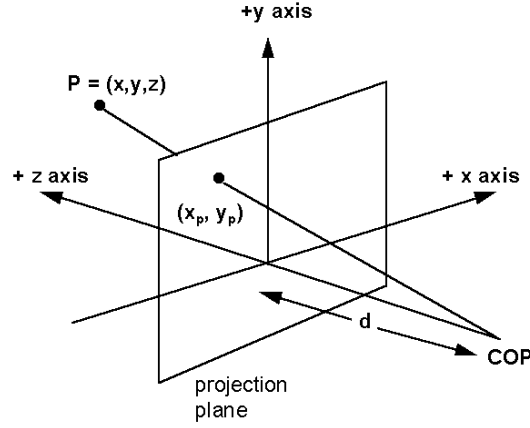
Figure 1 – Example of centre of projection

Now, the central projection is just the mapping of the projective space in 3D to 2D using the camera, the image and the world coordinates $<X, Y, Z>$. The image points $<x, y>$ can be represented in terms of 3D, by the projection matrix:

$$\begin{bmatrix} x \\ y \\ f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad (1)$$

with f being the focal length (the points being in non-homogeneous coordinates). These images points are then mapped into the image coordinates using a multiplication matrix called camera matrix calibration:

$$K = \begin{bmatrix} f & s & c_x \\ 0 & af & c_f \\ 0 & 0 & 1 \end{bmatrix} \qquad (2)$$

that takes into account different centers of projection, non-square pixels and skewed coordinate axes. The variable a that stands for the aspect ratio and the s that stands for skew are, in most of applications in practice, respectively 1 and 0, turning the matrix K to:

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_f \\ 0 & 0 & 1 \end{bmatrix} \qquad (3)$$

This matrix is also used as a camera parameter and can be obtained after various methods of camera calibration (such as [4]). One last projection is needed, the matrix

6

representing the transformation between the camera and the world coordinates. That equation is:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{4}$$

being $<X_c, Y_c, Z_c>$ the camera coordinates, $<X_w, Y_w, Z_w>$ the world coordinates, R the rotational matrix and t the translation matrix. Now, joining all this projections, the camera matrix can be calculated as being:

$$P = K[R \quad t] \tag{5}$$

and it expresses a projective camera of a point in space.

2.2   Stereo Vision

2.2.1   Camera Calibration

Camera Calibration (CC) it is an important procedure on the field of computer vision and a necessary step of stereo vision. Efficient calibration of cameras leads to accurate measurements, used in applications of stereoscopy, motion detection, and object recognition, among others. For a long time, this topic is being under study of many computer vision and photogrammetry researchers, and thus some specific applications/algorithms have been developed to fulfill the needs on this area. The techniques used for CC can be generally separated in two [5]: Self-calibration and object-based calibration. Self-calibration can be done by moving a camera around an unknown static scene. The camera is then calibrated using differences between points or lines detected in both stereo images. With these measures, the goal is to find the camera parameters and respect the two constraints (Kruppa equations and Huang-Faugeras) [6]. Using the same camera with fixed internal parameters, few images are sufficient to obtain the external and internal parameters. Some problems can arise with this method; in the case of featureless images or in dynamic scenes. Another way of self-calibration could be done by finding camera parameters from associating vanishing points with parallel lines [5]. Object-based calibration is made when a camera observes an object with 1D/2D/3D geometry and known dimensions. These dimensions are compared with the ones taken by the camera analysis and the calibration parameters are found. The object-based calibration usually needs two or three planes orthogonal to each other or

sometimes a plane with a known translation [7]. This object-based calibration method if highly efficient [8] but can be very expensive and the setup time to prepare this method it is elaborate and time consuming [9]. The 3D object-based calibration it is a commonly used object-based calibration and uses the information of the metric measures of a 3D object to compare it to the measures obtained from the camera images taken from the object.
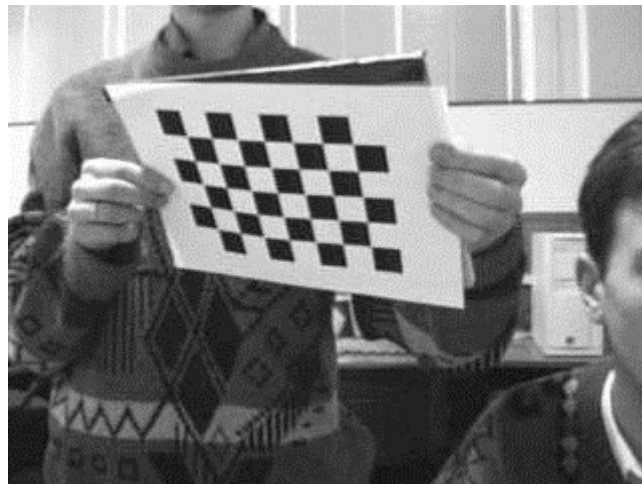
Figure 2 – Example of a calibration using a chess board pattern

The 2D object-based calibration, for example, makes use of a known planar pattern and by moving it with different positions, taking measures at same time, calibration can be done (the use of a chess board image is common in this calibration, example in the Figure 2). The measures taken are used to solve the camera parameters from the homography between the points of the image and the known points of the planar pattern used [9]. A 1D object-based calibration consists on retrieving one dimensional information from the scene, at least three point features [10].

2.2.2 Image Rectification

Within the topic of stereo vision there are some techniques, such as epipolar geometry and image rectification, that are important to understand and use. The epipolar geometry describes the geometric relationship between two perspective views of the same 3D scene [11]. Epipolar geometry defines epipolar lines in the both images of the same frame as well as the epipolar plane, as can be seen in Figure 3.
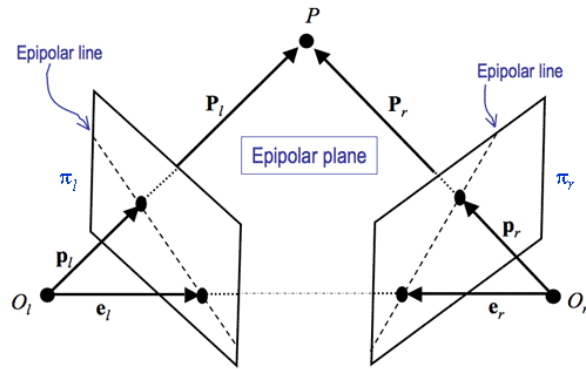
Figure 3 – Epipolar geometry illustration

After computing this epipolar geometry, the epipolar line can be used to constrain the search to the correspondent pixel on the other image. To better solve the stereo correspondence problem before using the epipolar geometry constraint, image rectification can be used to simplify the calculation. Image rectification can be defined as being the transformation of each image plane such that pairs of conjugate epipolar lines become collinear and parallel to one of the image axes. The important advantage of rectification is that computing stereo correspondences is reduced to a 1D search problem along the horizontal raster lines of the rectified images [12]. The Figure 4 clearly illustrates the original images taken by two cameras in different positions (1) and the rectified images (2).
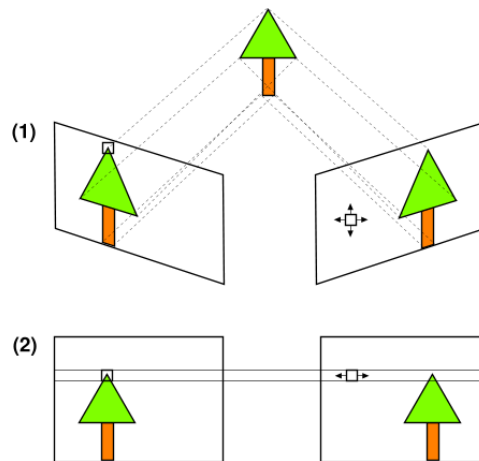
Figure 4 – Image rectification illustration (a) two different images taken by different positions (b) the correspondent images but now rectified

2.2.3 Stereo Matching

Stereo matching it is stereo process that involves finding matches for every pixel in an image to its correspondent pixel on the other pair image. This process it is very difficult to accomplish with success because of colour repeatability and uniform textures. These last features of some images prevent pixel correspondence from being found in the correct way. The usual stereo matching algorithms are based on stereo correspondence using a cost matching window. This cost matching window centres itself on a pixel and calculates a cost (which differs from algorithm to algorithm) around it, and then, it searches for the pixel in the correspondent window that has the nearest cost value around it. Some algorithms use other statistics around the pixel to find the best match, but most can be seen in the Figure 5 [13].

MATCH METRIC	DEFINITION
Normalized Cross-Correlation (NCC)	$\dfrac{\sum_{u,v}\left(I_1(u,v)-\bar{I}_1\right)\cdot\left(I_2(u+d,v)-\bar{I}_2\right)}{\sqrt{\sum_{u,v}\left(I_1(u,v)-\bar{I}_1\right)^2\cdot\left(I_2(u+d,v)-\bar{I}_2\right)^2}}$
Sum of Squared Differences (SSD)	$\sum_{u,v}\left(I_1(u,v)-I_2(u+d,v)\right)^2$
Normalized SSD	$\sum_{u,v}\left(\dfrac{\left(I_1(u,v)-\bar{I}_1\right)}{\sqrt{\sum_{u,v}\left(I_1(u,v)-\bar{I}_1\right)^2}}-\dfrac{\left(I_2(u+d,v)-\bar{I}_2\right)}{\sqrt{\sum_{u,v}\left(I_2(u+d,v)-\bar{I}_2\right)^2}}\right)^2$
Sum of Absolute Differences (SAD)	$\sum_{u,v}\left|I_1(u,v)-I_2(u+d,v)\right|$
Rank	$\sum_{u,v}\left(I_1'(u,v)-I_2'(u+d,v)\right)$ $I_k'(u,v)=\sum_{m,n}I_k(m,n)<I_k(u,v)$
Census	$\sum_{u,v}HAMMING\left(I_1'(u,v),I_2'(u+d,v)\right)$ $I_k'(u,v)=BITSTRING_{m,n}\left(I_k(m,n)<I_k(u,v)\right)$

Figure 5 – Stereo matching metrics and their correspondent definitions [13].

2.2.4 Triangulation

A representation of 3D scene geometry is required for many tasks in computer vision, robotic navigation, computer graphics, and a variety of techniques have been proposed for acquiring the geometry of real-world objects. In these techniques, sensors are used to extract information of the real-world that then is analyzed by computers. The recovery of a 3D scene structure by means of images acquired by digital cameras has been for a long time a topic of interest in many fields such as photogrammetry, remote sensing and computer vision. Although it is well known that stereo-vision is more accurate than monocular setups for scene reconstruction, many successful approaches have been proposed to perform 3D Euclidean reconstruction using

monocular cameras [11]. Another approach to recover a 3D scene structure is to obtain depth, using triangulation [14].

Stereo triangulation refers to the process of determining a point in 3D space given its projections two images. In order to solve this problem it is necessary to know the parameters of the camera projection function from 3D to 2D for the cameras involved (showed in section 2.1). The scenario for this problem is similar to the one represented in Figure 6.
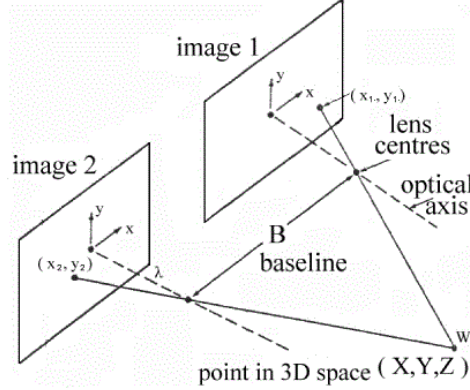
Figure 6: The stereo triangulation problem

The problem starts by the existence of a pair of two fixed cameras separated by a constant distance, known as the baseline. This baseline constraint allows us to compute 3D estimates from image measurements [15]. Given a pair of corresponding points, every one in each camera, we can triangulate [11] the position of the point in 3D with respect to the stereo-rig. In such a configuration, where the cameras are fixed in the same line and orientation, a stereo correspondence pair lies along the same horizontal axis (epipolar line) in both image planes [11]. Given this, the point in 3D space observed in two different frames, can be obtained by the following triangulation equation:

$$P(x, y, z) = \left(\frac{x_l.b}{d}, \frac{y_l.b}{d}, \frac{f.b}{d}\right)^T \tag{6}$$

where, the point $(x_l, y_l)^T$ are the coordinates of the projection in the left image frame, the $d = x_l - x_d$ is the disparity, b the baseline, f the focal length of the camera and P is the point is the 3D point in the scene, expressed with respect to the left camera [16].

The disparity is the difference in image location of a point seen by the left and right camera, resulting from the cameras horizontal separation. The knowledge of disparity

can be used in further extraction of information from stereo images, such as depth or distance calculations. (see Figure 7);
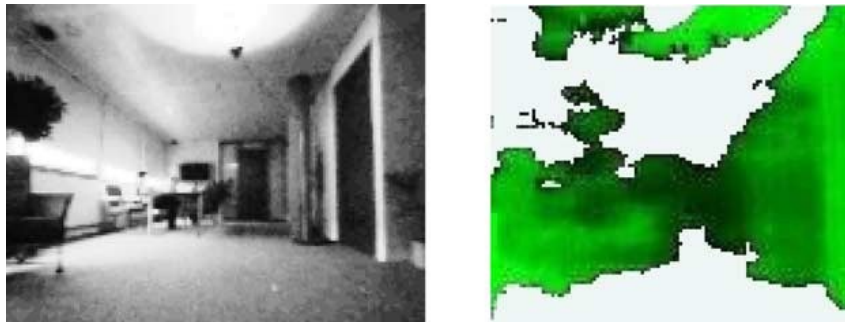
Figure 7: Plain image (left) and the corresponding disparity map (right). In the disparity map, light gray areas are missing values and dark areas correspond to points that are far from the robot [17]

2.3   Global Positioning System (GPS) and GPS Coordinates

The Global Positioning System is a navigation system guided through satellites that provides location and time information anywhere in the earth. The system works using satellites that are in orbit around the earth and transmit radio signals that consist of the satellite´s position and the time it transmitted the signal. These signals are then received on Earth with a simple device that can be bought at any electronics store to various localization applications. The distance between a satellite and a receiver can be computed by subtracting the time that the signal left the satellite from the time it arrives at the receiver. If the distance to four or more satellites is measured, then a three-dimensional position on the Earth can be determined [18]. The position can then be transmitted in terms of GPS coordinates. The GPS coordinates are commonly displayed in Latitude and Longitude. The latitude of a point on the Earth's surface is the angle between the equatorial plane and a line that passes through that point and the longitude is the angle east or west from a reference meridian to another meridian that passes through that point. These coordinates can be displayed in various ways; two common ones are the decimal degrees (DD) and the degrees/minutes/seconds (DMS). The DMS coordinates came up in the form of Latitude 60° 32' 59.99" Longitude -45° 32' 59.99" and DD coordinates came up in the form of simple degrees as Latitude 60.55" Longitude -45.55" (conversion from the DMS coordinates in the previous example to DD).

Chapter 3

Traffic Sign Detection, Recognition and Localization

This chapter aims at giving the reader the latest scientific developments and technologies concerning traffic sign detection, recognition and localization (since most study have the last part incorporated). This research intends to give knowledge and ideas for the development of this project to fulfill their objectives. There are two main areas of concern for this project; the first one relates to road sign detection and the second to road asset management regarding global localization of traffic road signs. The recognition phase is also a topic of interest because of its big influence on the detection phase, as being the next step in most of the systems developed nowadays.

Before going into detail, it is important to give a succinct explanation of which environment this system are used. Many mobile road data acquisition systems exist nowadays and all of them have different sensor systems that help them extract in many different ways the information of the roads. It is also implicit that the systems and the algorithms used to process all the recorded data are different, because it depends on the equipment used by them i.e. the type of sensor systems. These different sensor systems allow then different techniques to be used to process the acquired data. Traffic sign detection and recognition is mainly based on the usage of images taken by a camera positioned in a car moving in the road. Global localization of the road signs can be discovered by the use of SVS and GPS but also with powerful infra-red sensors and GPS. Some existent techniques are presented below that show the read the current state of art in this matter.

3.1   Traffic Sign Detection, Recognition

The developments on traffic sign detection and recognition methods have been under study for many years already, and dozens of studies and techniques have already been developed in this area. A study was made in 2006 by Nguwi et al. [19] that summarizes most of the methods developed till then and the next pages will be based on that study following a more recent update from the evolution of those methods till the present day.

For better understanding lets first contextualize the motivation for every researcher of this area and the need for the study on this area to be developed. Road signs are an important road asset that is used by drivers to drive under safety regulations to avoid accidents and keep the order in the road tracks. The motivation can then be understood by trying to prevail this accidents from happening resulting from the distractions of the drivers that can sometimes disregard some road signs. The automatic road sign detection and classification is then used to some applications such as:

- Alert the drivers for the approximation of a road sign (detection part) and which sign it is (recognition part). Most of the methods under study and that have been developed had the objective of fulfilling this purpose, especially brands of commercial cars that have research teams working with this final objective;

- Road asset management; that by consequence uses the information of the position of the road signs to evaluate road traffic conditions to have a better management of the road traffic and optimize it.

Innumerous issues appear when the problem of road sign detection and recognition is addressed. The conditions of the road sign make the problem of detection and recognition a very challenge and debated problem. Some of those conditions are:

- Lighting conditions of the scene, that depends a lot in the country weather, the day and light situation and that influences the image processing part (example in Figure 8);

- The unsteady camera that is inside the vehicle, due to vibration of it, and causes blurring in the image;

- The sign is not always facing the vehicle in the same way, so the processing part has to clearly adapt itself to all different angles that the sign might take in relationship with the vehicle (example of tilted sign in Figure 8);

- The sign deterioration with time and ambient conditions, that affects essentially the recognition part, makes it more difficult to get which sign is it; This can also alter the shape of the sign, causing another problem for the application of any method.
- The signs which are in the same vertical pole, or coupled together can be more difficult to process (a cascaded example can be seen in the Figure 8);
- The size of the signs depends in the distance to the vehicles. The size changes from a view to a view and that has also to be taken into account.
- The objects that might be occluding same part of the signs also input difficulty in the detection and recognition methods

And there are more problems along this list, making a perfect solution for this problem really difficult to obtain, creating even methods for sub problems inside the all detection and recognition process.
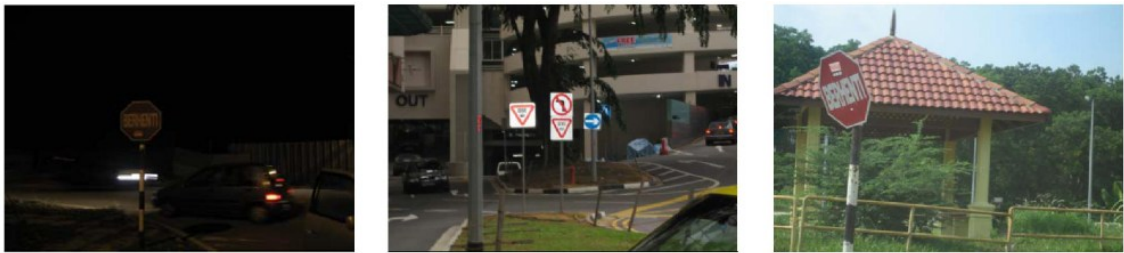
Figure 8 – Images taken with different sign conditions [19].

Many solutions to this problem have been presented over the last 20 years and Nguwi et al. decided to agglomerate all the solutions on three groups. It is important to say before that most of these solutions are based on the withdrawal of sign features such as colour, shape and pattern. Colour-Based approaches, Shape-Based approaches and Other approaches sum up the existent types of approaches.

- Colour-based approaches face in general less problems that shape-based approaches, but however there are some limitations such as weather conditions, bad sign conditions and complicated scenes that make the colour-based approaches. The use of shape-based methods can compensate some of these last problems of the colour-based approaches.
- The Red-Green-Blue (RGB) colour format is not very suitable for image segmentation because of its variation under different light and weather scenes. Other colour spaces are used, such as Hue-Saturation-Intensity (HSI)

(Figure 9) or Lightness-Chrome-Hue (LCH) that outcome some of these problems.

- The Other approaches are the ones that the researchers are more focused now because of its good results. Space-Variant Sensor Window and Neural Networks approaches already can achieve good results.

Figure 9 – RGB colour space and HSI colour space [19] (correspondently from the left image to the right)

In 2006, García-Garrido et al. [20] proposed a method that in the detection phase uses the Hough Transform and in the recognition phase a Neural Network (NN). A tracking method is also applied using a kalman filter in order to provide the system with memory and as a consequence only detection the same sign once (as an example Figure 10 illustrates this phenomenon). In the results it is showed that the camera used for the system should be positioned towards the direction of the car. Their system has a good performance once the signs are facing the camera correctly, for example, if it appears a circled sign facing the camera but the image records it as an ellipse more two parameters have to be added to the Hough Transform to be detected. They system shows to have an overall run time of 30ms per frame being then candidate for real-time applications.

Figure 10 – Example of kalman filter usage in road sign tracking [20].

Lafuente-Arroyo et al. [21] show, by an analysis to the importance of the use of a tracking method in a detection and recognition system of traffic signs that using the Kalman filter estimator reduces significantly the appearance of non-sign detections. Broggi et al. [22] design a system that in the detection phase uses colour segmentation and shape recognition, with a pattern matching and an edge detection algorithm. The overall system shows to be successful for different light environments but inconsistent with subsequent frames (no tracking system used). The signs are detected at an average distance of 20m that can be slightly change if the focal length is changed, missing however the close signs at the side of the road. Broggi et al. advise the future work on solving the problem of the non-interested junction signs once they are of no concern to the drives in certain directions. In 2008, Chen et al. [23] built a system that uses an Adaboost algorithm with a cascaded algorithm for the detection phase and a thresholding with a connected component analysis for the recognition phase. The results is said to be robust and accurate of the system even though no data is showed to confirm it.

In 2009 more methods were developed, Youngping et al. [24] builts other detection and recognition system that also uses the Adaboost algorithm but now combined with the Hough Transform. In the recognition phase it uses an improved template matching based in the use of sequential frames to reject non-signs. The results show the efficiency of the system in dark scenes and bad sign conditions (Figure 11) and also can be used to improve the reliability and reduce computational time.

Figure 11 – Different sign detections scenes under different sign conditions [24].

Feixiang Ren et al. [25] present a system with a detection phase that has a RGB conversion to HSI colour space and Hough Transform to recognize special and potential sign shapes and a recognition phase that uses Scale-Invariant Feature Transform (SIFT) or Speed Up Robust Features (SURF) to features matching. An example of the SIFT application on traffic sign recognition can be seen in the following Figure 12. This approach results in a hit rate bigger than 95% and an execution time of 80ms for the detection phase and 100ms for the recognition phase.

Figure 12 – Example of sign recognition using SIFT [25].

Another system with the use of a different colour space has been implemented by Chourasia et al. [26] that in the detection phase does a colour centroid matching and colour segmentation using the luma, blue-difference and red-difference chroma components (YCbCr) colour space which proved to be make the detection process independent of the illumination characteristic of the images. The sign is extracted by considering the maximum distance boundary pixels from the centroid and a minimum distance classifier is used to detect the sign shape. A NN is used for sign recognition and in the end the results show a 100% effectiveness for colour extraction, 98% for shape recognition and an overall result of 92% for the all detection and recognition system. The last documented paper for 2009, present by Ruta et al. [27], has a detection, tracking and recognition phases such as in [20]. The detection phase has a colour segmentation that extracts Regions Of Interest (ROI) representing instances of equiangular polygons. The tracking algorithm is used to reduce computation time and predict position and scale. In the recognition phase the discrete-colour images are compared with a set of discriminative local region off-line from idealised template sign images. This set is uses the principle of one-vs-all dissimilarity maximisation based on the so called Colour Distance Transform. The results prove to be more effective than the systems that used the Adaboost and Principal Component Analysis (PCA).

Moving to the year of 2010, three other papers are commented with alternative suggestions for this problem. Qin et al. [28] map a matrix with the colour distance and then apply a distance threshold (the Figure 13 shows an example of the application of this threshold). The detection phase is followed by the recognition phase where there is the use of linear Support Vector Machine (SVM). The SVM is used with the distance to border features of the segmented blobs to get the shape information and then realize the Hough classification based on a Colour-Geometric Model. The traffic sign classification is implemented using Radial Basis Function (RBF) Kernel based support vector machine with edge related pixels of interest as the feature. As a conclusion, the threshold values seem to be very important to the success of this system that proved to be succinct and conductive in real-time processing.
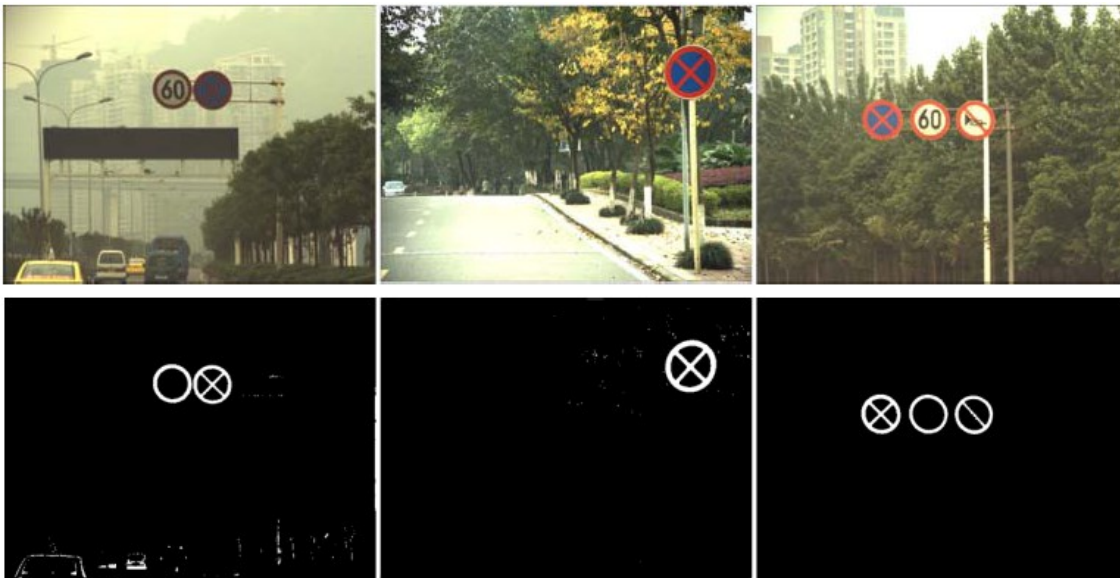
Figure 13 – Result of the applied distance threshold in [28].

Chuan et al. [29] focus on segmentation and, by the analysis of the Maxwell Triangle, the traffic sign is segmented by region growing with maximum and minimum threshold. The result of the method show proves of robustness and possibility of usage in real-time applications, paving a foundation for improving the precise recognition of sign candidates. An attention based traffic sign recognition is also a method used to solve this problem and an example of it was developed by Kastner et al. [30]. The attention based system outputs a ROI with a potential traffic sign candidate (Figure 14) and after an array of weak classifiers is used to compute the probabilistic value of each sign classes. The result for this approach shows specially the efficiency on the detection of all types of signs and is intended as a first step towards an overall traffic sign recognition.
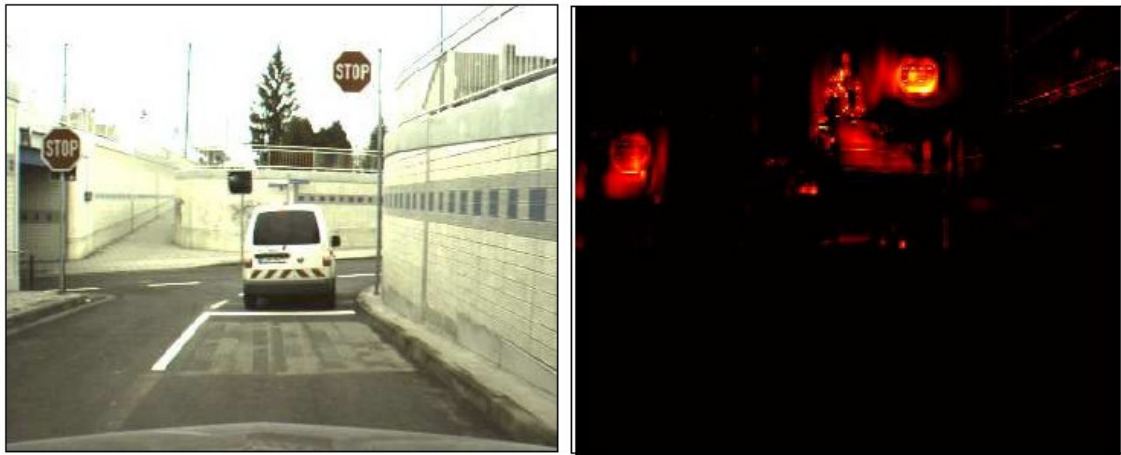
Figure 14 – Example of ROI in [30] (Original image in the left and ROI image in the right).

In the year 2011 more elaborate methods were design and are presented here to show some of the most updated technologies on this matter. Chen et al. [31] proposed a method that uses a Simple Vector Filter algorithm for colour segmentation, Hough transform and curve fitting approaches in shape analysis to divide traffic signs into categories according to the colour and shape properties which be considered the first part of the recognition phase. In the second part of the classification phase, the Pseudo-Zernike moments features of traffic sign symbols are selected for classification by SVM. The method is tested for a big set of frames and it has a hit rate around 98% depending on the illumination of the frames (weather conditions of the day and daylight time). Houben [32] decided to explore the Hough-like methods and tried to increase their performance based on simple adjustments. After running the Hough-like methods, to a publicly set of images, a new probabilistic measure for traffic sign colour segmentation is found and hence a new Hough-like algorithm for detecting circular and triangular signs. The results prove to increase significantly the detection performance in this kind of systems. Khan et al. [33] propose a method that uses Joint Transform Correlation (JTC) with the integration of shape analysis. There are three stages in the approach: The first stage segmentation is done by clustering the pixels based on the colour features to find the ROIs; The second stage is a traffic-sign detection using two novel classification criteria; The third and last stage is the recognition of the road sign using a distortion –invariant fringe-adjusted JTC (FJTC) to match the unknown signs with the reference signs stored in a database. The results show a high success rate and a very low false hit rate and demonstrate that the proposed framework is invariant to translation, rotation, scale and partial occlusions of the road signs (example results in Figure 15 to prove the conclusions).
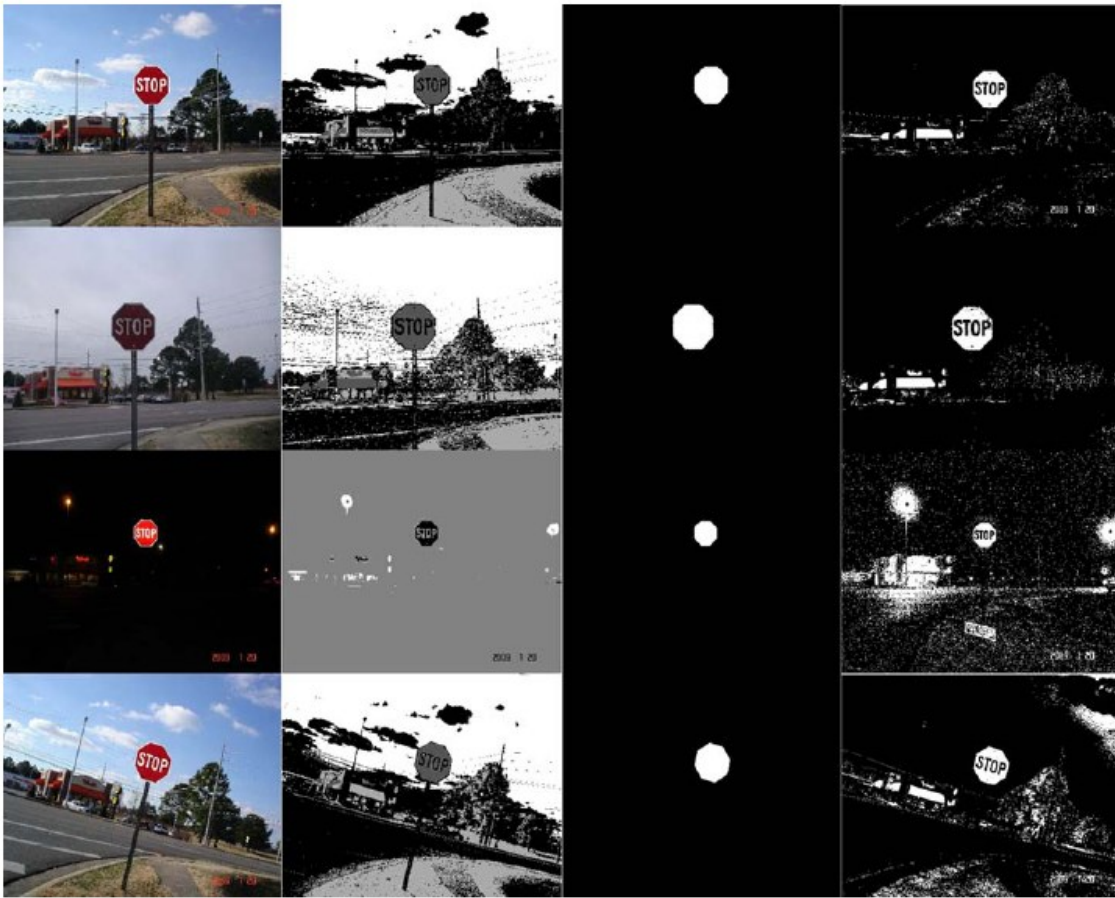
Figure 15 – Example results shown in [33]. From the left to the right different conditions of the stop sign are processed, detected and recognized.

Deguchi et al. show in [34] the effectiveness of their approach on adaptive learning based on online gathering of training samples. Basically, the system that already has a sign detector previously trained off-line, is constantly re-trained online (or in other words, at real-time) and because of that becoming more efficient on traffic sign detection. Gu et al. [35] prove with their study the importance of high resolution images in the detection process. The install of a dual-focal active camera system, the shape, colour features and the relationship between continuous frames are used together in all the traffic sign detection system. The Haar Wavelet features can also be used to detect signs [36]. In this last method a colour-based segmentation is used to find ROIs that are then used to find sign candidates that are detected by a set of Haar Wavelet features obtained from Adaboost training. After this, the SURF is applied for sign recognition. The results have a 90% recognition accuracy on a real-time application of this method.

As seen, through the techniques presented, there is still some space for improvement. All of the techniques showed have some problems with the detection or

recognition phase's efficiency and accuracy. The time of processing can also be factor of improvement for real-time applications. For this reasons a reduced search mask for the detection phase is approached and studied to be considered as a measure of improvement for some of the methods showed above.

3.2   Traffic Sign 3D Localization

The global localization of traffic signs has been under study for some years especially for road asset management purposes. Technology advances in Mobile Mapping Systems (MMS) have appeared and quite some applications have been in use already that involve the recover data of the 3D location of the traffic signs. Tao wrote an article in 2000 summarizing most of the technologies used for mobile mapping technology for road network data acquisition. These systems have a wide variety of sensor integrated systems that because of technology are more and more fast and cheap every year that is passing by. These MMS incorporate three different developments over time [37]:

- Photo-logging: Because these systems did not had the capacity to work at real-time and many algorithms that would process all the data was not yet discovered, a set of images was recorded with previous off-line processing.
- Video-logging: This development came after the photo-logging and it facilitates a lot the interpretation of the road assets because of the better information retrieval that can be made.
- Mobile mapping: This is the latest technology development in the multi-sensor integrated mapping technology business. A common feature of MMS is that more than one camera is mounted on a mobile platform, allowing for stereo imaging and 3D measurements. Direct georeferencing of digital image sequences is accomplished by the multi-sensor navigation and positioning techniques. Multiple positioning sensors, GPS, Inertial Navigation System (INS) and Dead-Reckoning (DR) are combined for an efficient data processing georeferecing (Figure 16).

A global coordinate system   A local coordinate system   An object of interest
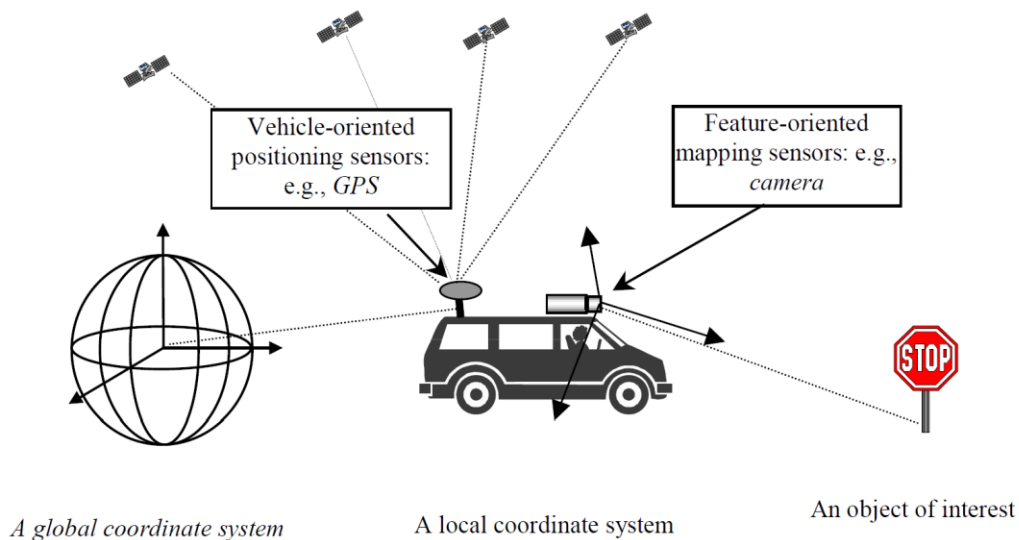
Figure 16 – Illustration of a mapping mobile system shown in [37].

An example of one of these integrated solutions (among others such as ARAN [38], TruckMAP [39], GPSVision [40]) is the Video, Inertial and SATellite GPS (VISAT[TM]) system [41]. The VISAT[TM] system integrates a cluster of charge-coupled device (CCD) cameras, a GPS receiver, and an INS, to automatically collect data in a road corridor at velocities up to 120km/h and to store this data in a Geographic Information System (GIS) format [41]. The updated GPS/INS information is used to geometrically correct the images collected by the CCD cameras which record all the data along the road in a window of 50m. The cameras and the INS system are synchronized with the GPS receiver. The error margin for this system can be summarized by the table 1 as being considerably satisfying.

Component	Average	Std Dev	RMS
Easting	2.79 cm	± 2.75 cm	3.86 cm
Northing	1.81 cm	± 2.60 cm	3.10 cm
Heigh	2.13 cm	± 1.33 cm	2.49 cm
Planimetric	3.95 cm	± 3.08 cm	4.95 cm
3D	4.73 cm	± 2.98 cm	5.54 cm

Table 1 – Results for the VISAT[TM] system

Basically, this system, has a GPS receiver to obtain the information about the global position of the car, has the cameras to withdraw all the information about the traffic road and then in an off-line environment it combines all the data that has been acquired and processes all of it to get information of all the road inventory (such as the

position of traffic signs). In general, there is a car that collects all the necessary data and then there is a processing stage that takes all the data and runs some of the algorithms approached in section 3.1 to obtain the traffic sign information (among other information such as the road width or the height of bridges). Portable systems also exist and can be putted into chosen cars to collect also this data [42]. This relative novel system (built in 2008) allows also allows both geometric and photometric measurements to be made on objects in the scene (such as traffic signs). Any object of the scene can be positioned on the earth´s grid trough the fusion of a SVS with global positioning system technology. As said before, nowadays a big number of MMS already exist with different kinds of sensors, GPSs and that use different algorithms between each other (like the ones shown in section 3.1).

All of the existent applications that calculate the global position of the road signs show deviations from its real values. There are still some possible improvements in this area with the rise of new methods for its calculation or improvement of existent ones. In this project, a simple method of calculation is addressed to solve this problem.

Chapter 4

Stereo Matching Evaluation

This chapter represents the developments done within this project in the area of stereo matching evaluation. Firstly, a state of art part is introduced to the reader to some of the existent algorithms of stereo matching and to explain the ones that were used in this project. Secondly and finally, the development part of stereo matching evaluation is presented with a proper summary, methodology and results.

4.1 The Middlebury Stereo Evaluation

In 2002, Scharstein et al. design a very famous method to evaluate stereo matching algorithm that now a days is considered the most accepted comparing evaluation between existent algorithms [43]. In this work, taxonomy of dense, two-frame stereo methods was developed to assess different components and design decisions made in individual algorithms. Also to evaluate the stereo algorithms a multi-frame stereo data set with ground truths was produced to compare efficiently the results of each algorithm. Using this evaluation more than 100 algorithms were submitted and ranked according to its results (the rank can be seen in [44]).

The first 3 algorithms are going to be present along with their summary of the techniques used on each of theme. The first algorithm was developed by Mei et al. and it is based on several key techniques: Absolute Differences – Census (AD-Census),

which the correspondent formulas can be seen in section 2.2.3, cost measure, cross-based support regions, scanline optimization and a systematic refinement process [45]. This system presents itself in the first position in the Middlebury rank but it shows that it is still really challenging to produce good results in the real world that is affected by image noise, rectification errors and illumination problems that might cause serious problems for cost computation and support region construction. The Figure 17 illustrates the results of this algorithm in the Middlebury evaluation dataset that runs in 0.1 seconds and as a 3.92% of bad pixels.

Figure 17 – Stereo matching results from the algorithm in [45]. From the top to the bottom is the original image, the correspondent disparity map and the error map with the bad pixels.

The second ranked algorithm was developed by Klaus et al. and proposes the utilization of colour segmentation on the reference image and a self-adapting matching score that maximizes the number of reliable correspondences [46]. After this a robust plane fitting technique and a Belief Propagation (BP)-optimization makes up for excellent results on the Middlebury evaluation. The Figure 18 illustrates the results for the Middlebury evaluation data set and can be compared with the previous results from the algorithm ranked in first place. This method takes between 14-25 seconds of running time and it has a percentage of 4.23 of bad pixels.

The third method was design by Wang et al. and it is based on an inter-regional cooperative optimization. The method follows the steps [47]:

- Firstly, a color based segmentation method is used to segment the reference image into regions with homogeneous color.
- Secondly, a local window-based matching method is used to determine the initial disparity estimate of each image pixel. And then, a voting based plane fitting technique is applied to obtain the parameters of disparity plane corresponding to each image region.
- Finally, the disparity plane parameters of all regions are iteratively optimized by an inter-regional cooperative optimization procedure until a reasonable disparity map is obtained.

This method shows an example running time of 20 seconds (for the tsubuka image) and 4.41% of bad pixels for the Middlebury evaluation (Figure 19 shows the image results).

| reference images | ground truths | our results | 'bad pixel' maps |
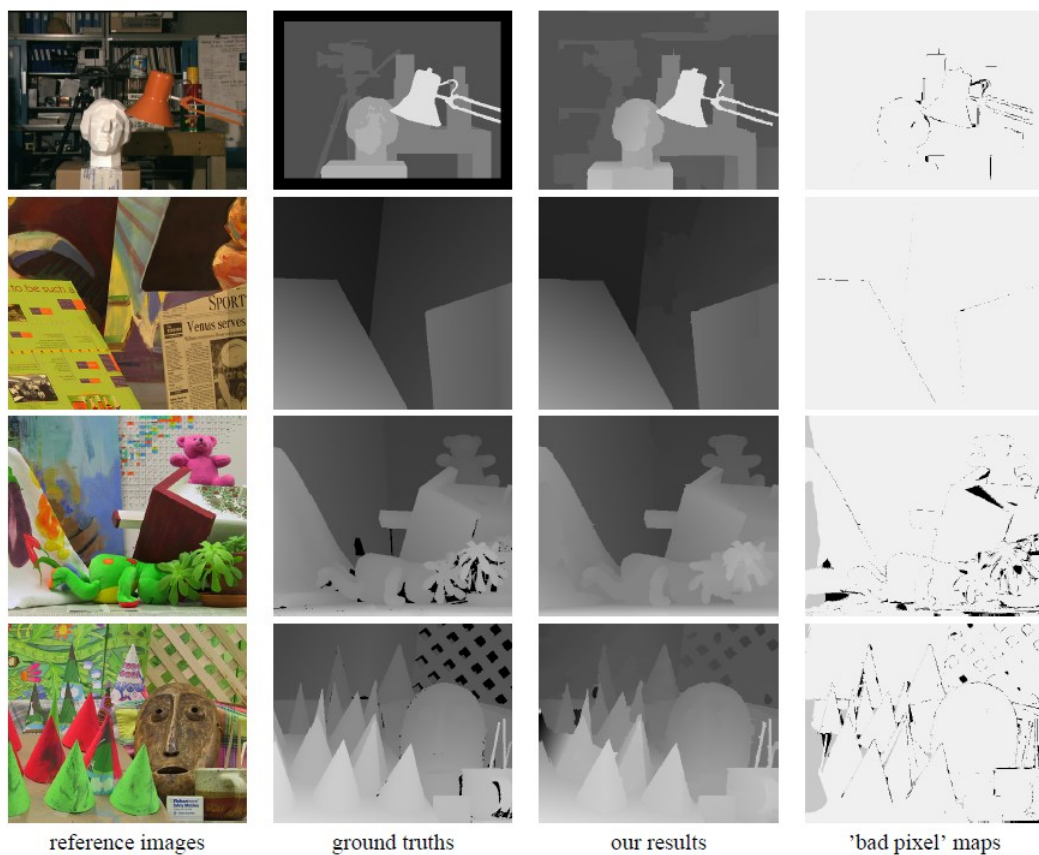
Figure 18 - Stereo matching results from the algorithm in [46].
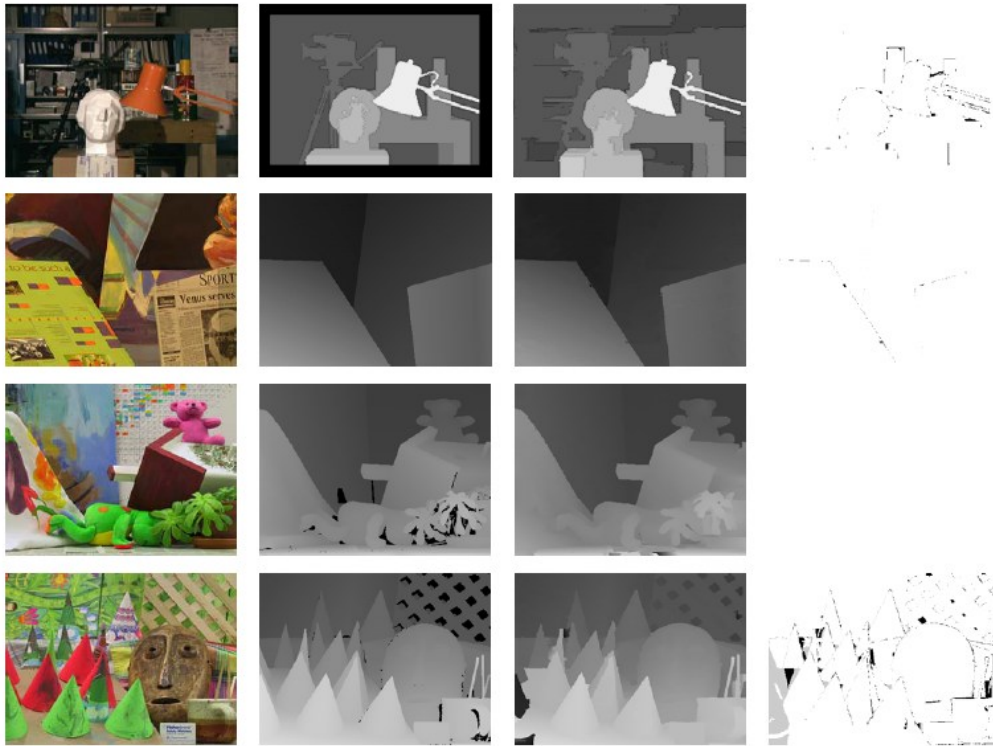
Figure 19 - Stereo matching results from the algorithm in [47]. From the left to the right is the original image, the correspondent ground truth, resulting disparity map and the error map with the bad pixels.

4.2 **The Selected Stereo Matching Algorithms**

In this section the stereo matching algorithms used are going to be presented along with a summarized explanation of how they work and example results given by their developers. It is important to say beforehand that the search for these algorithms was based upon two conditions: they had to have an open Matlab code implementation or they´re execution file had to output the disparity image and be able to work with the dataset used in this project. Because of these conditions the few available methods ranked in the Middlebury evaluation (most of them don´t have available code or execution files open to public use) were not taken into account.

With this reduced search 5 algorithms were found, 4 full available Matlab code implementations and one execution file previously compiled from C++ (no code available). The names of these algorithms were simplified in order to address them more easily during the comments of this project.

The "Blockset" Algorithm

This algorithm is based on the Stereo Vision Demo of the Computer Vision System Toolbox^{TM} of Matlab that uses the methods, from various papers, to obtain the 3D projection of a scene [48]. The algorithm includes seven steps:

1. Read Stereo Image Pair
2. Basic Block Matching
3. Sub-pixel Estimation
4. Dynamic Programming
5. Image Pyramiding
6. Combined Pyramiding and Dynamic Programming
7. Backprojection

Only the first four steps were used because the all processing of the steps took a lot of processing time and the 4 first steps seemed to be enough to have a good disparity image.

The block matching involved the use of a matching window of chosen size and the search in between a disparity range that could also be chosen. The image regions are compared using the sum of absolute differences (SAD). The better block matched would take the correspondent disparity range according to its deviation from the correspondent pair image. Sub-pixel estimation is done to smooth the fast changes in neighbourhood pixels or, in other words, to smooth the disparity image. The dynamic programming a smoothness constraint is used. This constraint allows a pixel to have a disparity with possibly sub-optimal cost for it locally. This extra cost must be offset by increasing that pixel's agreement in disparity with its neighbours. In particular, we constrain each disparity estimate to lie with ±3 values of its neighbours' disparities, where its neighbours are the adjacent pixels along an image row. Finally the image disparity is obtained and is this one that it is used in the rest of the project. The example result showed by Matlab can be seen in the following Figure 20.
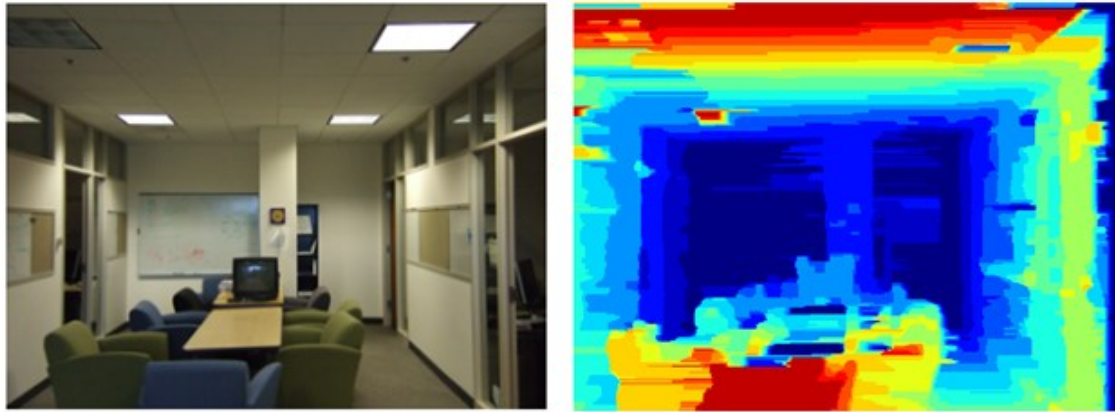
Figure 20 – Stereo matching result after block matching, sub-pixel estimation and dynamic programming [48]. In the left is the original image and in the right the correspondent disparity map (red colouring for close objects and dark blue for farthest objects)

The "Lankton" Algorithm

The second algorithm was developed by Shawn Lankon [49] is based on method that is ranked in the second place in the Middlebury evaluation. In this approach a reference image is chosen (in this case, the right image), and the other image slides across it. As the two images 'slide' over one another the algorithm subtracts their intensity values. Additionally, it subtracts the gradient information from the images (spatial derivatives). Combining these two processes it is achieved better accuracy, especially on surfaces with texture. The algorithm performs the slide-and-subtract operation from right-to-left (R-L) and left-to-right (L-R) and then it tries to eliminate bad pixels in two ways. First, the disparity from the R-L pass or the L-R pass are used depending on which has the lowest matching difference. Next, all the bad points are marked where the R-L disparity is significantly different from the L-R disparity. Finally, the algorithm reaches to the final disparity map (an example is showed in the Figure 21).

Figure 21 - Stereo matching result from [49]. In the left is the original image and in the right the correspondent disparity map (red colouring for close objects and dark blue for farthest objects)

In the next step, a filtering method is applied where the image information is combined with the pixel disparities to get a cleaner disparity map. First, the reference image is segmented using a technique called Mean Shift Segmentation which is basically a clustering algorithm that "over-segments" the image. Then, for each segment, the algorithm looks at the associated pixel disparities. In this rather simple implementation, each segment has the median disparity of all the pixels within that segment. The resulting disparity image shows a more smoothed map showed in the following Figure 22.
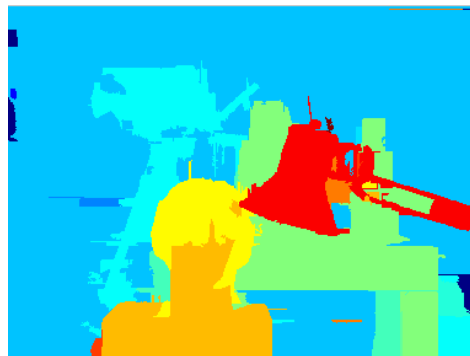
Figure 22 - Stereo matching result shown in [49] after applying the filtering method (Figure 21 shows the previous processed images)

The "Region" Algorithm

This algorithm was built by Alagoz and it represents one of the algorithms of its work in [50]. In this work, region based stereo matching algorithms are developed for extraction of depth information from two colour stereo image pairs. A filter eliminating unreliable disparity estimation was also used for increasing reliability of the disparity

map. The chosen algorithm was the so called Global Error Energy Minimization by Smoothing Functions. This method can be summarized in three steps:

1. For every disparity, in disparity search range, calculate error energy matrix.

2. Apply average filtering iteratively to every error matrix calculated for a disparity value in the range of disparity search range.

3. For every pixel, find the minimum error energy and assign its disparity index to the disparity map.

An example for this algorithm can be seen in the following Figure 25, with the use of a 3x3 matching window.
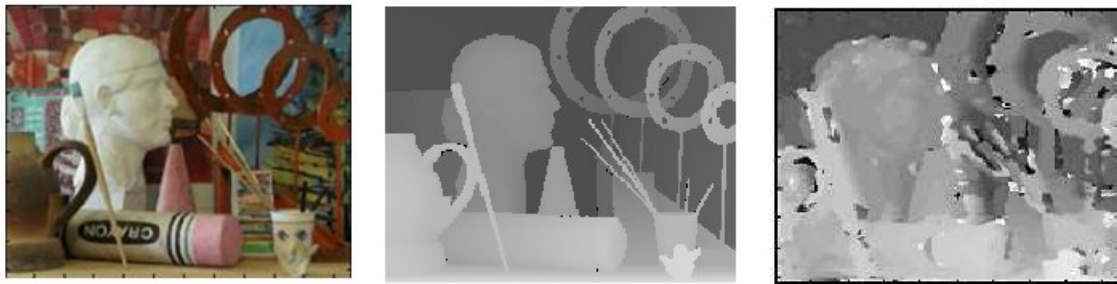
Figure 23 – Stereo matching result of [50]. From the left to the right, there is test image, the correspondent ground truth and the resulting disparity map.

The "Vision" Algorithm

This algorithm was design by Wim Abbeloos and it uses a simple SAD in two rectified pair of images to obtain the disparity map [51]. Not much can be said from this implementation since it was found in the Mathworks forum and no more documentation was provided by the author to present or illustrate previous results for it.

The "Yang" Algorithm

This stereo algorithm was developed by Yang et al. and the executable was provided by the author to obtain the disparity map from the input of two stereo image pairs. This executable has the algorithm described in [52] and it is a belief propagation based global algorithm that generates high quality results while maintaining real-time performance. This algorithm is ranked in the first position of the Middlebury evaluation

test. An example of the output of this algorithm can be showed in the Figure 24 confirming a good disparity map output.

Figure 24 - Stereo matching result of [52]. From the left to the right, there is test image, the correspondent ground truth and the resulting disparity map.

4.3 Methodology

In order to get the distance to the road signs and the reduced search mask the disparity map is needed from each pair of frame of a selected stereo video sequence. The stereo video sequence used was taken from an available dataset called Karlsruhe Dataset [53]. An example of the images of this Dataset can be seen in the Figure 25 and Figure 26. This Dataset has various stereo video sequences and, in between them, 25 pairs of frames of different sequences, containing none, one or more signs, were chosen with the objective of fulfilling the goals of this project (Figure 25 and Figure 26 as an example). The images are already rectified and most of them have the size of 1344x393.

Figure 25 – Example image withdraw from [53].

Figure 26 - Other example image withdraw from [53].

Having already available test images, five algorithms were chosen to compare and evaluate the output disparity maps and continue the project with the one that outperforms best in the results of this section. The algorithms were already presented in the section 4.2 and are now used as followed:

- Algorithm 1 - "Blockset";
- Algorithm 2 – "Lankton";
- Algorithm 3 – "Region";
- Algorithm 4 – "Vision"
- Algorithm 5 – "Yang"

The methodology used was based on a two-stage evaluation and elimination. Both steps used the same pair of frames but different methods of evaluation.

4.3.1 The First Stage of Evaluation

In the first stage, the objective was to eliminate the two algorithms that qualitatively had worst results on their outputs. The qualitative factor used to evaluate the disparity outputs of each algorithm was the human eye and its perception for distance. The output disparity images are gray images and their correspondent matrixes have values are integers in the range of [0,255]. The pixel values represent the distance between the camera and the objects in the photo. This distance can then be interpreted as a difference between gray levels, with white meaning close to the camera and black the opposite. The five algorithms run the 25 pair of frames with standard parameters (in this step no much concern was given to find the best input parameters for each algorithm) and then the evaluation was done, and as said before, the three best algorithms would jump to the second stage of evaluation.

4.3.2 The Second Stage of Evaluation

The second stage appeared because of the need to have a quantitative way to evaluate the algorithms besides only using the eye as evaluation performance indicator (qualitative evaluation). The problem then arise; there was no evident way of doing a quantitative evaluation, especially because there were no ground truth images for the Dataset used (such as Figure 27).
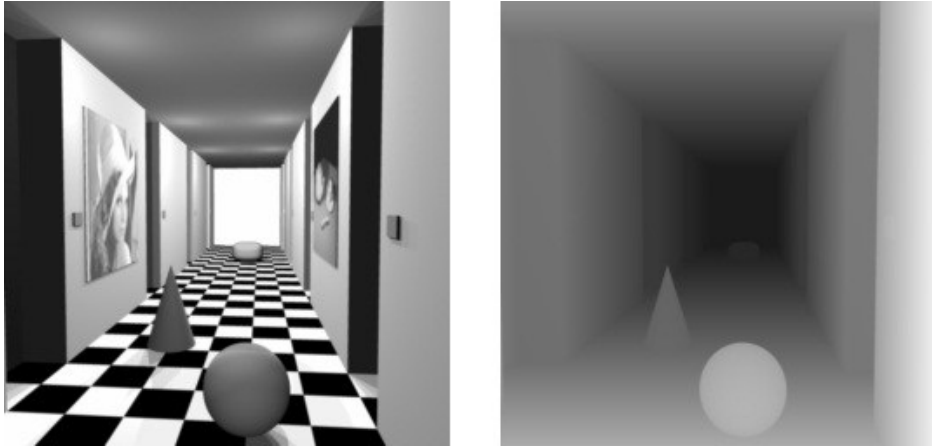
Figure 27 – Example of an image and the correspondent ground truth.

So, the question was how to evaluate on a precise way the outputs since there is no ground truth for them. The concept was to create with a simple image editor a ground truth that would have 5 to10 different intensity levels and compare it with the disparity output of the each algorithm. The division between these levels was simply based on the perception of depth that the human eye has of an image. The divisions of these areas are difficult to obtain in detail, so image parts, as for example small objects, had to be ignored. Two examples of a created ground truths and the correspondent images can be seen, respectively, in the Figure 28 and Figure 29 and in the Figure 30 and Figure 31.

Figure 28 – Example of one of the images chosen to test the algorithms.

Figure 29 – Ground truth created for the Figure 28.

Figure 30 - Example of one of the images chosen to test the algorithms.

Figure 31 - Ground truth created for the Figure 30.

The comparison could not be done in a direct way because computing the difference from each pixel of the disparity output to the correspondent pixel in the ground truth image would not be reasonable. The reason why can be rather obvious since the creation of the ground truths was done manually and though not representing the true values of disparity but more of a comparison between each disparity values inside the same image. Because of that, a different metric had to be found to compare, in a more intelligent and wise way, this "hand" made ground truths with their correspondent output results. The metric found and chosen to test this stage was a modified Kendall-tau correlation found in Costa et al. [54]. The Kendall-tau correlation coefficient it is a

statistic used to measure the association between two parameters, in this case, the output of the algorithms and the created ground truth. This statistic takes both images (the disparity output and the ground truth images):

$$disp\_output = \begin{bmatrix} 1,1 & 1,2 & ... \\ 2,1 & 2,2 & ... \\ ... & ... & ... \end{bmatrix} \quad ground\_truth = \begin{bmatrix} 1,1 & 1,2 & ... \\ 2,1 & 2,2 & ... \\ ... & ... & ... \end{bmatrix} \quad (7)$$

where $disp\_output$ is the matrix that represents the output disparity image and the $ground\_truth$ is the matrix that represents the ground truth image. Both images are gray scale format. To run the test easier the matrixes where reshaped to vectors. The comparison to obtain the statistic can be showed by the following pseudo-code:

```
for i < size(disp_output)
        for j < size(ground_truth)
                if ( disp_output(i) - disp_output(j) ) x ( ground_truth(i) -
        ground_truth(j) ) < 0
                        error++;
                j=j+1;
        end;
        i=i+1;
end;
```

Basically, this test runs all pairs of pixels from both images and compares them. If the pairs are consistent (which mean, that if one pair from the output disparity image has the first pixel bigger than the second pixel, than in the correspondent pair of the ground truth image it has to happen the same thing in order for them to be consistent) than it is assumed that the output is correct, if they are not consistent an error is incremented. In the end, this error is divided by all the pairs run and a percentage will give us the indication of the best output algorithm.

The test set is different from the first stage of this evaluation. In the first stage the algorithms run the 25 pair of frames at the standard parameters, in this stage, before testing the evaluation method proposed above, each algorithm will run at different parameters with the objective of finding out also the best combination of parameters for the each one of the algorithms. The parameters that can be changed as inputs of the algorithms are not the same in all of them, but one is, the disparity range. The disparity range is, like the expression suggests, the range of disparities that the output image will have. Naturally, if there is not much depth difference in the image (which means, if there is not much distant difference of the components of the image between each other)

a low range is enough to express the depth levels of the image. In the other hand, if the image is, for example, a portrait of a farm scene, and all of the components of the image are significantly distant from each other, the range of disparity will be more expressed if there are more levels. Remember that when distance is mentioned, it is measured according to the distance of the object to the camera that is taking the photo frame of the scene. So if the distance is being discussed between two objects of an image, the distance to be regarded is the difference of distance between each one of the objects to the camera. The other input parameters are:

Algorithm "Blockset" - The size of the block matching window it is an input of this algorithm, and like is meant, it varies the size of the window that tries to match the center pixel in one of the images with the correspondent pair of frame.

Algorithms "Lankton" and "Region" – The common parameter of these algorithms is the window of the smoothing filter applied. This bigger window, the smaller the detail is, but if it is too small also, might not eliminate the noise that can exist in the output disparity image.

As presented above, there are 2 parameters changed in each of the three algorithms. The algorithms were run with the followed input: 50,100 and 150 as first input corresponding to the disparity range and 3,5 and 7 corresponding to the size of the block matching window in the algorithm "Blockset" and to the size of the smoothing window in the algorithms "Lankton" and "Region". All possible combinations between the inputs of the algorithms were run and then tested in this evaluation. Some examples of the way that these parameters change the disparity map images can be seen in the following Figures:

Figure 32 – Test image used to show an example of the different influence of the input parameters in each algorithm
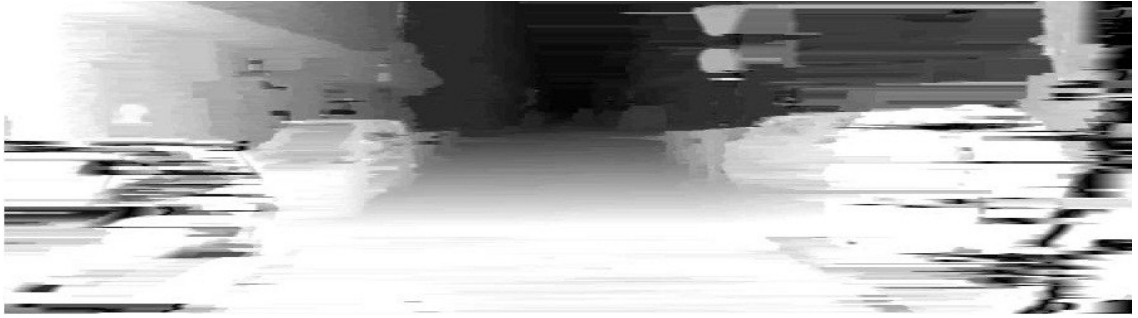
Figure 33 – Resulted disparity map of "Blockset" with the input parameters of 50 for disparity range and 3 for block matching window.

Figure 34 - Resulted disparity map of "Blockset" with the input parameters of 50 for disparity range and 7 for block matching window.

Figure 35 - Resulted disparity map of "Blockset" with the input parameters of 100 for disparity range and 5 for block matching window.
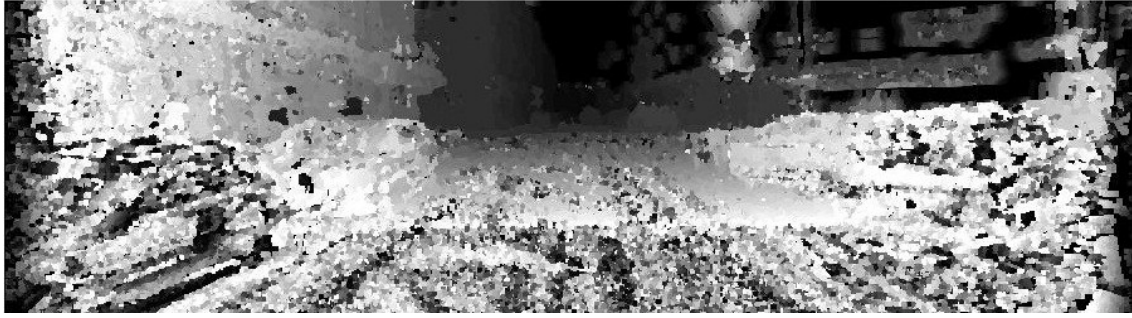
Figure 36 - Resulted disparity map of "Lankton" with the input parameters of 50 for disparity range and 3 for block matching window.
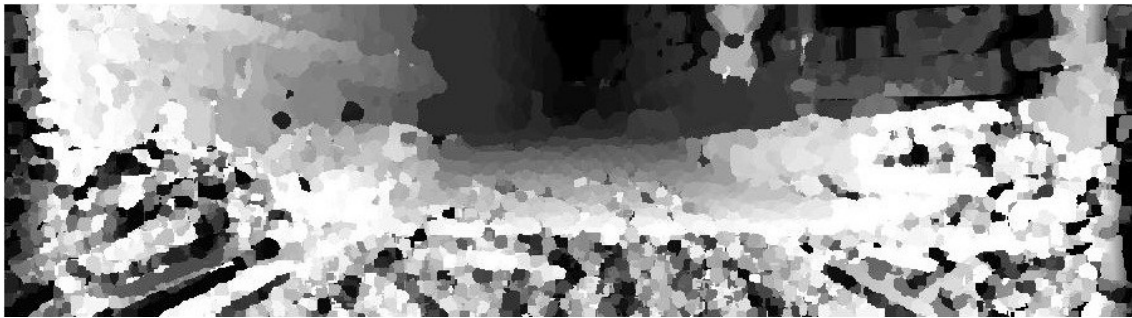
Figure 37 - Resulted disparity map of "Lankton" with the input parameters of 50 for disparity range and 7 for block matching window.
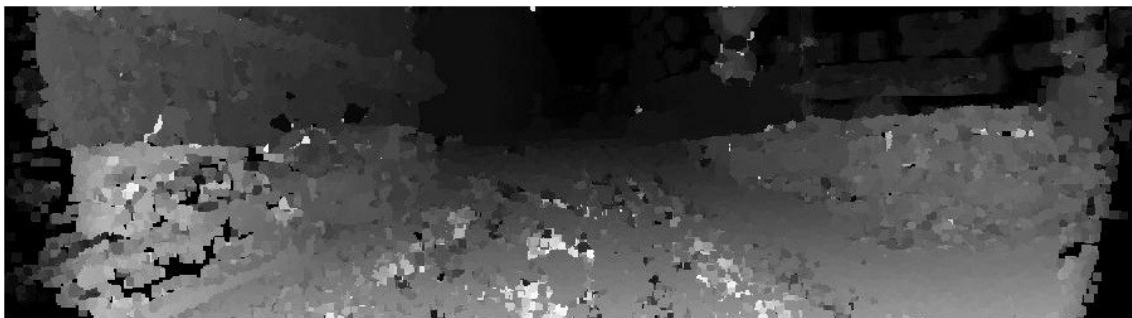
Figure 38 - Resulted disparity map of "Lankton" with the input parameters of 150 for disparity range and 5 for block matching window.
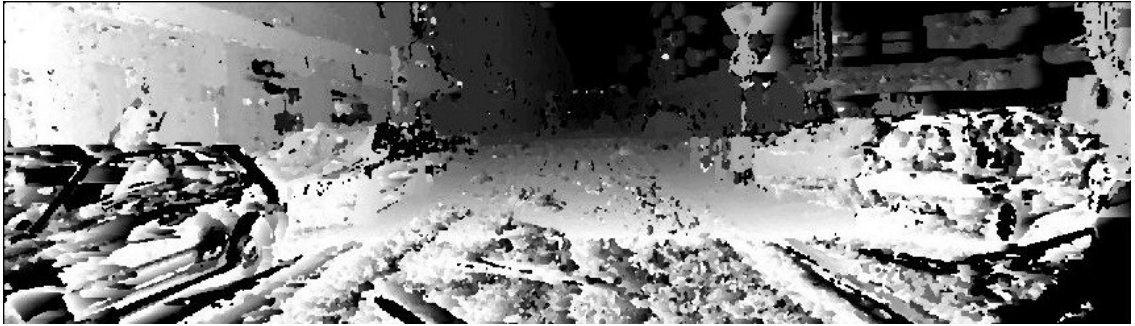
Figure 39 - Resulted disparity map of "Region" with the input parameters of 50 for disparity range and 3 for block matching window.
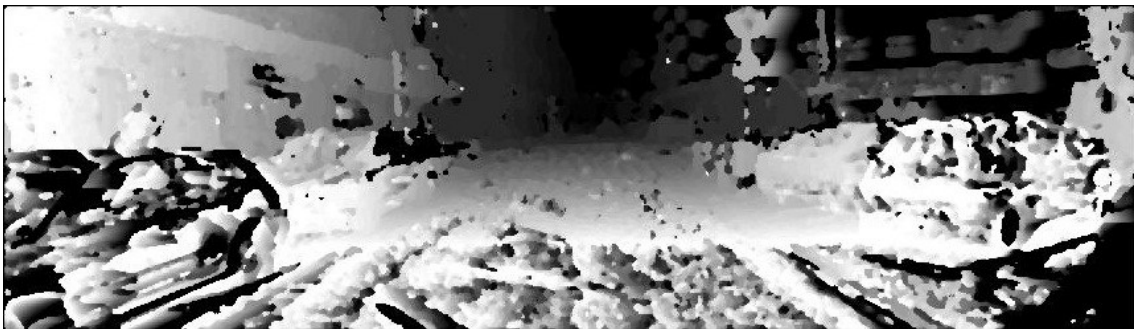
Figure 40 - Resulted disparity map of "Region" with the input parameters of 50 for disparity range and 7 for block matching window.

Figure 41 - Resulted disparity map of "Region" with the input parameters of 100 for disparity range and 5 for block matching window.

The influence of the parameters presented before are easily expressed in the images showed above. Rising up the value of disparity range, the image depth differences of the disparity maps stay, in general, more perceptive. The filtering value influence also can be seen by looking at the smoothness from one disparity map with a filter value and one with a higher one (Figure 36 to Figure 37 and Figure 39 to Figure 40). The influence of the block matching window size does not seem to be conclusive just by looking at the images resulted from the different input values (Figure 33 to Figure 34).

4.4 **Results and Discussion**

In this section the results will be presented divided in the two stages of evaluation. Examples of output images will be shown and also the statistics of the results. The results will be discussed and specific conclusions will be considered.

4.4.1 Results of the First Stage of Evaluation

In this stage, as said in the methodology of this part, an eye evaluation will withdraw the worst two algorithms for the next evaluation. Two outputs will be showed for all the algorithms along with the original image.

Figure 42 - First test image chosen to evaluate the outputs of the algorithms in the first stage of evaluation

Figure 43 - Resulted disparity map of the "Blockset" algorithm
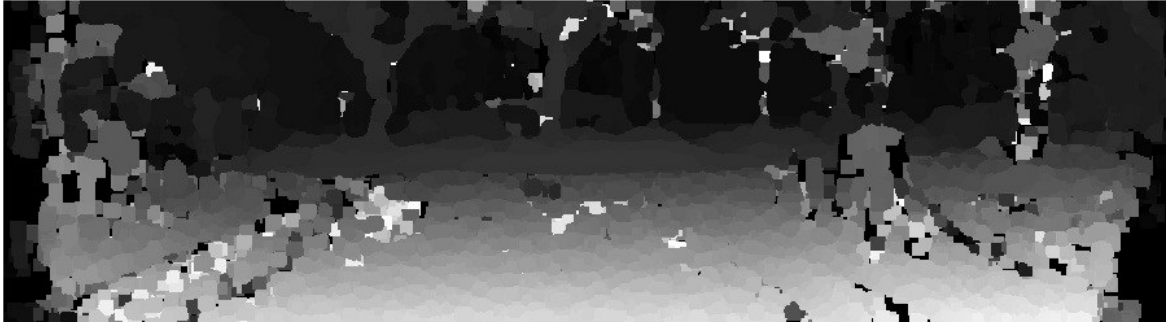
Figure 44 - Resulted disparity map of the "Lankton" algorithm
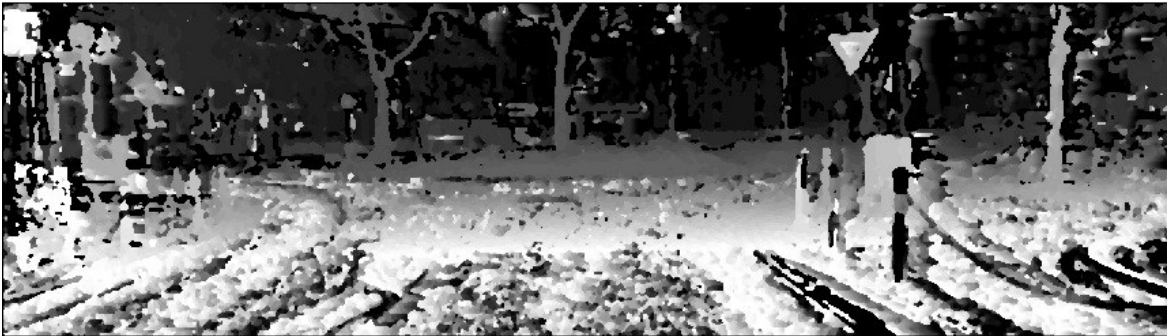
Figure 45 - Resulted disparity map of the "Region" algorithm

Figure 46 - Resulted disparity map of the "Vision" algorithm

Figure 47 - Resulted disparity map of the "Yang" algorithm

The evaluation now can be done by evaluating the depth perception that the disparity images create by their gray intensity levels. If the reader contrasts the original image with all the outputs, it can be said that for this set of output images, it is already clear that the fifth algorithm "Yang" has the worst disparity map of all five algorithms (Figure 47). It is also clear, that first two algorithms, "Blockset" and "Lankton", have the best output disparity map (respectively Figure 43 and Figure 44). The third and fourth algorithms are more difficult to objectively difference them, so in order to choose the algorithm another set of images is presented to the reader to make a point on the chosen decision.

Figure 48 – First test image chosen to evaluate the outputs of the algorithms in the first stage of evaluation

Figure 49 – Resulted disparity map of the "Blockset" algorithm
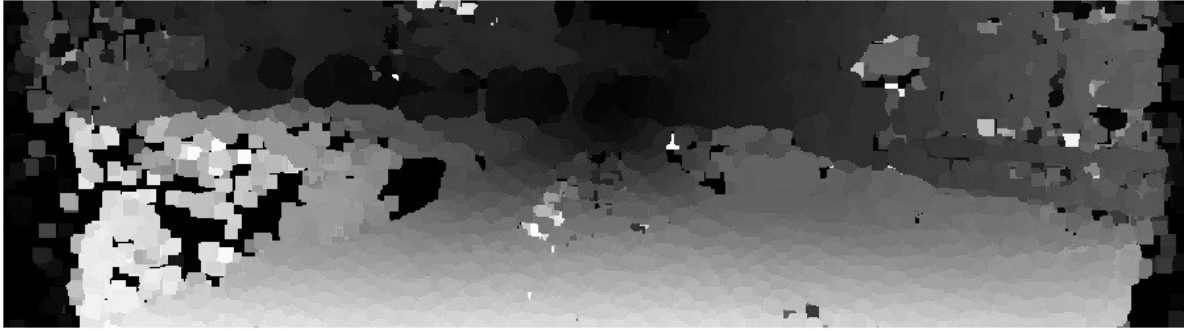
Figure 50 - Resulted disparity map of the "Lankton" algorithm

Figure 51 - Resulted disparity map of the "Region" algorithm
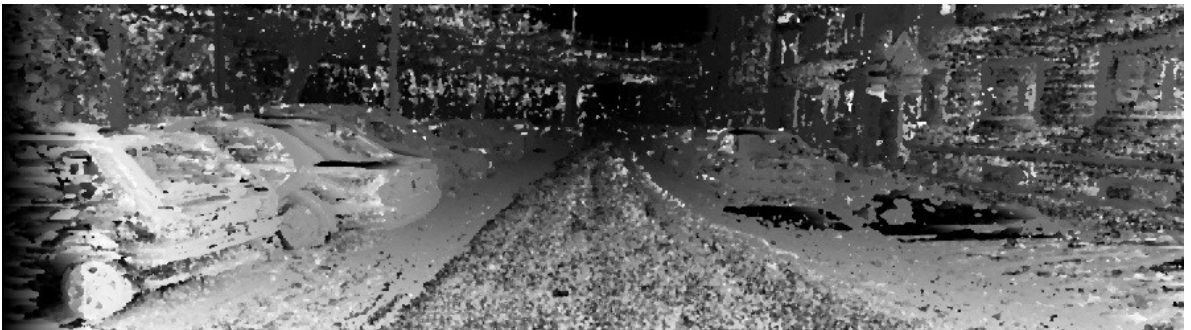
Figure 52 - Resulted disparity map of the "Vision" algorithm

Figure 53 - Resulted disparity map of the "Yang" algorithm

Regarding the comment done about the first set of images, a comparison can now be done between the third and the fourth algorithms. They both have different perspectives of depth but it is difficult to say which one is the best. There is a point though that makes the third algorithm better than the fourth, which is the objective of this part and for this project, which is choosing the best disparity output in order to reduce the search mask for vertical road sign detection. If one observes the signs that are to be detected in both images, the third algorithm seems to have the best clear vertical road signs, and by that it is meant that they stand out more on the third output image. The decision is then to take the first algorithm "Blockset", the second algorithm "Lankton" and third algorithm "Region" to the second part of the evaluation.

4.4.2 Results of the Second Stage of Evaluation

This evaluation run the test set containing the outputs of all the three algorithms chosen in the first evaluation. All the three algorithms took different input parameters, as already explained in the previous methodology of this part, so the test set was considerably bigger than the first test set. The tables below show the results of this part and they represent the percentage of inconsistent pixels in each disparity image. The percentage measure can be considered erroneous if the evaluation only takes an algorithm and tries to understand it separately, but the mainly objective here is to try to compare the results from all three algorithms and reach a fair conclusion for this not so rigorous test.

To help the discussion and comparing the performance of the algorithms other statistics were calculated, such as the difference between the "Blockset" and the "Lankton" algorithms and between the "Region" and the "Lankton" algorithms. Basically, the difference of them was calculated and the results were grouped to a sum of pixels which were below 0 and above 0. If the cells are below 0 it means that the subtracted algorithm has higher inconsistence percentage, if not it means has less. The difference tables pretend show objectively which algorithms perfume better in terms of percentage difference between each other.

Algorithm 1 - "Blockset"

	Parameters								
	50			**100**			**150**		
Frame	**3**	**5**	**7**	**3**	**5**	**7**	**3**	**5**	**7**
19	13.65	16.52	17.96	6.91	6.39	6.22	6.73	5.89	5.48
26	17.45	18.37	19.03	4.15	4.16	4.37	4.15	4.18	4.38
34	8.36	8.81	9.44	5.90	5.75	5.71	5.90	5.76	5.68
77	26.36	22.28	21.30	15.21	11.41	10.11	15.21	11.26	10.10
80	14.25	16.15	16.50	7.73	6.98	6.71	7.73	7.18	6.93
150	14.07	12.99	13.04	5.49	5.41	5.43	5.50	5.42	5.49
160	10.79	10.88	12.04	4.03	3.96	4.02	4.06	4.36	4.46
194	9.72	10.09	12.01	5.84	5.76	5.74	5.84	5.76	5.75
212	13.81	16.81	17.82	5.30	5.09	5.08	5.31	5.10	5.08
250	22.70	25.56	27.31	9.51	9.56	10.91	8.41	8.17	9.64
263	21.27	19.72	20.96	16.02	14.50	15.37	15.44	12.40	12.06
285	15.72	14.83	15.74	5.78	6.35	7.26	5.78	6.36	7.30
322	10.90	12.64	13.62	6.63	6.39	6.25	6.63	6.38	6.23
440	16.95	16.41	17.21	4.25	4.23	4.32	4.25	4.24	4.33
535	20.51	21.21	23.27	13.24	14.2	14.22	12.52	11.77	11.07
592	14.49	16.95	18.88	7.19	7.43	8.37	7.19	7.42	8.48
605	19.28	21.58	22.8	6.29	5.78	5.55	6.29	5.75	5.54
671	15.47	17.3	18.12	7.96	7.54	7.19	8.01	7.68	7.35
820	16.7	18.55	19.82	9.03	8.56	8.23	8.49	7.8	7.41
858	16.26	19.8	21.32	8.12	8.40	8.82	8.12	8.46	8.88
876	18.03	19.61	21.25	7.24	7.18	7.76	7.24	7.24	7.94
1052	13.14	15.02	15.92	7.19	7.16	7.19	7.42	7.38	7.33
1158	23.51	24.53	25.57	11.64	11.72	11.66	8.64	8.31	8.24
1275	31.13	29.79	23.94	26.19	28.36	23.81	25.26	27.23	22.73
2514	22.50	24.07	24.46	6.98	6.94	7.06	6.98	6.94	7.06
Average	17.08	18.01	18.77	8.55	8.37	8.29	8.28	7.94	7.79

Table 2 – Results of stereo matching evaluation of the "Blockset" algorithm. The values represent the percentage of inconsistencies between the disparity maps and the ground truth images.

Algorithm 2 - "Lankton"

	Parameters								
	50			**100**			**150**		
Frame	**3**	**5**	**7**	**3**	**5**	**7**	**3**	**5**	**7**
19	27.36	26.74	26.11	13.45	11.19	10.32	14.46	11.95	11.09
26	26.91	25.02	23.79	14.12	11.25	9.94	14.53	11.40	10.10
34	21.13	20.18	19.67	14.14	11.53	10.71	16.07	12.92	11.82
77	28.99	28.28	27.37	12.12	9.55	8.55	12.62	9.65	8.60
80	27.64	25.95	24.88	16.03	13.36	12.28	17.15	14,00	12.92
150	27.94	26.79	26.36	13,00	10.80	9.93	13.72	11.16	10.23
160	27.57	26.79	26.39	13.32	10.88	9.73	13.78	11.22	10.03
194	27.03	26.85	27.23	14.81	12.95	12.33	15.85	13.39	12.52
212	26.2	25.28	24.48	12.75	11.39	10.83	13.49	11.79	11.13
250	30.86	30.12	29.61	15.67	13.33	12.60	17.10	14.30	13.65
263	28.04	28.23	28.48	22.17	20.87	20.50	22.41	20.89	20.16
285	27.98	27.26	26.89	13.94	11.20	10.06	15.45	12.25	10.78
322	22.39	21.03	20.24	12.18	9.96	9.26	13.25	10.70	9.65
440	25.29	24.15	22.88	8.96	7.57	7.42	9.92	7.91	7.56
535	32.76	32.58	32.26	23.80	21.44	20.05	23.72	20.88	19.25
592	27.19	25.83	25.20	13.28	10.58	9.57	15.35	11.49	10.61
605	29.53	29.22	29.07	11.85	10.51	10.23	12.76	11.06	10.61
671	27.37	26,00	25.13	13.97	12.18	11.30	14.61	12.37	11.37
820	33.52	33.35	33.28	20.19	17.79	16.62	21.95	19.27	18.31
858	26.69	26.15	26.40	13.79	12.13	11.89	14.66	12.61	12.34
876	25.97	25.54	25.06	11.63	10.93	10.91	12.47	11.41	11.26
1052	25.95	25.51	24.95	13.36	12.23	11.95	13.94	12.51	12.16
1158	30.51	30.49	30.44	22.71	21.11	20.43	25.63	24.29	24.10
1275	27.52	25.64	24.43	22.85	19.52	17.89	23.71	19.87	18.09
2514	30.89	29.82	29,00	15.17	11.32	9.60	16.53	12.20	10.36
Average	27.73	26.91	26.38	15.17	13.02	12.19	16.21	13.66	12.75

Table 3 - Results of stereo matching evaluation of the "Lankton" algorithm. The values represent the percentage of inconsistencies between the disparity maps and the ground truth images.

Algorithm 3 - "Region"

Frame	50			100			150		
	3	5	7	3	5	7	3	5	7
19	26.85	26.87	26.90	10.94	10.25	9.71	10.43	9.55	8.90
26	33.01	32.87	32.82	20.62	20.04	19.37	21.74	21.01	20.22
34	18.05	17.49	16.99	10.69	9.67	8.92	11.91	10.78	9.89
77	29.67	29.75	29.92	20.42	19.67	19.19	20.88	20.08	19.61
80	28.78	28.56	28.34	12.68	11.51	10.55	12.96	11.66	10.60
150	29.38	29.38	29.4	14.91	13.45	12.31	16.02	14.47	13.25
160	29.29	29.42	29.53	15.73	14.41	13.36	16.78	15.39	14.31
194	27.69	27.5	27.28	11.90	11.11	10.47	12.88	12.05	11.30
212	26.38	26.37	26.42	12.04	11.53	11.15	13.10	12.53	12.11
250	36.22	36.41	36.53	22.37	21.74	21.14	21.88	21.05	20.29
263	27.12	27.17	27.32	19.01	18.72	18.50	17.09	16.65	16.35
285	27.64	27.46	27.37	16.14	15.30	14.80	17.50	16.64	16.14
322	25.48	25.16	24.86	13.97	13.28	12.73	15.57	14.80	14.16
440	29.24	29.00	28.76	22.17	21.35	20.50	24.99	24.32	23.57
535	31.38	31.47	31.71	20.14	19.57	19.14	17.19	16.48	15.90
592	29.47	29.30	29.08	19.37	18.49	17.78	19.44	18.45	17.63
605	31.46	31.29	31.13	21.38	20.73	20.23	23.14	22.41	21.88
671	27.60	27.30	26.98	13.37	12.36	11.60	15.01	13.84	12.94
820	32.14	32.29	32.53	15.18	14.29	13.60	14.77	13.82	13.06
858	29.18	28.90	28.65	17.80	16.78	15.91	19.19	18.14	17.31
876	28.11	27.81	27.55	17.39	16.32	15.40	18.75	17.58	16.60
1052	26.70	26.38	26.05	14.13	13.28	12.63	15.00	14.04	13.36
1158	32.08	32.23	32.39	19.92	19.41	18.99	14.90	14.15	13.63
1275	27.18	26.78	26.46	23.88	23.27	22.79	23.19	22.44	21.81
2514	30.72	30.65	30.71	19.84	18.88	18.15	21.82	20.85	20.02
Average	28.83	28.71	28.62	17.04	16.22	15.56	17.45	16.53	15.79

Table 4 - Results of stereo matching evaluation of the "Region" algorithm. The values represent the percentage of inconsistencies between the disparity maps and the ground truth images.

	"<0"	">0"
Sum of percentages	-51,62	1574,61
Sum of tests	12	212
Average percentage	-4,3	7,43

Table 5 – Difference table between the stereo matching evaluation results of the "Blockset" and "Lankton" algorithms. The sum of percentages represents the sum of the percentage difference between the both algorithms. The sum of tests represents the total tests that are different between both algorithms. The average percentage is calculated based on the average difference of percentage between both algorithms. The columns">0" and "<0", represent respectively, when the difference between the both algorithms is less or bigger than zero.

	"<0"	">0"
Sum of percentages	-185,41	703,51
Sum of tests	68	157
Average percentage	-2,73	4,48

Table 6 - Difference table between the stereo matching evaluation results of the "Lankton" and "Region" algorithms. The sum of percentages represents the sum of the percentage difference between the both algorithms. The sum of tests represents the total tests that are different between both algorithms. The average percentage is calculated based on the average difference of percentage between both algorithms. The columns">0" and "<0", represent respectively, when the difference between the both algorithms is less or bigger than zero.

By looking at all result tables some conclusions can be made that agree with the eye evaluation. Comparing the algorithms, the "Blockset" algorithm was the one that seemed to have a better depth perception in the first stage evaluation, and here it proves to be also the one that has less inconsistences and outperforms the others as it can be seen by comparing all the tables regarding the average lines. The difference from this algorithm to the other two algorithms it is significant in this test and slightly visible from the first stage. Comparing the "Lankton" and the "region" algorithms, it is more difficult to reach out to an evident conclusion, but still the "Lankton" performs better in the test than the "Region". Although the minimum and the maximum average of the best parameters test (note Table 3 and Table 4 for this) is close, the difference between

them can be better seen in the table 6, where the "Lankton" algorithm has significantly more tests with better results than the "Region" algorithm, reaching to the conclusion that "Lankton" wins "Region". The Table 6, comparing "Lankton" with the "Blockset" algorithm leaves no doubt of the best algorithm between all of them. The reader can also notice the best and worst average from each algorithm, represented respectively by green and red colouring in the last line of each table, and reach the same conclusions as previously presented.

Some conclusions can also be made upon the results between the input parameters of each algorithm and their influence in the best disparity image (follow the Tables 2-4 within this paragraph). The "Blockset" algorithm seems to improve every time that the disparity range grows, which is expected because the detail for the disparity map grows. In terms of block matching window size it seems to also improve slightly when the window is bigger, instead of in the disparity range 50 which can be considered as not conclusive concerning the validity of this test. For the "Lankton" and "Region" algorithms the disparity range 100 seems to be better for both of them. The explanation of this might be in the fact that if it is too big, the matching window has to search for the best pixel broadly in the image, generating more mismatches. Then again, the test is not strong enough to withdraw these conclusions that can be called somehow mistrustfully. In terms of the smoothing window size input parameters, the bigger the size, the better the output, which can be explained by the extension of the error that is done by creating only 5-10 levels of intensity in the ground truth images, which can favour the smoothing process by eliminating detail but keeping the high scores in this test. In result of this evaluation, the last three algorithms and all their different outputs will be used in the next part of the project to try and confirm the true validity of this test.

The tests were run in a PC Intel Core 2 Quad @2.5GHz using Matlab® R2011a. The time spent by the algorithms to generate the disparity maps was not taken much into consideration but still the results are showed in the Table 7 and succinct comparison can be made. The "Blockset" algorithm takes clearly more time to run the stereo pairs than the other algorithms. The "Lankton" algorithm is the one with less time consumption. All the algorithms waste more time if the disparity range is raised and also a slight increase in time is noticed if the second parameter (block matching window size in the case of the "Blockset" algorithm and filtering parameter in the others case) are also raised.

The time spent on each evaluation done with a ground truth image and the correspondent disparity map was also calculated in between 2 hours and 40 minutes to 3 hours (depending on the inconsistencies found). As it is noticed this evaluation method is very time consuming and not so efficient in terms of prove for comparison

evaluations. It is important to say that these codes were implemented in Matlab but if it would be in C++ a significant decrease of time consuming would be verified, making this tests more potentially more suitable for real-time applications and also faster in off-line applications.

Algorithm	Parameters		Time
Blockset	**50**	**3**	11-13 minutes
	100	**3**	21-23 minutes
	150	**3**	28-30 minutes
	50	**5**	12-14 minutes
	100	**5**	20-22 minutes
	150	**5**	30-32 minutes
	50	**7**	13-15 minutes
	100	**7**	22-24 minutes
	150	**7**	32-34 minutes
Lankton	**50**	**3**	~30 seconds
	100	**3**	~1 minute
	150	**3**	~1,5 minutes
	50	**5**	~30 seconds
	100	**5**	~45 seconds
	150	**5**	~1,4 minutes
	50	**7**	~30 seconds
	100	**7**	~1 minute
	150	**7**	~1,5 minutes
Region	**50**	**3**	3-4 minutes
	100	**3**	8-9 minutes
	150	**3**	10-11 minutes
	50	**5**	3-4 minutes
	100	**5**	6-7 minutes
	150	**5**	10-11 minutes
	50	**7**	3-4 minutes
	100	**7**	6-7 minutes
	150	**7**	9-10 mins

Table 7 – Time consumption of each algorithm concerning their input parameters.

Chapter 5

Reduced Search Mask

This experimental part intends to reduce the search mask of the vertical road signs. With the objective of doing that, the 25 previous frames are going to be used along with more 20 frames that contain none, one or more vertical road signs. The disparity images used are the result of the output of the three algorithms, to the chosen frames, selected in the first experimental part. Some morphological methods will be used to reduce the mask along with other image processing techniques. The results will show the percentage of image that can be reduced, with the techniques used, and off course keeping all the vertical road signs inside the mask.

5.1 Methodology

The methodology thought to complete this part of the project was mainly based on using basic image processing techniques and arrange them in a way that the percentage of mask reduced is considerable. The goal of reducing the mask comes from an inner objective which is detecting the vertical sign in the first place. The vertical sign has a unique property in the disparity image, that can be easily observed, that is the fact of the sign keeping the same intensity value (matrix pixel value). This phenomenon can be seen by looking at the original image in the Figure 54 and the correspondent disparity map image in the Figure 55; both have the signs circled with a red line.

Figure 54 – Test image showing two road signs circled with a red line

Figure 55 – Disparity map from the image shown in the Figure 54 with the two road signs circled with a red line

This happens because the way that the vertical road signs are place is "face to face" to the camera, which will reflect to the sign having the same depth intensity in all of its object description in the disparity image. This property of the sign was the one chosen to reduce most part of the search mask. First of all, many operations were thought and evaluated in a hypothetical way to extract in an efficient way the areas of the disparity image were the intensity it is the more less the same. Such operations would have to find these areas, that in same way are represented as sudden changes of intensity values and be separated by, for example, edge detectors, among other techniques. Having reasoning the possible and most efficient path to fulfil the goal of this part a procedure of operations was done and it is presented below in the order that it is applied (in a programming point of view).

XZ-plane Transposition – This operation consists on transposing the xy-plane of the disparity map image in which the pixel intensity is represented in the matrix value of the corresponded (x,y) position to a xz-plane, creating then a density image. This operation is better described in Cardoso et al. [55], but basically, when transposed, the value at the position (x,z) of the density image denotes the number of points in the depth image at the position x. This operation can be better understood if one looks to the original disparity image example in the Figure 56 and in the resulting

transposed image represented in the Figure 57. The formula applied in this transposition is:

$$d(x,z) = \sum_y \delta(D(x,y) - z) \quad \text{with} \quad \delta(n) = \left\{ \begin{array}{l} 1 \; if \; n = 0 \\ 0 \; otherwise \end{array} \right. \tag{8}$$

where $D(x,y)$ is the depth information value at the position (x,y) and $d(x,z)$ is the value of the density image at the position (x,z).

Figure 56 – Example of a ground truth [55].

Figure 57 – XZ-transposition image of the Figure 56 [55] (different colours represent different objects).

Median Filtering – A median filter is pre-processing image operation that is commonly used to reduce the noise. The median filter, in image processing, works by passing a window through all the pixels of the image and choosing the median value of that window, replacing it in the centre pixel of the window chosen. The illustration of this operation can be seen in the Figure 58 and an example of the result of applying it in the Figure 59. The Matlab function medfilt2 is a 2-D median filter that works for image processing and has an input value of the image to be processed and the size of the window to be considered. It is reasonable to say that, the bigger the considered window is, the more averaged will be the output image, and more detail will lose, so a balanced window has to be chosen. The median filter is to be applied in the density image and its objective is not to withdraw the noise but to eliminate low density areas before the next

step, the thresholding. Three sizes of window will be considered and tested, to see the implication of this operation before thresholding. Their values are 0, 1 and 2.
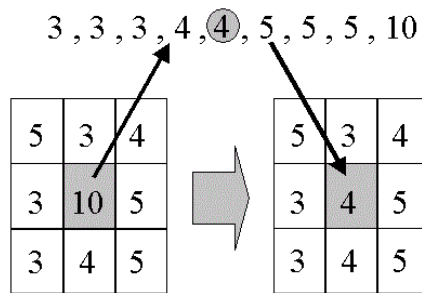
Figure 58 – Illustration of the median filter operation
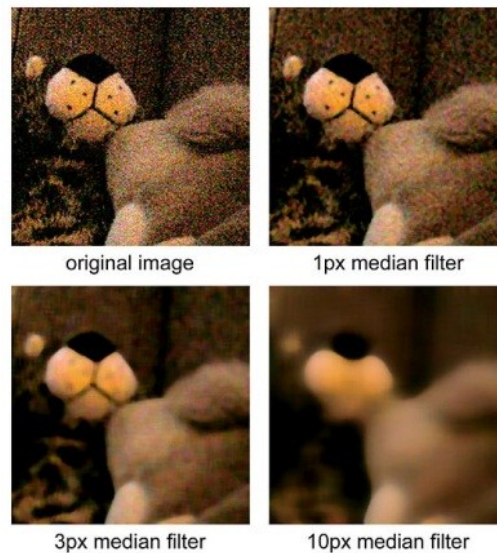
Figure 59 – Example of the median filter application and the influence of its window size

Close and far intensity cuts – This operation takes out the really closest values where vertical road signs cannot be present, and really far away intensity values where signs are not present also. For this operation the values choosen were to cut the 10% of the low intensity values and 20% for the farest away values. The operation basicly takes the density matrix and for lines bigger than 255 minus 10% of 255 every matrix value becomes zero, and for lines smaller than 0 plus 20% of 255 the matrix values also become zero.

Thresholding - Thresholding is one of the simplest image processing operations and used mainly in image segmentation. The simplest way to apply it and the one also used here is by putting the image pixel values to zero if they are below a certain value (cutoff value). It is represented by the Equation 9, where th is the threshold cutoff

value $d(x,z)$, is the density image coming from the previous operation and $d_{th}(x,z)$ is the threshold binary image:

$$d_{th}(x,z) = \begin{cases} 1 \ if \ d(x,z) \geq th \\ \quad 0 \ otherwise \end{cases} \tag{9}$$

As said in the equation, the threshold is applied to the density image in order to keep the areas that have strong pixel intensity repeatability. The result of this operation is a binary matrix that represents those areas. The threshold cutoff values chosen to test this operation were 10, 15 and 20.

XY-plane Transposing – This operation represents the returning path of the xz-transposition done in the xz-transposition step. back to xy-plane that represents the image (see Figure 60 as an example). This transposition, like the reader can easily understand, is impossible to do in perfection since some information was lost by the disappearance of the y-plane and by using the median filter and the thresholding operations. The most accurate transposition is done by going through the binary matrix outputted in the thresholding step and for every pixel value that has the value true mapping it in a new image. This matrix value that is true, is going to be transposed to the new matrix, keeping the same x value but putting in the new matrix all the pixels of the original image that have the same pixel intensity corresponding to the value z of the binary image. This operation can be also showed by the equation:

$$I(x,y) = \begin{cases} O(x,y) \ if \ d_{th}(x,O(x,y)) = true \\ \quad 0 \ otherwise \end{cases} \tag{10}$$

where $I(x,y)$ is the new image, that has the same size (in resolution and format) as $O(x,y)$. $O(x,y)$ is the right image of the input pair of frame and $d_{th}(x,y)$ is the binary thresholded density image. The new image $I(x,y)$ represents that same as the density image, but now in the original view xy-plane.
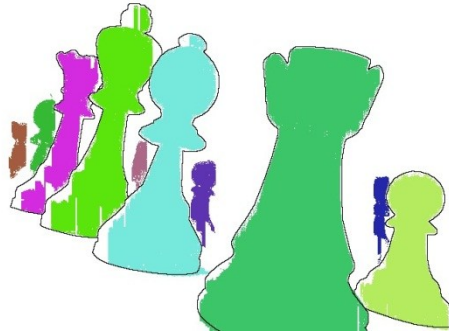
Figure 60 – XY-transposition using the XZ-transposition image shown in the Figure 57 [55].

Shape Selection – The result of the last processed image $I(x, y)$ is going to have some geometric forms that can be eliminated because they do not represent the possibility of being a vertical road sign. The example of that are lines, dots, and other strange geometric forms. This operation eliminates some of those forms by only keeping an area inside the image intersects completely the square shape of size $\boldsymbol{n}\mathbf{x}\boldsymbol{n}$. An example of this operation can be seen in the Figure 61, where can be seen 4 stages of this operation. The sizes tested will be 4x4 and 6x6.

Figure 61 – Illustration of the shape selection operation. The red block represents the shape window 4x4 pixels, the black block the image to be applied the operation and the green point a pixel to get the sense of dimension. Four stages can be seen, the initial stage, the next stage where the shape window eliminates part of the image, the third stage that again eliminates a form that does not fit inside the shape window block and the final result of the operation.

Region Growing – After the shape selection there is a necessity to grow the areas that might have vertical road signs for them to fit all inside them. This operation simply takes a pixel and grows it around with a matrix 3x3. The parameters run ins this operation are 4,6 and 8, and they represent the times the operation is run, i.e., the times that the matrix 3x3 is used in all pixels. This operation can be showed by the example Figure 62. The final matrix obtained in this operation is the final reduced mask and it is this image mask that it is going to be tested.
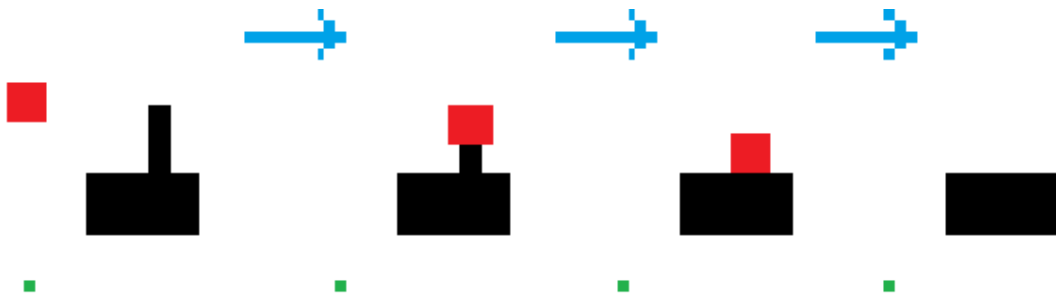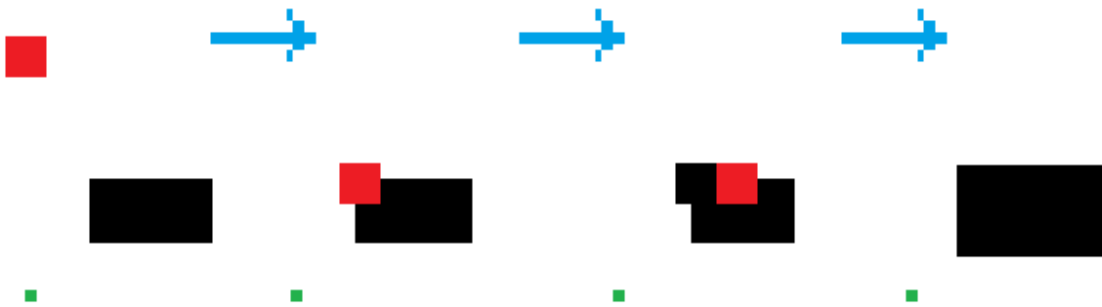
Figure 62 - Illustration of the region growing operation. The red block represents the growing window 3x3 pixels, the black block the image to be applied the operation and the green point a pixel to get the sense of dimension. Four stages can be seen, the initial stage, the next stage where the grow window centres itself in the left superior pixel and grows its area, the third stage that again repeats the process to other pixels and the final result of the operation, a bigger block (more accurately, with 1 more pixel in each dimension).

All the steps that were explained can be gathered and represented by the following flow diagram:
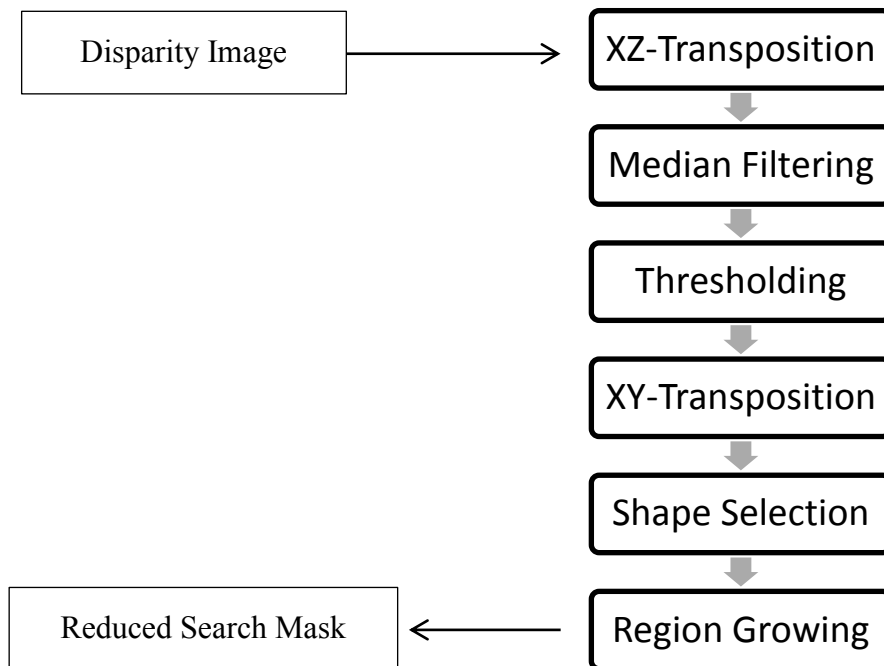
Figure 63 – Flow diagram representing the reduced search mask procedure

5.2 Results

In this section intermediate results will be showed for better understanding of the final results. The intermediate results will be basically the outputs of the operations that are successively executed until the final reduced search mask. The final results are the evaluation of the statistics obtained after running the tests for various input parameters of each operation.

5.2.1 Intermediate Example Results

The original image used in this example is showed in Figure 64 and the correspondent disparity image is showed in the Figure 65. The algorithm for the disparity image is the first algorithm "Blockset" with the inputs 150 for disparity range and 7 for block matching window.

Figure 64 – Test image used to show the results for the reduced search mask

Figure 65 – Disparity map of the image in the Figure 63 (the "Blockset" was used to give as an example of the results obtained).

The result of applying the XZ-transposition is a density image that looks like Figure 66.
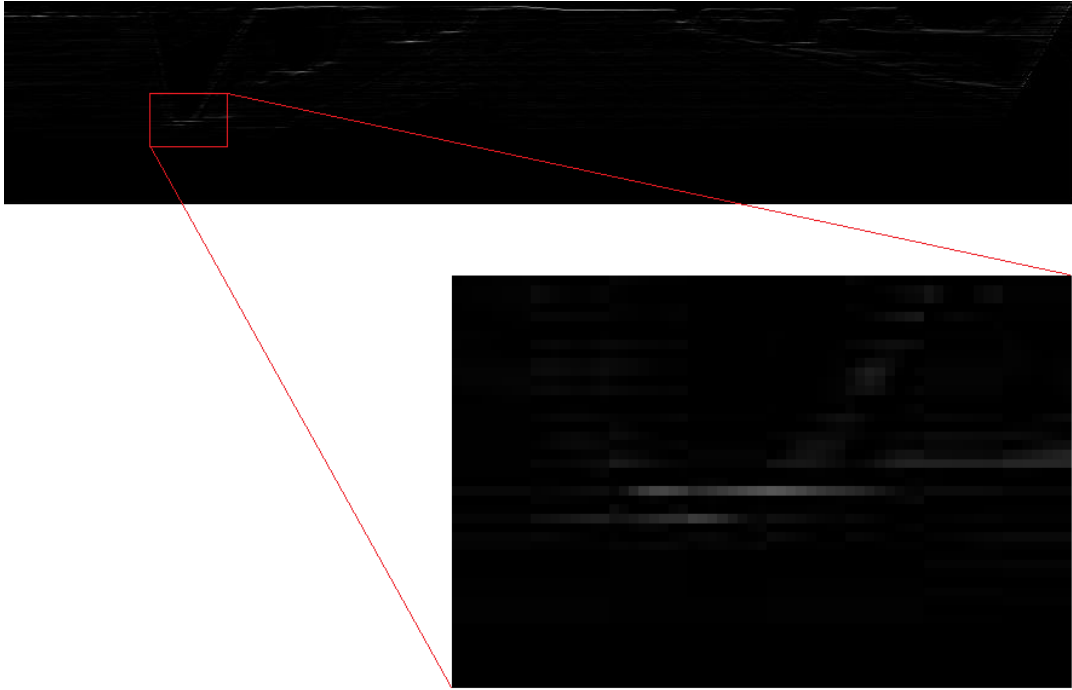
Figure 66 – XZ-transposition and an example segment of the density map.

After applying the XZ-transposition it is the time of the median filtering, if this filter is applied with a 2x2 window, the output image is showed by the Figure 67. The unfiltered segment in the Figure 66 can then be compared with the resulting filtered image segment in the Figure 67.
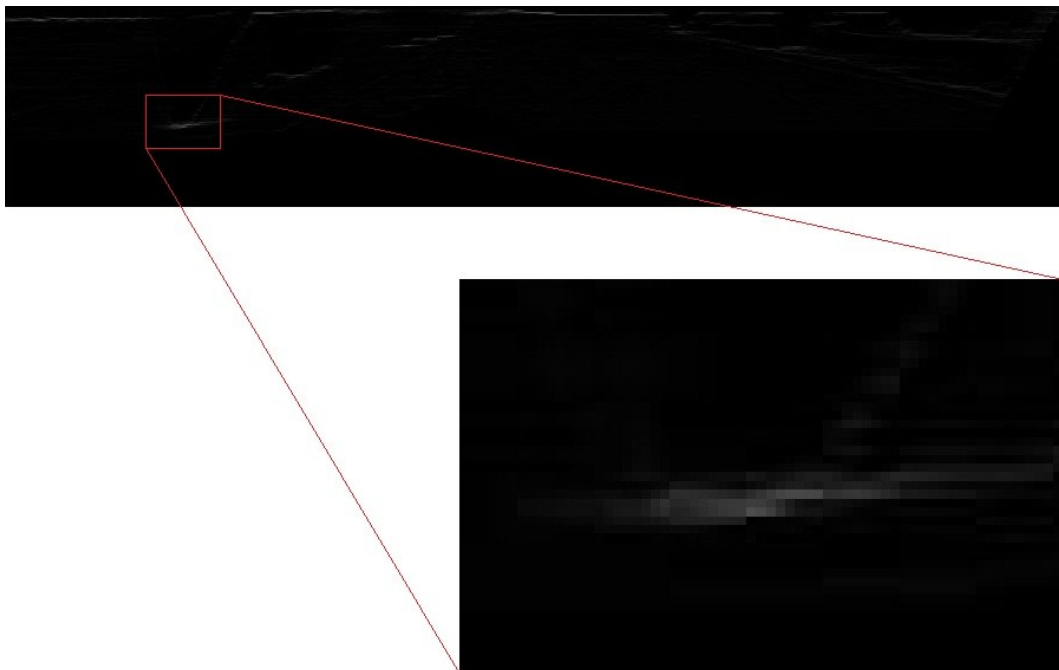
Figure 67 – Median filter application to an example segment shown in the Figure 67.

Depending on the density image is filtered, or not, and after the intensity cuts explained in the methodology section, the threshold will cut more the size of the mask, or not, but the objective of it can still be showed by the Figure 68 (a binary image of the density image). The result image also depends on the value of threshold also applied the mask making it be also smaller or bigger.
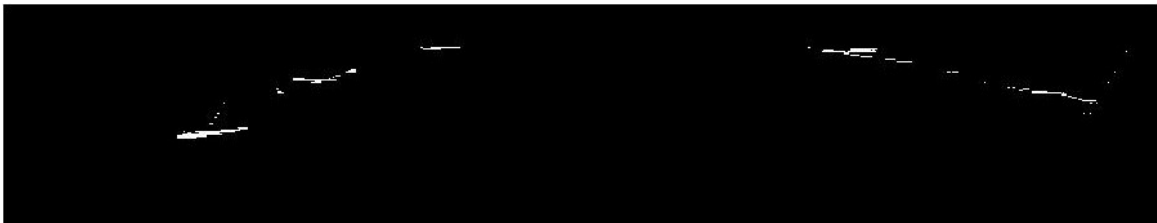
Figure 68 – Binary image after applying the threshold to the image shown in the Figure 67

The next step is the XY-transposition; this operation can be showed by the example of the following Figure 69 and an amplified segment of the resulting image:

Figure 69 – XY-transposition based on the binary image in the Figure 69 and the original image in the Figure 64. An example segment is shown to illustrate further processing steps.

The shape selection also filters same parts of the image and also depends of the effect of the previous parameters chosen in the previous operations of course.

Nevertheless, a segment of an image can illustrate the execution of this operation. The Figure 70 shows the effect of this operation (look at the segment of the previous Figure 69 and compare it with the following segment) with the input parameter of 4, creating a 4x4 shape window.
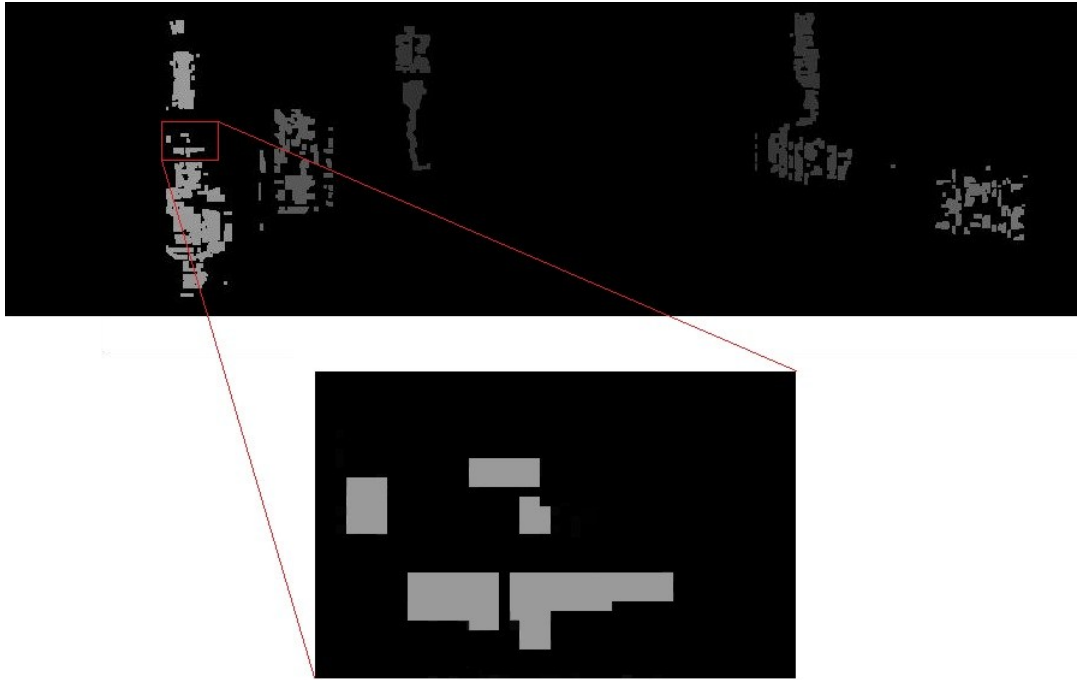
Figure 70 – Shape selection using the image shown in the Figure 69. The image segment illustrates better the operation done from the previous look of the segment in the Figure 69.

The last operation of this part has the main objective of fitting the vertical road signs into the masks. As said before, and since this is the last of all operations, depends of all of the last ones, but an image segment can still be showed by the Figure 71 (once again, look at the previous image segment in the Figure 70 and compare it with this segment). The red box represents the segment that has been being analysed till the end of this example.
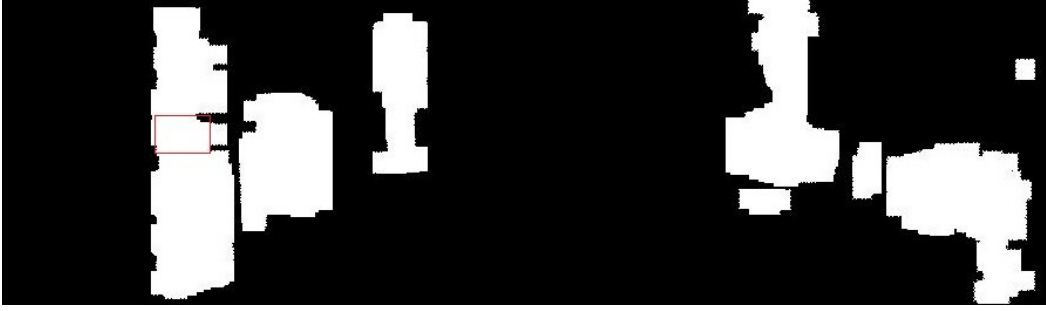
Figure 71 – Binary image after applying the region growing operation. The red rectangular represents the segment that has been show in the previous images.

The Figure 72 can show that the two existent signs rounded with a red circle in the original image (Figure 64), are inside the mask, since the image represents the application of the mask in the Figure 71 to the original image.
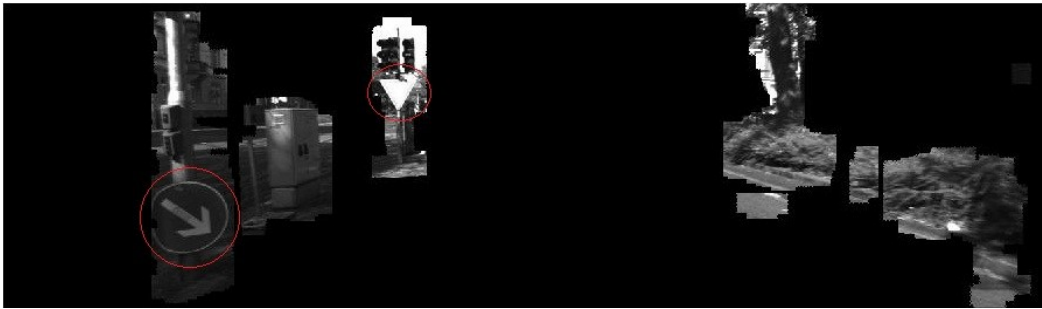
Figure 72 – Resulted mask after applying the binary image shown in the Figure 72 with the original image shown in the Figure 65.

5.2.2 Final Results

The final tests were obtained after running an extensive set of input parameters. All possible combinations were done with the followed set of input parameters:

- Median Filter: 0 and 2 (meaning 0 that no filtering was done);
- Threshold: 10,15, and 20;
- Shape Selection: 4 and 6;
- Region growing: 4,6 and 8;

In total 486 tests were ran with two variables to evaluate, which were the percentage of signs included in the mask and the size of the mask compared with the original image. The tests were run with 45 different stereo frames with 51 different vertical road signs. The objective here is to have all the signs inside the mask and at the same time

the smallest search mask. The statistics results are difficult to display and because of that only the more important results will be display and discussed. To explain it better the syntax "algorithm_DisparityImage_MedianFilterParameter_ThresholdParameter_ ShapeParameter_RegionGrowParameter" will be used (example: blockset_2_2_10_4_ 4). The following table shows the code for the disparity images interpretation:

Disparity Algorithms Parameters		Code
50	**3**	1
	5	2
	7	3
100	**3**	4
	5	5
	7	6
150	**3**	7
	5	8
	7	9

Table 8 – Disparity algorithms input parameters code.

The Table 8 is used to facilitate the interpretation of the results to the reader. The final results of this part are presented in three separate tables; one for the results of the method using the "Blockset" algorithm disparity images, other for the "Lankton" disparity images and another one for the "Region" disparity images. Each table has 9 different results that were chosen upon a certain criteria. First, three scenarios are chosen, having zero, one and two signs missed. In each of these scenarios, the results are present according to:

Maximizing the average percentage of fitting of the mask to the signs in the images (in other words, if the signs are totally inside the mask, and if they are not, which percentage it is). In this restriction, the result that has bigger average is represented in the table along with the correspondent parameters used.

Minimizing the average mask size with the objective of finding the parameters that best minimize it and to which average sign it corresponds.

Find the parameters with the result of average sign bigger than 95% and see to what average mask size it corresponds. This approach was intended to give the reader a more balanced output between the average sign detection and the average mask size.

The combinations of parameters and the results are represented along the lines of the table. The equal results column represents the total combinations of input parameters that fulfil the restriction represented in the first three columns of the table. Remember

that the disparity images also represent in this case an input parameter of the algorithm that comprises all the operations in this part.

The time spent in each detection run using a disparity map and a set of chosen input parameters was of average of 6 to 7 seconds in a PC Intel Core 2 Quad @2.5GHz.

Now it is time to look at the tables and withdraw some conclusions based on them.

Algorithm "Blockset"			Disparity Image	Paremeters				Results		
				Threshold	Grow Region	Shape Selection	Median Filter	Average Mask (%)	Average Sign (%)	Equal Results
Signs Missed 0	Maximizing	Average Sign	7	4	6	10	0	39,14	99,37	4
	Minimizing	Average Percentage	4	4	4	10	2	32,67	96,11	2
	>95%	Average Sign	4	4	4	10	2	32,67	96,11	2
Signs Missed 1	Maximizing	Average Sign	8	6	4	10	2	34,14	99,01	6
	Minimizing	Average Percentage	6	4	4	20	2	22,91	86,62	4
	>95%	Average Sign	5	4	4	15	2	27,56	95,66	2
Signs Missed 2	Maximizing	Average Sign	9	4	4	10	2	29,74	99,11	2
	Minimizing	Average Percentage	8	4	4	20	2	19,75	87,5	2
	>95%	Average Sign	8	4	4	15	2	24,13	96,35	4

Table 9 – Chosen results to represent the "Blockset" disparity maps performance on the reduced search mask procedure.

In the Table 9, it is represent the results for the disparity images from the "Blockset" algorithm and let us discuss the influence of each parameter in the final results:

The disparity map images that have best average sign percentage are the ones with disparity range of 150 but one the other hand the range of 100 seems to minimize the size of the mask. Since the objective is to keep all the signs inside the mask, the disparity images with best results for this have the disparity range of 150. The block matching window size varies in a way that it is not conclusive or discussable to talk about the influence of it in the results;

The median filter seems to be an operation required in the all of the restrictions used, as in the maximizing of the average sign and minimizing of the average mask size it gives the best output results (minus the first line of results);

The threshold seems to influence mainly the average mask size. In the first lines of results the threshold is 10 because there must be no signs missed, so the threshold is low and a result of that is the average mask size being so high. Once one or two signs are missed, the threshold can be higher and a consequence of that is the average mask size diminish. Between missing one or two signs it is also clear the influence of the threshold in the result of the size of the mask since if the average sign is maximized the threshold must be lower than if the average mask size is minimized. The influence of the threshold can then be summed up to be more important in the reducing of the mask size. Other conclusion that can be made is that if the efficiency of the results are lowered, or in other words, if it is accepted to miss two signs (meaning reduce the detection from 100% to more or less 96%) the threshold can be higher and the mask size considerably reduced. It is also important to say that the failure in the detection of 1 or 2 signs can happen because the sign is too distanced from the camera or too close, making it more possible to be detected in the previous or forward frames.

The growing region parameter and the shape selection parameter are parameters that are difficult to analyse. In the results the value 4 it is enough for all the restrictions, but since the scenario of non-using this methods was not tested here it is not conclusive to say that they are essential. Never the less, as said before in the methodology section, various tests were done to choose the parameters to analyse, so it is safe to say that they are important to reduce the average mask size, mainly because of the way that they work. The shape selection clearly eliminates areas of the mask that are not important (showed in the intermediate results part), and the region growing method just grows the areas to fit the signs in the mask. The equal results also prove that by rising the values of this parameters not much is changed in the average sign or in the average mask size, since the repeated results exist the rising of those parameters keep those values still.

The Table 10 and Table 11 show the results for the disparity images of the "Lankton" and "Region" algorithms. The N/A cells, that represent the not applicable abbreviation, appear because for the restrictions used there is no result that fulfils them.

Algorithm "Lankton"			Parameters					Results		
			Disparity Image	Threshold	Grow Region	Shape Selection	Median Filter	Average Mask (%)	Average Sign (%)	Equal Results
Signs Missed 0	Maximizing	Average Sign	7	8	4	15	0	67,88	99,07	4
	Minimizing	Average Percentage	5	4	4	20	2	21,24	62,52	2
	<95%	Average Sign	5	8	4	10	2	54,33	96,94	2
Signs Missed 1	Minimizing	Average Percentage	5	4	4	20	2	18,88	66,41	2
	Maximizing	Average Sign	5	8	4	15	0	67,88	99,07	4
	<95%	Average Sign	9	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Signs Missed 2	Maximizing	Average Sign	N/A	8	4	15	2	31,37	79,19	2
	Minimizing	Average Percentage	7	4	4	20	2	16,98	59,06	2
	<95%	Average Sign	8	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 10– Chosen results to represent the "Lankton" disparity maps performance on the reduced search mask procedure.

Algorithm "Region"		Disparity Image	Paremeters				Results		
			Threshold	Grow Region	Shape Selection	Median Filter	Average Mask (%)	Average Sign (%)	Equal Results
Signs Missed 0	Maximizing — Average Sign	7	6	4	10	0	69,82	99,45	4
	Minimizing — Average Percentage	6	6	4	10	2	38,85	93,96	2
	>95% — Average Sign	5	8	4	10	2	44,86	95,66	6
Signs Missed 1	Maximizing — Average Sign	5	6	4	10	0	60,11	99,31	6
	Minimizing — Average Percentage	8	4	4	10	2	27,54	89,81	2
	>95% — Average Sign	9	6	4	10	2	38,61	95,69	2
Signs Missed 2	Maximizing — Average Sign	6	8	4	20	0	35,96	98,21	4
	Minimizing — Average Percentage	7	4	4	20	2	14,91	68,58	4
	>95% — Average Sign	8	6	4	20	0	30,36	96,05	8

Table 11– Chosen results to represent the "Region" disparity maps performance on the reduced search mask procedure.

The results presented in Tables 10 and 11 also sustain in general the same influence of the parameters commented above but with a more irregular pattern. It is clear that, by

looking at all of the tables, the "Blockset" disparity images have a better performance once applied all of these operations. The "Lankton" images have in general lower average mask size when the last is minimized but the comparison between those results and the "Blockset" results show that the average sign is much higher for the "Blockset" results, meaning that the results lower the mask, but the signs fit very poorly inside the masks.

To have a better understanding of this results, some image masks obtain after running this set of operations are showed above. It can be helpful to look at the result tables and the next Figures at the same time to follow the discussion. Two frames were chosen and some results are displayed to illustrate the comments and conclusions done so far.

Figure 73 – Example of a binary image mask outputted from the set of reduce search mask operations.

The Figure 73 represents the binary search mask of the result Blockset_7_0_10_4_4 to the frame 885 and intends to give the reader an example of the output of this set of operations. In the next Figures the mask will be applied to the original image in order to see clearly the effects of it in the image and its search for signs.

Figure 74 – Reduced search mask of the frame 885 resulting from the input parameters of Blockset_7_0_10_4_4.

Figure 75 - Reduced search mask of the frame 1248 resulting from the input parameters of Blockset_7_0_10_4_4.

The Figures Figure 74 and Figure 75 are both outputs of the parameters Blockset_7_0_10_4_4. This input has zero signs missed and 99.37% of average sign fitting. It is visible that in the both chosen frames the signs are inside the mask. The downsize of this set of input parameters, as already said before, is the size of the mask 39.14% which is still high.

Figure 76 - Reduced search mask of the frame 885 resulting from the input parameters of Blockset_9_2_10_4_4.

Figure 77 - Reduced search mask of the frame 1248 resulting from the input parameters of Blockset_9_2_10_4_4.

In the Figures Figure 76 and Figure 77, which represent the outputs of Blockset_9_2_10_4_4, some changes can be observed. Even though the average mask size doesn´t change a lot from the previous analysed figures, the Figure 76 has a sign which doesn´t fit completely in the mask. This might happen with some masks and the

one represent in the Figure 76 is one of them (between 51 different masks for each result). This result has 2 signs missed and it reduces 10% of the search mask for 29.74%, as it is visible if the reader compares the set of Figures Figure 74 and Figure 75 with the set Figure 76 and Figure 77, keeping almost the same average sign percentage (even though the Figure 76 doesn´t show it).

Figure 78 - Reduced search mask of the frame 885 resulting from the input parameters of Blockset_8_2_20_4_4.
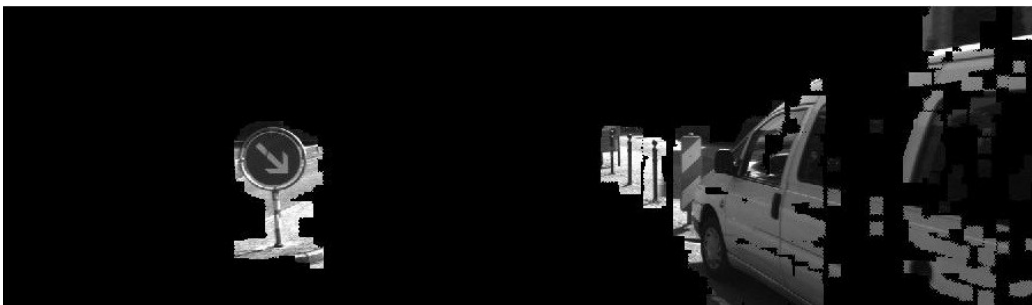

Figure 79 - Reduced search mask of the frame 1248 resulting from the input parameters of Blockset_8_2_20_4_4.

In this last to Figures, Figure 78 and Figure 79, the mask sign is minimized and it represents the input parameters Blockset_8_2_20_4_4. In these results the average mask size goes down 10% from the previous analysed set of mask (this can be seen by looking at the pair of Figures Figure 76 and Figure 77 and the above two Figures Figure 78 and Figure 79). The big downsize is that the sign percentage reduces a lot, from 99.11% to 87.50%, which can be illustrated by looking at the Figure 79.
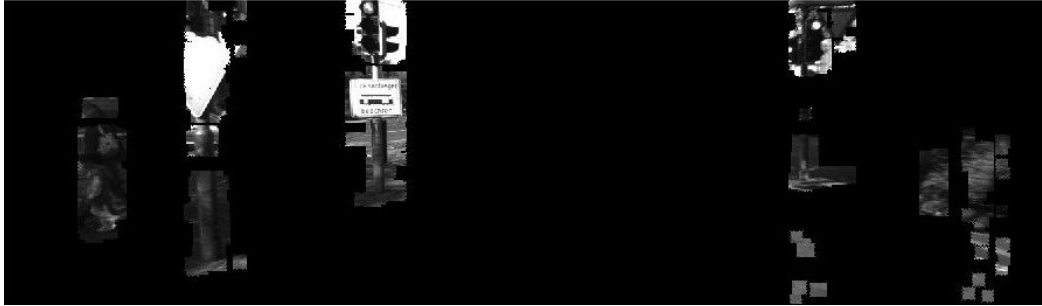
Figure 80 - Reduced search mask of the frame 885 resulting from the input parameters of Lankton_5_0_15_4_8.

Figure 81 - Reduced search mask of the frame 1248 resulting from the input parameters of Lankton_5_0_15_4_8.

The Figures Figure 80 and Figure 81 are an example of the input Lankton_5_0_15_4_8 and represent the maximizing of the average sign fitting. It is clear that by looking at these Figures that the mask is really big proving the average mask size of these results of 67.88%. Comparing this output results with the output results of Blockset_7_0_10_4_4 the average mask size is considerably reduced from 67.88% to 39.14% having the more less the same average sign percentage. The "Lankton" inputs are shown to have worst results than the "Blockset" inputs.
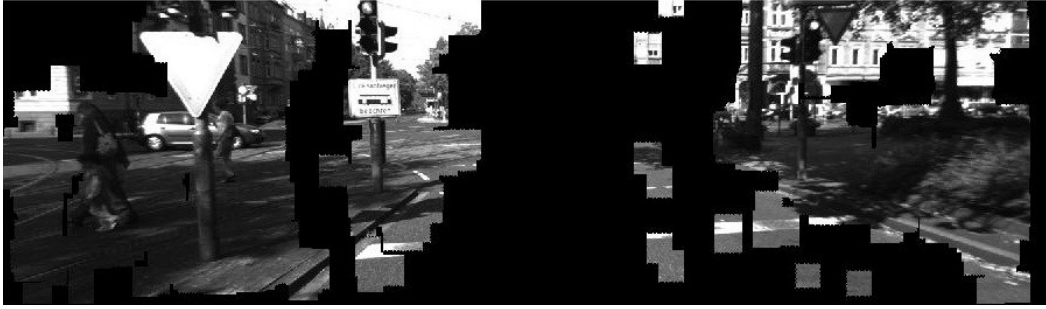

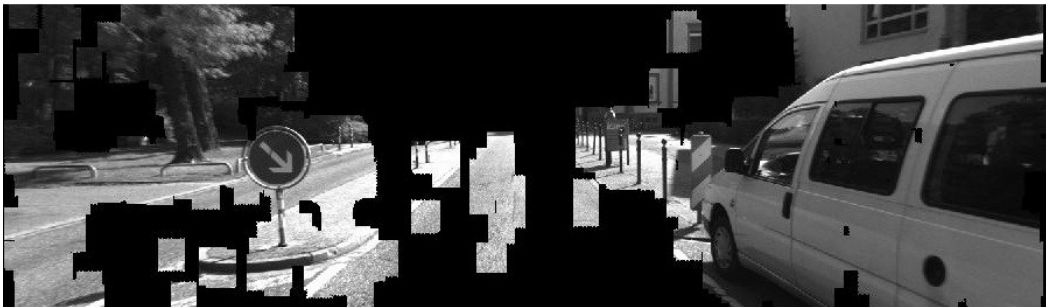Figure 82 - Reduced search mask of the frame 885 resulting from the input parameters of Region_7_0_20_4_8.
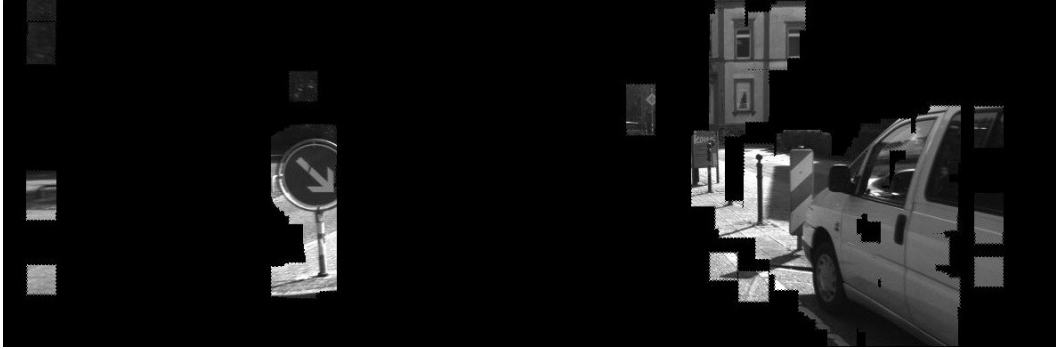
Figure 83 - Reduced search mask of the frame 1248 resulting from the input parameters of Region_7_0_20_4_8.

The Figures Figure 82 and Figure 83 show two mask of the results obtain of the input parameters Region_7_0_20_4_8. These input parameters miss 2 signs, such as the parameters Blockset_9_2_10_4_4, but they fail to have a better average sign fitting and a higher average mask size than the last one. For these showed examples and by analysing the Tables 9 and 11 can also be conclusive that the "Blockset" disparity images also outperform the "Region" disparity ones.

An extra experiment was done out of curiosity to see how much the mask would have to grow if we take the best results, in this case the "Blockset" results, and try to reach 100% of average sign fitting. The results showed that if the region growing parameter rises till the average sign reaches 100%, in the case of the input Blockset_7_0_4_6_10, the average mask size grows from 39.14% to 47.40% for only raising the average sign from 99.37% to 100%. In the case of two misses, if the input parameters of Blockset_9_2_4_4_10 are used and the growing parameter is raised to also reach 100% of average sign, the results show an increase of 12.16% of the search mask just to get from 99.11% to 100%. These results prove in a way that the use of the growing parameter to reach 100% of average sign should not be done because it causes the entire mask to grow a lot just to cause a small influence in the average mask size.

Chapter 6

Global Localization

This is the last part of the experimental procedure and it uses the disparity map images as an input to get to its final goal. This part has the objective of finding out the earth localization of the vertical road sign and it has two separate parts: the relative localization (from the sign to the car, where the camera is) and the global localization (from the car to the earth). The methodology section describes the line of thought that is behind the functions that were built to find the both localizations and how it was implemented. The results and discussion section shows the outcomes from the tests ran.

6.1 Methodology

The methodology part is divided in two parts since there were two functions built to fulfil different objectives.

6.1.1 Relative Localization

This part, as said before, involves the relative localization from the car to the vertical road sign. The idea here is to combine the information of depth in the disparity map image and get, having as an input the detection of the vertical road sign, the distance to it by transforming the disparity measure to meters. There are two operations that have to be done to get to the distance in measures; the first one is to get the average disparity of the vertical road sign and the second is to pass that disparity to meters. The problem now is to average the disparity of the road sign, using the disparity map image, in the correct way. If one looks at the disparity maps (Figure 84), more specifically to the signs on the image, it can be seen that the disparity that represents the sign has some

noise (due to bad stereo correspondence) and averaging it would lead to bad results. The solution is to withdraw the bad pixels by using the XZ-transposition and threshold methods used in section 5.2. First, the ground truth mask of the vertical road signs (like the one in Figure 85) is used to only keep the disparity image that is inside that ground truth (Figure 86).

Figure 84 – Example of disparity image containing a vertical road sign circled by a red line.
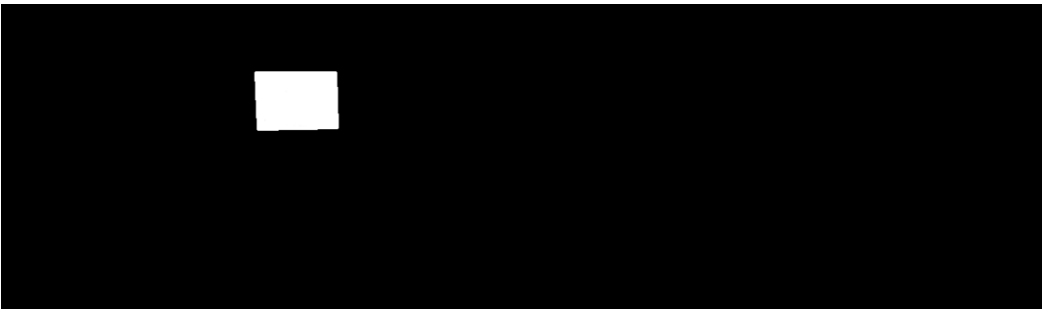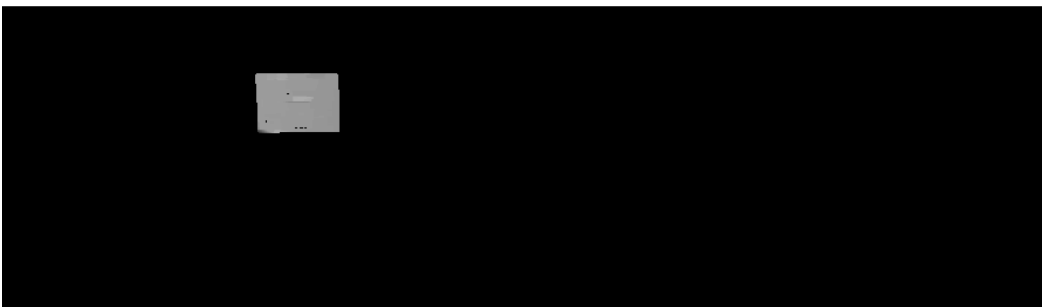

Figure 85 – Example of a road sign ground truth.

Figure 86 – Disparity image after applying the mask of the ground truth. (the ground truth image used can be seen in the Figure 83).

After this, the XZ-transposition is applied followed by a threshold cut, in the density image, for pixels that are less than the value 10. Having the bad pixels withdraw, now a balanced mean can be done directly in the XZ-plane, using the following pseudo code:

```
for i < size(density_image)
    for j < size(density_image)
        if density_image(i,j)~=0
                count_disparity=count_disparity+disparity_image(i,j)*j;
                pixel_count=pixel_count+density_image(i,j);
        end;
        j=j+1;
    end;
    i=i+1;
end;


disp_average=count_disparity/pixel_count;
```

The average disparity can now be used along with the camera parameters to find the distance in meters of the sign to the car (where the camera is). The camera parameters used are the focal length and the baseline between both cameras. The formula, explained in section 2.2.4, is then:

$$z = \frac{b.f}{disp\_average} \tag{11}$$

where b is baseline (the distance between the stereo cameras), f is the focal length of the cameras and z is the distance between the camera and the vertical road sign. The tests in this section will be mainly comparative since there is no possibility to know for sure the distance from the car to the signs. The tests consist on comparing signs that have visible distance differences and see if the distance in meters is consistent with the eye perception of depth in the original images. Different images will be used to compare the distances between each signs, and also images with more than one sign will be used. The disparity images that are used in these tests are the ones that have better results on the previous parts, in the evaluation part and in the sign detection part. Having a look at both parts, the disparity images from the algorithm "Blockset" with the inputs 150 (disparity range) and 7 (block matching window) are the ones chosen.

6.1.2 The Global localization

The global localization of the vertical road sign takes the measure from the previous relative calculation and sums up the calculation of the cars position on the earth. The car position is measure in GPS coordinates and it is given by the GPS device. Having the GPS coordinates and the distance to the sign, the measure of the cars orientation is still needed, which is also given by the GPS device inside the car. For this, a function with the entry of GPS coordinates, the car orientation and the distance of the car to the sign was built with the output of GPS coordinates relating to the global position of the vertical road sign. This function was built open the formula:

$$lat_f = \text{asin}\left(sin(lat_i) * cos\left(\frac{d}{R}\right) + cos(lat_i) * sin\left(\frac{d}{R}\right) * cos(\theta)\right) \tag{12}$$

$$lon_f = lon_i + atan2\left(sin(\theta) * sin\left(\frac{d}{R}\right) * \cos(lat_i), \cos\left(\frac{d}{R}\right) - sin(lat_i) * sin(lat_2)\right) \tag{13}$$

where θ is the cars orientation in relationship to the North direction, d is the distance from the car to the sign, R is the earths radius with the value of 6371 Km, $< lat_i, lon_i >$ the initial coordinates of the position of the car and $< lat_f, lon_f >$ the final coordinates, the global position of the vertical road sign. The tests of this function will be shown with random chosen inputs and their correspondent results. The reason for this function cannot be correctly verified to prove its value is that the GPS position of road signs are not known, making impossible to test if the GPS coordinates that are outputted in the present function are correct or not. The function was withdrawn from [56] in a web script implementation of GPS coordinates conversions. Unfortunately was not possible to recover the original source of this formula even though many website script developers of GPS coordinates calculations use it. Some tests were done using Google Maps information of GPS coordinates and the function proved to work for big distance differences but this method of testing could not be used to prove the function for smaller meter distances.

6.2 Results and Discussion

This section will be also divided in two results parts, as the methodology section is.

6.2.1 Relative Localization

The relative localization, as said before, has the objective of calculating the distance between the camera and the sign. To test these two images, three comparisons

scenarios are going to be presented. The first two, are comparative distances between signs in the same images.

Figure 87 – Test image used as an example for the results of global localization of road signs. The image has two signs that are circled by a red line.

Figure 88 – Ground truth of one of the signs of the image in the Figure 85.

Figure 89 - Ground truth of one of the signs of the image in the Figure 85.

In this first comparison, it is obvious that by looking at the original image (Figure 87) that the sign represented by ground truth image in the Figure 88 it is closer than the one represented by the Figure 89, and the results also show that. The sign represented by the ground truth image in the Figure 88 is 2.8 meters away from the camera (in the car) and the other sign is 4,9 meters according to the algorithm calculation.

Figure 90 - Test image used as an example for the results of global localization of road signs. The image has two signs that are circled by a red line.
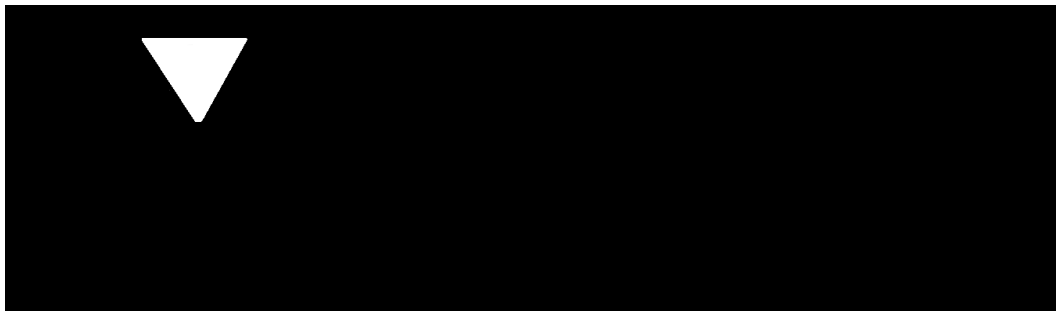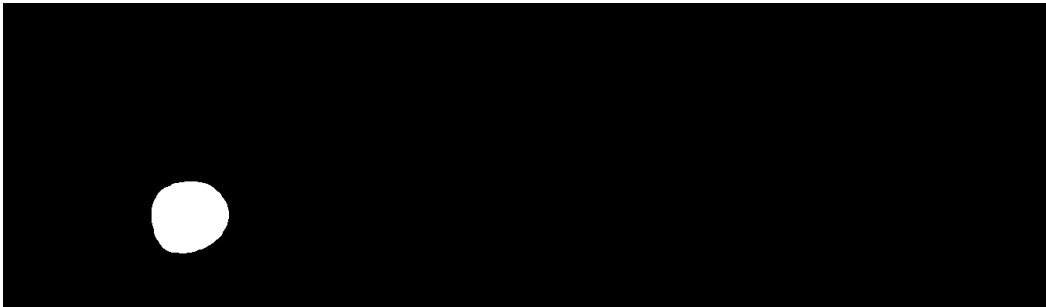

Figure 91 - Ground truth of one of the signs of the image in the Figure 88.
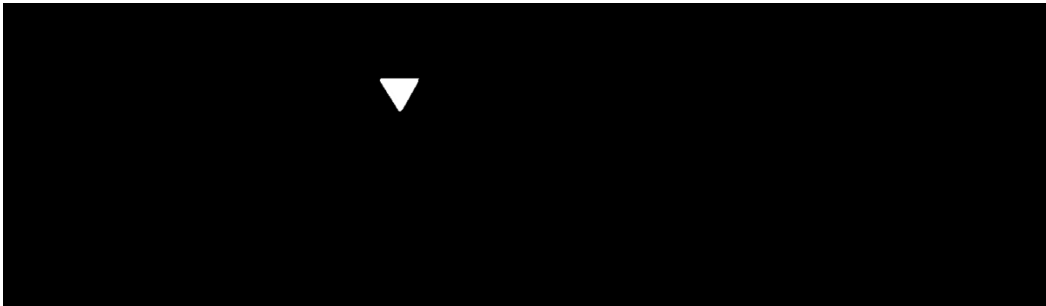
Figure 92 - Ground truth of one of the signs of the image in the Figure 88.

In this set of images the resuls are also coerent. The sign represented by the ground truth image in the Figure 91 is 2,6 meters away from the camera and the other sign represented by the Figure 92 is 7.7 meters away. The next set of images are distint ones but the signs selected in the images have clear distance differences from each other (the signs compared are the ones rounded with a red line).

Figure 93 - Test image used as an example for the results of global localization of road signs. The image has one sign that is circled by a red line.

Figure 94 - Test image used as an example for the results of global localization of road signs. The image has one sign that is circled by a red line.

Figure 95 - Test image used as an example for the results of global localization of road signs. The image has one sign that is circled by a red line.

The sign in the Figure 93 is the one that is closest to the camera from all of them, the distance calculatin gives the result of 2.6 meters. The farest away sign, respresent in the Figure 95 is 7.5 meters away from the camera and the sign that is between the both signs (in terms of distance perception) is 5.8 meters away from the camera (Figure 94). If one looks now at all of the comparisons done, it can be also clear that the results are more less consistent between each other, at a given degree of incertainty fo this test of course. The time spent in each relative localization run using a disparity map and the

correspondent sign mask was of average 0.1 seconds in a PC Intel Core 2 Quad @2.5GHz using Matlab® R2011a.

6.2.2 The Global localization

Three examples of inputs and their correspondent results are showed above:

First input

Initial GPS coordinates of the localization of the car:-70, 60

Distance to the sign: 5 meters

Orientation of the car (in relation to the north direction): $0°$

Second input

Initial GPS coordinates of the localization of the car: 30,-45

Distance to the sign: 8 meters

Orientation of the car (in relation to the north direction): $45°$

Third input

Initial GPS coordinates of the localization of the car: -10, -50

Distance to the sign: 500 meters

Orientation of the car (in relation to the north direction): $70°$

Results of the first input

Final GPS coordinates of the position of the sign: -69.999955033919690, 59.999999999999993

Results of the second input

Final GPS coordinates of the position of the sign: 30.000050873299440 -44.999941256528572

Results of the second input

Final GPS coordinates of the position of the sign: -9.998462042005652 -49.995709406863284

The time spent in each global localization run was calculated of less than 0.1μ seconds in a PC Intel Core 2 Quad @2.5GHz using Matlab® R2011a.

Chapter 7

Conclusions and Future Work

The conclusions are presented focusing in the goals of this project and in the results of it. Future work is proposed along the written conclusions whenever it can be applicable to what is being discussed.

Evaluating stereo matching algorithms with a dataset of stereo images without ground truth it is very difficult and challenging problem. The method created to evaluate these algorithms could not be proved to work in a precise way. Evidence of some truthfulness and veracity on the method was proved but only when the eye perception of the disparity maps could also match what the metric evaluation resulted. This means that the stereo matching algorithms with a big and eye perceptive difference in the disparity maps can be evaluated with this method but it does not prove that it can work in the evaluation of stereo algorithms that have smaller differences on their performance i.e. in their disparity maps. As a conclusion of this part, further work has to be done and explored to see when this method can be used and in what conditions proves itself to be an efficient evaluation technique of stereo matching algorithms.

In the second part, a series of techniques of image processing were joined into a procedure that, upon the input of the disparity map obtained by the stereo matching algorithm (that use the stereo pair of images given by the SVS installed in the moving vehicle) outputs a reduced search mask. The best results of this techniques showed a reduced search mask of around 40% of the original image can be done without disregarding any vertical road sign. If the system accepts to lose 2 signs, in the tested dataset of 51 signs (i.e. 96% of hit rate) the image can be reduced to around 30%. These results were based on the disparity maps of the best stereo matching algorithm, the "Blockset". The time consumption of this set of techniques was of 6-7 seconds in a PC Intel Core 2 Quad @2.5GHz using Matlab$^®$ R2011a. As a final conclusion of this part it

can be said that with some simple techniques applied to the disparity map the search for road signs can be reduced but it is not conclusive that it can be of use in traffic sign detection systems. Characteristics as time consumption are very important to define and evaluate to see if these techniques are worth of using as a useful pre-processing step.

In the third part of development of this project, the problem of global positioning of vertical road signs, given as input the disparity map and the detected road sign was addressed. A simple usage of the triangulation principles was used to calculate the distance of the cameras to the road signs after calculating a balanced disparity value of the road sign based on the disparity map. Tests show that, using a comparative evaluation, the method proved to work. It was not possible though to show if the calculation of the sign distance was completely correct or not because that would only be possible if the information about the distance of each frame to the signs was given. The relative localization cannot then be proved to work with accuracy. The final calculation of the global localization using the measure calculated in the relative localization and the GPS information about the cars position and orientation was done but not tested. As future work for this part, an experimental setup is advised to be done, with scenes shot and measurements taken of distances to the signs as well as the GPS information of both the signs and the car.

Generally overviewing the future work that further can be done in this project many things can be said. The stereo matching algorithms found and used were very poor according to most of the algorithms described in the Middlebury's evaluation. An implementation of one of the best scored algorithms is advised to be done to further testing of the techniques used here. Other conclusions can maybe be taken if a better disparity algorithm is used. Time consumption as also to be better studied for all of the techniques proposed to see if they are worthy of being used in real-time applications as well as in off-line ones. A bigger dataset should also be used to prove the possible usage of these techniques as well as a better suitable dataset (instead of the tested one that has a panoramic view that is of no interest to detect road signs). Other simple image processing techniques and other input parameters should also be tested to see which ones output a more reduced search mask with less time consumption.

References

[1] Ballard, D. H. and Brown, M.B., *Computer Vision*, New York: Prentice-Hall Inc, 1982.

[2] Szeliski, R., *Computer Vision: Algorithms and Applications*, Springer, 2010.

[3] De Cubber, G., *Variational methods for dense depth reconstruction from monocular and binocular video sequences*, PhD Thesis, *Vrije Universiteit Brussel, Royal Military Academy*, 2010.

[4] Bouguet, J. Y., *http://www.vision.caltech.edu/bouguetj/calib_doc/*. [acessed 2011 7 September].

[5] Peng, E. and Li, L., *Camera calibration using one-dimensional information and its applications in both controlled and uncontrolled environment,* Pattern Recogn., 43(3), pp. 1188-1198, 2010

[6] Faugeras, O., Luong, Q. and Maybank, S., *Camera self-calibration: Theory and experiments,* Second European Conference on Computer Vision, pp. 321-334, Santa Margherita Ligure, Italy, 1992.

[7] Tsai, R.Y., *A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses*, in *Radiometry*, B.W. Lawrence, A.S. Steven, and H. Glenn, Editors, pp. 221-244, 1992.

[8] Faugeras, O., *Three-Dimensional Computer Vision - A Geometric Viewpoint,* MIT Press, 1996.

[9] Zhang, Z., *Flexible camera calibration by viewing a plane from unknown orientations*, in ICCV, pp. 666--673, Kerkyra, Greece, 1999.

[10] Cao, X. and Foroosh, H., *Camera calibration without metric information using 1D objects*, in Image Processing, ICIP '04 International Conference, 2004.

REFERENCES

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

[12] Fusiello, A., Trucco, E. and Verri, A., *A compact algorithm for rectification of stereo pairs,* Machine Vision Appl., 12(1), pp. 16-22, 2000.

[13] Brown, M.Z., Burschka, D. and Hager, G. D., *Advances in computational stereo,* Pattern Analysis and Machine Intelligence, IEEE Transactions, 25(8), pp. 993-1008, 2003.

[14] Davis, J., Ramamoothi, R. and Rusinkiewicz, S., *Spacetime stereo: a unifying framework for depth from triangulationm,* Pattern Analysis and Machine Intelligence, IEEE Transactions, 27(2), pp. 296-302, 2005.

[15] Ferrer, J. and Garcia, R., *Bias reduction for stereo triangulation,* Electronics Letters, 46(25), pp. 1665-1666, 2010.

[16] Ozanian, T., *Approaches for stereo matching*, Modeling, Indentification and Control, 16(2), pp.65-94, 1995.

[17] Porta, J.M., Verbeek, J. J. and Kröse, B. J. A., *Active Appearance-Based Robot Localization Using Stereo Vision,* Autonomous Robots, 18(1), pp. 59-80, 2005.

[18] *The Global Positioning System : a shared national asset : recommendations for technical improvements and enhancements / Committee on the Future of the Global Positioning System, Commission on Engineering and Technical Systems, National Research Council*, ed. E. National Research Council, Commission on, S. Technical, and S. National Research Council, Committee on the Future of the Global Positioning. Washington, D.C. - National Academy Press, 1995.

[19] Nguwi, Y.Y. and Kouzani, A.Z., *A Study on Automatic Recognition of Road Signs*, Cybernetics and Intelligent Systems*, 2006 IEEE Conference, pp. 1-6,* 2006.

[20] Garcia-Garrido, M.A., Sotelo, M.A. and Martm-Gorostiza, E., *Fast traffic sign detection and recognition under changing lighting conditions*, Intelligent Transportation Systems Conference*, ITSC '06, IEEE*, pp. 811-816, 2006.

[21] Lafuente-Arroyo, S., et al. *A Tracking System for Automated Inventory of Road Signs*. in *Intelligent Vehicles Symposium, 2007 IEEE*. 2007.

[22] Broggi, A., Cerri, P., Medici, P., Porta, P.P. and Ghisio, G., *Real Time Road Signs Recognition*, Intelligent Vehicles Symposium*, IEEE*, pp. 981-986, 2007.

[23] Sin-Yu, C. and Jun-Wei, H., *Boosted road sign detection and recognition*, Machine Learning and Cybernetics, International Conference, pp.3823-3826, 2008.

REFERENCES

[24]   Yongping, W., Meiping, S. and Tao,W., *A Method of Fast and Robust for Traffic Sign Recognition*, Image and Graphics. ICIG '09, Fifth International Conference, pp. 891-895, 2009.

[25]   Feixiang, R.,   Huang, J.,   Jiang, R. and   Klette, R., *General traffic sign recognition by feature matching*, Image and Vision Computing, IVCNZ '09, 24th International Conference, pp. 409-414 New Zealand, 2009.

[26]   Chourasia, J.N. and Bajaj, P., *Centroid Based Detection Algorithm for Hybrid Traffic Sign Recognition System*, Emerging Trends in Engineering and Technology, 3rd International Conference, pp. 96-100 , 2010.

[27]   Ruta, A., Li, Y. and Liu X., *Real-time traffic sign recognition from video by class-specific discriminative features,* Pattern Recognitioin, 43(1), pp. 416-430, 2010.

[28]   Fei, Q., Bin, F. and Hengjun, Z., *Traffic Sign Segmentation and Recognition in Scene Images*, Pattern Recognition (CCPR), Chinese Conference, pp. 1-5, 2010.

[29]   Lin, C., Pan, S. and Zhang, F., *Traffic sign segmentation based on Maxwell color triangle*, Computer Application and System Modeling (ICCASM), International Conference, pp. 116-119, 2010.

[30]   Kastner, R., Michalke, T., Burbach, T., Fritsch, J. and Goerick, C. , *Attention-based traffic sign recognition with an array of weak classifiers*, Intelligent Vehicles Symposium (IV), IEEE, pp. 333-339, 2010.

[31]   Chen, Z., Yang, J. and Kong, B., *A Robust Traffic Sign Recognition System for Intelligent Vehicles*, Image and Graphics (ICIG), Sixth International Conference, pp. 975-980, 2011.

[32]   Houben, S., *A single target voting scheme for traffic sign detection*, Intelligent Vehicles Symposium (IV), IEEE, pp. 124-129, 2011.

[33]   Khan, J.F., Bhuiyan, S.M.A. and Adhami, R.R., *Image Segmentation and Shape Analysis for Road-Sign Detection,* Intelligent Transportation Systems, IEEE, 12(1), pp. 83-96, 2011.

[34]   Deguchi, D., Shirasuna, M., Doman, K., Ide, I. and Murase, H., *Intelligent traffic sign detector: Adaptive learning based on online gathering of training samples*, Intelligent Vehicles Symposium (IV), IEEE, pp. 72-77, 2011.

[35]   Yanlei, G., Yendo, T., Tehrani, M.P., Fujii, T. and Tanimoto, M., *Traffic sign detection in dual-focal active camera system*, Intelligent Vehicles Symposium (IV), IEEE, pp. 1054-1059, 2011.

[36]   Chen, L., Li, Q., Li, M. and Mao, Q., *Traffic sign detection and recognition for intelligent vehicle*, Intelligent Vehicles Symposium (IV), IEEE, pp. 908-913, 2011.

REFERENCES

[37]   Tao, C.V., *Mobile Mapping Technology for Road Network Data Acquisition*, Journal of Geospacial Engineering, 2, pp. 1-14, 2000.

[38]   Fugro, *www.roadware.com*.   [acessed 2011 7 September].

[39]   Chance, J.E., *www.jchance.com*.   [acessed 2011 7 September].

[40]   International, L.T., *www.lambdatech.com*.   [acessed 2011 7 September].

[41]   Hassan T., Lavigne, M. and El Sheimy, N., *VISAT^{TM}: The Avenue to Highway Data Banks*, 7th International Conference on Managing Pavement Assets, Calgary, 2008.

[42]   McLoughlin, S., Deegan, C., Mulvihill, C., Fitzgerald, C. and Markham, C., *Mobile mapping for the automated analysis of road signage and delineation,* Intelligent Transport Systems, IET, 2(1), pp. 61-73, 2008.

[43]   Scharstein, D., Szeliski, R. and Zabih, R., *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms*. in *Stereo and Multi-Baseline Vision (SMBV), Proceedings, IEEE Workshop*, pp. 131-140, 2001.

[44]   Scharstein D. and Szeliski, R., *http://vision.middlebury.edu/stereo/*.   [acessed 2011 7 September].

[45]   Mei, X., Sun, X., Zhou, M., Jiao, S., Wang, H. and Zhang, X., *On building an accurate stereo matching system on graphics hard*ware, to apppear in GPUCV'11, 1st IEEE Workshop on GPU in Computer Vision Applications, 2011.

[46]   Klaus, A., Sormann, M. and Karner, K., *Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure,* Pattern Recognition, ICPR, 18th International Conference, pp. 15-18, 2006.

[47]   Zeng-Fu, W. and Zhi-Gang, Z., *A region based stereo matching algorithm using cooperative optimization*, Computer Vision and Pattern Recognition, CVPR 2008, IEEE Conference, pp. 1-8, 2008.

[48]   The MathWorks, I., *http://www.mathworks.com/products/computer-vision/demos.html?file=/products/demos/shipping/vision/videostereo.html*. [acessed 2011 7 September].

[49]   Lankton, S., *http://www.shawnlankton.com/2007/12/3d-vision-with-stereo-disparity/*.   [acessed 2011 7 September].

[50]   Alagoz, B.B., *Obtaining Depth Maps From Color Images By Region Based Stereo Matching Algorithms,* OncuBilim Algorithm And Systems Labs, 8, 2009.

[51]   Abbeloos, W., *http://www.mathworks.com/matlabcentral/fileexchange/28522-stereo-matching*.   [acessed 2011 7 September].

REFERENCES

[52] Yang, Q., Wang, L., Yang, R., Wang, S., Liao, M. and Nister, D., *Real-time Global Stereo Matching Using Hierarchical Belief Propagation*, British Machine Vision Conference, pp. 989-998, 2006.

[53] Geiger, A., *http://rainsoft.de/software/datasets.html*. [acessed 2011 7 September].

[54] Costa, J.F.P.D., Alonso, H. and Cardoso, J.S., *The unimodal model for the classification of ordinal data,* Neural Network, 21(1), pp. 78-91, 2008.

[55] Cardoso, J.S., Cardoso, J.C.S. and Corte-Real, L., *Object-Based Spatial Segmentation of Video Guided by Depth and Motion Information*, Proceedings of the IEEE Workshop on Motion and Video Computing, IEEE Computer Society, p. 7, 2007.

[56] Veness, C., *http://www.movable-type.co.uk/scripts/latlong.html*. [acessed 2011 7 September].