Vision based Obstacle Detection for both Day and Night Conditions

Gijs Dubbelman
gijs.dubbelman@science.uva.nl
Artificial Intelligence —— Intelligent Autonomous Systems
December 2006

Supervisor: Drs. Wannes van der Mark

A thesis submitted to the Faculty of Science of the University of Amsterdam in partial fulfilment of the requirements for the Master of Science degree.

Intelligent Systems Laboratory Amsterdam
Informatics Institute
Faculty of Science
University of Amsterdam

TNO
Defence, Safety and Security
Electro-Optical systems

Intelligent Systems Laboratory Amsterdam
Informatics Institute
Faculty of Science
University of Amsterdam
Kruislaan 403
1098 SJ Amsterdam, The Netherlands
Tel: +31 (0)20 525 74 61 Fax: +31 (0)70 328 09 61
http://www.science.uva.nl/research/isla/

TNO Defence, Security and Safety
Electro-Optical Systems
Oude Waalsdorperweg 63, P.O. Box 96864
2509 JG The Hague , The Netherlands
Tel: +31 (0)70 374 00 00 Fax: +31 (0)70 328 09 61
http://www.tno.nl/defensie_en_veiligheid

Signature for approval of thesis submission by supervisor:

Drs. Wannes van der Mark,
January 2007,
Den Haag.

Abstract

The University of Amsterdam (UvA) and the Netherlands Organisation of Applied Scientific Research (TNO), developed the RoboJeep: a test platform for autonomous navigation in structured and unstructured terrain. The primary goal of any autonomous land vehicle is to make a distinction between drivable and non-drivable terrain. For this we have develop a stereo vision based obstacle detection system that can be used to detect positive and negative obstacles during both day and night conditions. Furthermore, we have quantitatively evaluated our system using a large real-world dataset. The system uses our novel disparity estimation algorithm together with triangulation to reconstruct the three dimensional coordinates of points visible through the stereo camera. Our disparity estimation algorithm is robust against a low signal-to-noise ratio making it suitable for night-time usage. It is based on a fine-to-coarse disparity selection scheme using a stereo image pyramid. Disparity selection is based on our disparity validity metric that reflects the reliability of a disparity estimate. This allows rejection of estimates from higher resolution levels in the pyramid and replacing them by estimates from lower resolutions levels when needed. Our tests based on a large and varied collection of day- and night-time images show the benefits of using our multi-resolution approach. The used obstacle detection techniques are column based and fast. For positive obstacle detection we introduce some new and promising techniques. Our method measures the terrain slope in a robust and efficient manner taking the inherent uncertainty in stereo reconstruction into account. And by using hysteresis thresholding the sensitivity to the angle between the slicing plane* and the surface normal is minimized. For evaluation we have used a wide range of parameter setting and plotted their response in ROC curves. We will show that we can reliably detect positive obstacles up to 50 meter during the day and 25 meter at night, without using narrow field of view cameras. Despite good results with positive obstacles, this research points out the difficulties of negative OD. While we can detect negative obstacles at acceptable distance during the day and night, false detections make reliable negative OD challenging. Finally, our OD benchmark datasets, methods, and metrics are the first step towards quantitative OD research using large datasets of real-world images.

Keywords: Stereo vision, Obstacle detection, Autonomous vehicles.

** plane defined by optical centre of the camera and the current image-sensor column used for slope estimation.*

Acknowledgement

During the time I worked on this thesis I was supported by a number of people. Foremost, I would like to express my gratitude to my supervisor Wannes van der Mark. His excellent advise on many topics was invaluable for the research described in this thesis. Furthermore, his patience and confidence allowed me to broaden my view on machine vision and to come up with innovative ideas. I also would like to thank Prof. Frans Groen and Johan van den Heuvel whose remarks gave this thesis the extra edge.

My colleges at TNO made me feel welcome and supported me with various aspects of my research. Especially Han van Bezooijen, who made it possible to use the Waalsdorpervlakte as our test terrain and also Marinus Maris for letting us use the Robojeep.

Of course I want to thank my family for supporting me throughout the years. My parents helped me in many ways and without them I probably would not have finished my masters. I also want to express my love for Ana who encouraged me throughout the months I was working on my research. Furthermore, special thanks go out to my uncle Kees Verkooijen who provided me with a roof over my head when I was a student in Amsterdam.

Finally I would like to express my gratitude to everybody who was involved with granting me a PhD. position for their confidence in me.

Gijs Dubbelman
January 2007
Den Haag

Content

Chapter 1

Introduction and overview

A vehicle can only travel safely through off-road terrain if it avoids obstacles such as trees, rocks and ravines. Autonomous robot vehicles must be able to accomplish this without the guidance of a human operator. It relies on obstacle detection (OD) algorithms that identify real-world obstacles in the path of the vehicle. The algorithms described is this thesis have been developed for RoboJeep: a platform for autonomous navigation research. RoboJeep is a joint research effort by TNO Defence Security and Safety and the Faculty of Science at the University of Amsterdam.

Figure 1.1: RoboJeep.

The goal of research related to RoboJeep is to develop methods that enable a vehicle to carry out missions autonomously. The vehicle should only use on-board resources and must be as reliable as a human operated vehicle. Furthermore, the vehicle must be able to carry out its mission in urban (structured) as well as off-road (unstructured) terrain during day and night. An autonomous vehicle like RoboJeep can be used for tasks that are hazardous to humans. Example applications can be found in disaster areas or for defence related reconnaissance,

land mine detection and personnel rescue. Also tasks that are considered tedious can be carried out by an autonomous robot vehicle. Examples are construction work, agricultural automated harvesting and automated transport. Apart from pure autonomous operation the developed methods can also be applicable for assistance systems in mainstream cars that enhance traffic safety. To accomplish its goals the vehicle must sense its environment and construct an accurate world model. For this, RoboJeep can use a stereo-vision camera, ultra sonic sensors, Laser Imaging Detection and Ranging (LIDAR), navigation aids such as an Inertial Measurement Unit (IMU) and a Global Positioning System (GPS). Based upon the constructed world model the vehicle must be able to reason about possible solutions for its tasks and make reliable decisions. To carry outs its decisions Robojeep uses an automated gearbox, steering wheel, accelerator and brake pedal. Current focus is to develop methods that enhance the environmental awareness of the Robojeep. In line with that goal the research described in this thesis focusses on methods that improve the obstacle detection capabilities of the RoboJeep.

1.1 Need for obstacle detection improvement

Without an accurate world model any autonomous vehicle is prone to make mistakes and most likely will fail to carry out its task. A key part of the construction of a world model is the incorporation of obstacles. The problem of reliable obstacle detection (OD) for autonomous ground vehicles has not been completely solved. A considerable research effort has been put into OD during day-time conditions. However, many solutions put restrictions on the type of environment the vehicle can operate in. Solutions that allow reliable OD in complex environments such as off-road terrain, cities and during various conditions such as day- and especially night-time do not exist. The problem is that no solution reaches the needed level of performance for save operation for both vehicle and environment, including humans. Whereas in most academic classification problems a performance level of 95% is considered good, hitting 5 out of 100 obstacles is unacceptable. An often used solution is to increase the sensitivity of the system. This means increasing the amount of obstacles that are detected at the expense of also increasing the amount of non-obstacles detected as obstacles. Although this will make the vehicle much safer it will also make it less efficient, because it will avoid a lot of false obstacles. The best performing systems use several LIDAR systems to scan the environment, Thrun [60]. While LIDAR has its own disadvantages they are also expensive. The advantage of using a stereo camera system is that besides estimating depth we can also use the obtained images for further image analysis. An example is the use of colour or texture based terrain classification or finding road signs.

It is clear that no autonomous ground vehicle can do without reliable obstacle detection. It is also clear that current vision based methods do not reach the needed level of performance, especially during low visibility conditions. These two observations together show that there is still a lot of work to be done in the field of obstacle detection for autonomous ground vehicles.

1.2 Problem statement

The topic of this study is to investigate the suitability of stereo based OD during night-time conditions and compare it to day-time conditions. The stereo based obstacle detection system should be able to segment an image into drivable and un-drivable terrain. For un-drivable terrain we make a distinction between positive and negative obstacles. Positive obstacles are objects that extend out of the ground plane, examples are trees, rocks and people. Negative obstacles are objects that extend into the ground plane, examples are ditches, ravines and holes. The first task of the OD system is to reconstruct the 3D coordinates of the terrain in front of the vehicle. To accomplish this we will use a stereo vision approach. Next, based on this 3D reconstruction we can search for obstacles and drivable terrain alike. The constraints to the stereo based OD algorithm are:

1. The algorithm must be applicable for unstructured terrain during day- and night-time.
2. The algorithm should be able to detect positive and negative obstacles at distances that allow avoidance of the obstacle.
3. The algorithm should be applicable for real-time implementation in the near future.

As mentioned earlier, the first task is to reconstruct the 3D properties of the scene in front of the vehicle, for this we rely on stereo vision. At the beginning of the research it was already clear that literature about stereo vision during low-visibility condition such as during the night is limited. As a consequence, thorough research into methods that increase the robustness of stereo vision systems are required. Furthermore, quantitative OD evaluation over large datasets is not the standard in the OD research community. To our knowledge review papers that compare different OD approaches in a true quantitative way over a large real-world dataset do not exist. This makes it very challenging to get insight into the performance differences and applicability of existing OD methods. As an effect considerable amount of work has to be dedicated to record datasets and find suitable performance methods and measures.

1.3 overview report

Our report starts with an introduction into the existing work from the research community. In chapter 2, the focus is on disparity estimation, obstacle detection and stereo geometry. Here we will also discuss the difficulties of robust scene reconstruction using stereo vision and the inherent limitations of stereo based OD. In chapter 3 we present our solutions to robust scene reconstruction and obstacle detection. We will also relate our methods to existing work from the literature. Furthermore, we present the recorded benchmark dataset and the measures and methods used to evaluate our system using various configurations and conditions. Chapter 4 presents the obtained result. We first focus on scene reconstruction performance. Next, we describe the OD results for our day- and night-time dataset. In Chapter 5 we present our conclusions. Finally, in chapter 6 we give potential topics for further research.

Chapter 2

Related research

In this chapter, methods are discussed for obstacle detection with stereo vision as well as related research from literature. In section 2.1, 2.2 and 2.3 we focus on computational stereo. First, we will present the geometrical aspects of stereo vision in section 2.2. Next in section 2.3, we describe the latest research into disparity estimation. Then, in section 2.4 the focus is on obstacle detection. In section 2.5 we discuss the limitations of stereo based obstacle detection and possible solutions. Finally, in section 2.6 we conclude the literature overview and present our research directions.

2.1 Stereo Vision

The goal of stereo vision is to recover the three-dimensional (3D) coordinates of real-world points seen through a binocular camera system. A binocular camera system consists of two cameras mounted on a baseline, see figure 2.1. In general there are two techniques for 3D reconstruction using a binocular camera system. There is active stereo and passive stereo also known as computational stereo. With active stereo we can control the convergence between the two cameras. When estimating the depth of an object an active stereo system tries to centre the object in both images by modifying the convergence between the cameras. If the object is centred we can compute its 3D position by triangulation based on the binocular camera parameters and the convergence between the two cameras. With passive stereo systems the convergence between the camera is fixed. 3D reconstruction is now based on finding corresponding point pairs. A corresponding point pair consist of two image points belonging to one real-world point, one in the left camera and one in the right camera, see figure 2.1. Given the image position of the corresponding points of a world point in both images and the binocular camera parameters we can calculate its 3D coordinates. In this thesis we only focus on passive stereo. The process of passive stereo sketched above will be described in more detail in the next sections. Section 2.2 describes the geometrical background of binocular camera systems. We will describe how to obtain the binocular camera parameters as well as how to reconstruct 3D coordinates given two correspondence points. The problem of finding corresponding point pairs for every point visible in both cameras, the so called correspondence problem, is discussed in section 2.3.

2.2 Multi view geometry and Passive Stereo

The computer vision field that deals with understanding and modelling of the geometry of multiple cameras is called multi-view geometry. In the last decades the theory and methods have reached a mature level. The book written by Hartley and Zisserman [17] gives an excellent in-depth information about the topic. In following sections we confine ourselves to a brief overview of theory and methods that apply to passive stereo.

The process of passive stereo can be subdivided in four components. First, the binocular camera system has to be calibrated. By stereo calibration we obtain the parameters of the binocular camera system. Secondly, based on the obtained binocular camera parameters we rectify the images taken by the camera system. Rectification will transform the images as if they were taken by an ideal binocular camera system, see figure 2.1. Thirdly, we have to find the points $\boldsymbol{p_l}$ and $\boldsymbol{p_r}$ that correspond to the same physical point $\boldsymbol{P_W}$ for every point visible in both images. Fourthly, once we have found the points $\boldsymbol{p_l}$ and $\boldsymbol{p_r}$ we can use their horizontal image coordinates to find the disparity d between them. Using the disparity together with the rectified binocular camera parameters we can compute the 3D coordinates of $\boldsymbol{P_W}$ relative to O_V. In our case O_V is the centre of the vehicle. In the coming sections we will go deeper into the process sketched above. First we present the geometrical model of single camera and binocular camera systems. Then undistortion, rectification, depth estimation and depth uncertainty will be discussed in more detail.

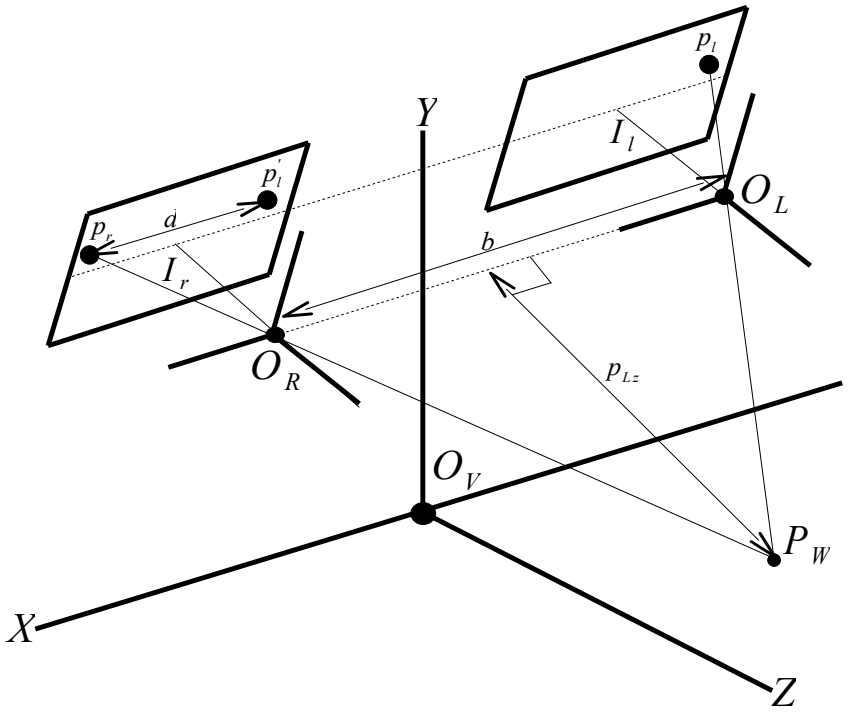


Figure 2.1: Binocular camera model.

2.2.1 Camera model

The model of a single camera can be subdivided in its internal and external properties also known as intrinsic and extrinsic parameters. The intrinsic parameters describe how a point is projected on the image plane *I*. The extrinsic parameters give the position of the camera's focal point O_C according to a world reference frame. The extrinsic parameters consist of two parts. First a rotation matrix **R** which describes the rotation between the world coordinate system and the camera coordinate system. Second a translation vector $\boldsymbol{T}$ that describes the translation between the world coordinate system and the camera coordinate system. The relation between the world coordinates and the camera coordinates for a given point is then described as.

$$\boldsymbol{P}_c = \mathbf{R}\,\boldsymbol{P}_W + \boldsymbol{T} \qquad (2.1)$$

Where $\boldsymbol{P}_w$ are the coordinates $\begin{pmatrix} P_{wx} & P_{wy} & P_{wz} \end{pmatrix}^T$ of point $\boldsymbol{P}$ in the world coordinate system and $\boldsymbol{P}_c$ are the resulting coordinates $\begin{pmatrix} P_{cx} & P_{cy} & P_{cz} \end{pmatrix}^T$ in the camera coordinate system. After the point is transformed to the camera coordinate system projection can be applied. We first describe the pin-hole camera model, see figure 2.2. The pin-hole model is a linear, thus distortion free, model for projection three-dimensional points to a two dimensional (2D) image. It tries to capture the process of light falling trough a lens onto an imaging plane. The lens is modelled as one single point O_C, that is the pin hole through which all light falls. The line through O_C perpendicular to the imaging plane *I* is known as optical axis. The length from O_C to *I* which is the focal length is written down as *f* (note that in the formulas we assume the focal plane to be in front of the optical centre hence the positive *f*). The image coordinates p_{ix} of p_{iy} can be computed from the camera coordinates of *P* using:

$$\boldsymbol{p}_i = \begin{pmatrix} p_{ix} \\ p_{iy} \end{pmatrix} = \begin{pmatrix} f\dfrac{P_{Cx}}{P_{Cz}} & f\dfrac{P_{Cy}}{P_{Cz}} \end{pmatrix}^T \qquad (2.2)$$

And the *x* and *y* coordinates of $\boldsymbol{P}$ relative to O_C can be computed from the image coordinates $\boldsymbol{p}_c$ (when $\boldsymbol{p}_{cz}$ is known) using:

$$\boldsymbol{P}_c = \begin{pmatrix} P_{cx} \\ P_{cy} \\ P_{cz} \end{pmatrix} = \begin{pmatrix} \dfrac{P_{cz}\,p_{ix}}{f} & \dfrac{P_{cz}\,p_{iy}}{f} & P_{cz} \end{pmatrix}^T \qquad (2.3)$$

The Pin-hole camera is a simple and straightforward model for image projection. Almost all real camera systems come equipped with a glass lens instead of a pin-hole. Lens projection can add several types of distortions to the images. These have to be removed before the pin-hole model can be applied. To do this we use a more complex camera model as described by Zhang [66], and estimated its parameters for the used camera by means of camera calibration, Heikkilä and Silvén [19].

Camera calibration is usually performed using a plane on which several markers are printed. The size of the plane and relative position of these markers is known beforehand. By taking several images of this calibration plane under various orientations and extracting the image coordinates of the markers we can estimate the projection parameters of the used camera. Once we have the parameters of the complex model we can define a transformation from the complex model to the Pin-hole model. This transformation can be applied to the images and hence they will appear as if they were taken by a Pin-hole (linear distortion free) camera. This process is usually referred to as undistortion. In the next page we describe the used complex camera model and its parameters. In all other sections we will assume that the used cameras behave according to the pin-hole model.

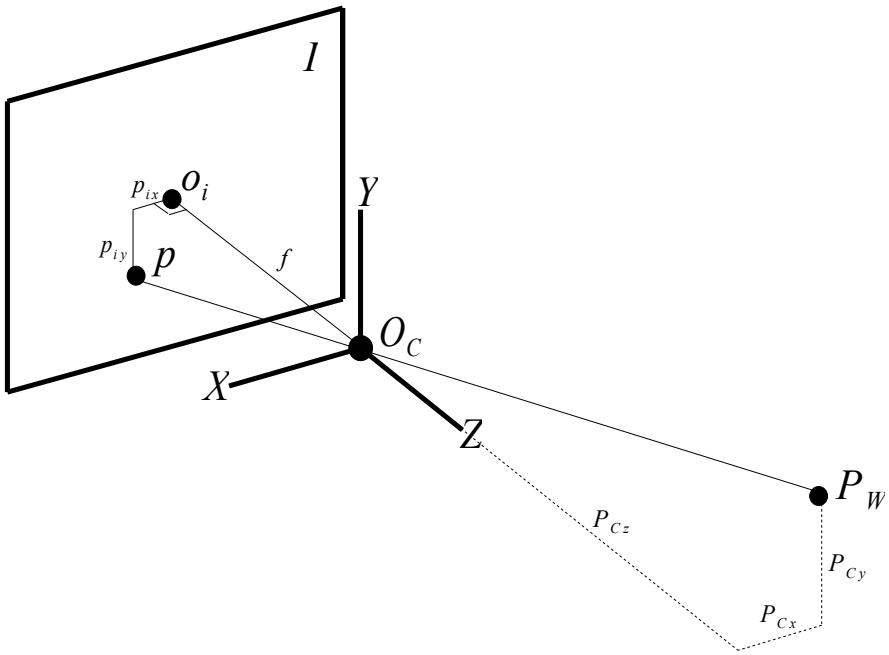


Figure 2.2: Pin-hole camera model.

The used camera model model calculates the image projection in three stages. First the normalized (*f = 1*) image coordinates are calculated with the formula below.

$$\boldsymbol{p_n} = \begin{pmatrix} p_{nx} \\ p_{ny} \end{pmatrix} = \begin{pmatrix} \frac{P_{Cx}}{P_{Cz}} & \frac{P_{Cy}}{P_{Cz}} \end{pmatrix}^T \qquad (2.4)$$

Then radial distortion defined by the parameters k_1 and k_2 is applied using:

$$r^2 = x^2 + y^2$$
$$\boldsymbol{p_d} = \begin{pmatrix} p_{dx} \\ p_{dy} \end{pmatrix} = (1 + k_1 r^2 + k_2 r^4)\, \boldsymbol{p_n} \qquad (2.5)$$

Where *r* is the radius from the image centre point o_i. The possible effects of radial distortion are shown in figure 2.3. Finally the distorted image coordinates are un-normalized using:

$$\boldsymbol{p_d} = \begin{pmatrix} f_x p_{dx} + cc_x \\ f_y p_{dy} + cc_y \end{pmatrix} \qquad (2.6)$$

Here cc_x and cc_y model the coordinates of the principal point. The principal point O_i is the projection of the optical centre O_C onto the imaging plane *I*. The focal length is modelled by f_x and f_y.

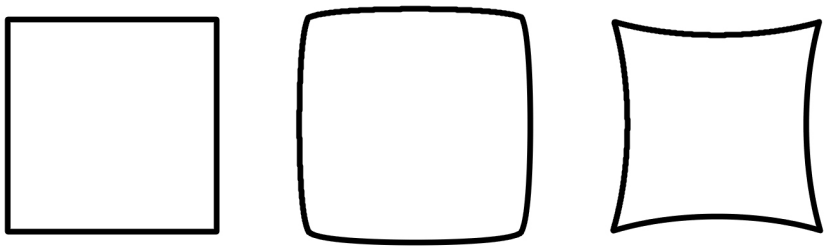


Figure 2.3: Effects of radial distortion, undistorted (left), barrel distortion (middle), pincushion distortion(right).

2.2.2 Epipolar geometry

Epipolar geometry refers to the geometry of two cameras. It builds upon the geometrical model of a single camera as described in section 2.2.1 and adds a rotation and translation that describes the relative position and orientation between the two cameras. The general set-up of a binocular camera system is shown in figure 2.5. A photo parallel camera system, see figure 2.4, is a set-up where the optical axis of both camera's are parallel to each other. Furthermore, the baseline connecting the two optical points runs parallel to the image lines in both imaging planes. Searching for corresponding points is less difficult when using a photo parallel system. However, it is very difficult to align two cameras manually so that they are exactly photo parallel.

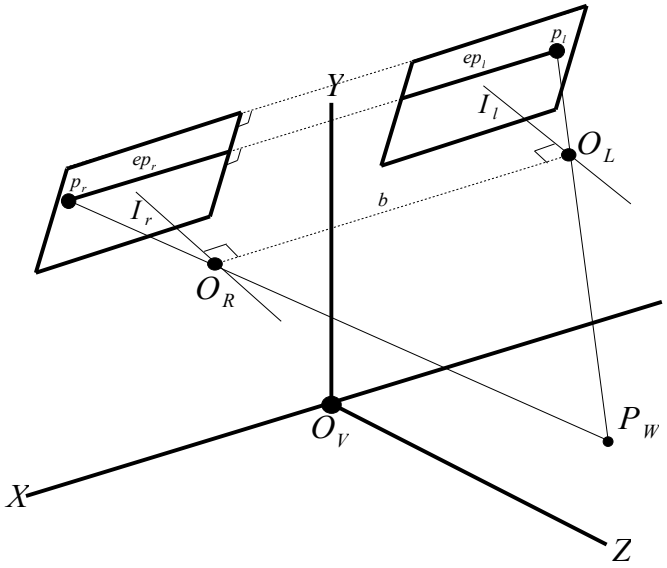


Figure 2.4: Photo parallel binocular camera system.

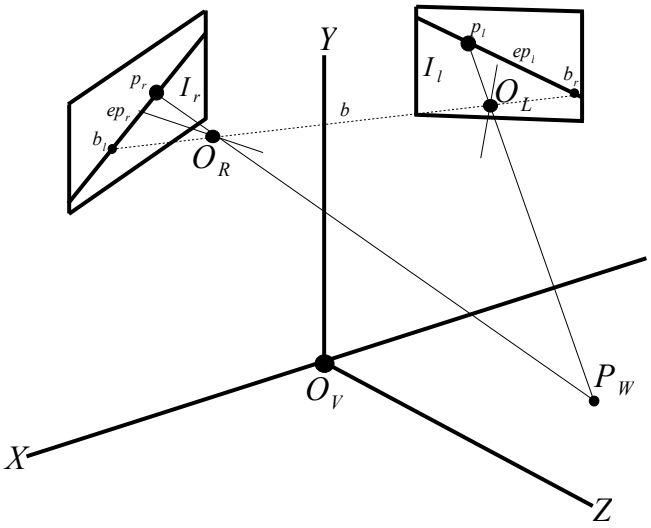


Figure 2.5: Non rectified binocular camera system.

By means of stereo calibration we can estimate the stereo parameters of the binocular camera system. Using these parameters we can construct a transformation from the used binocular camera system to a photo parallel binocular camera system. Applying this transformation to the images i.e. epipolar rectification will transform the images as if they were taken by a photo parallel camera system. In the text below we explain the process and benefits of stereo calibration and epipolar rectification. First we will introduce a number of terms. The *epipoles* b_l and b_r are the points of intersection of the baseline b, the line joining the optical centres O_l, and O_r, with the image planes I_l and I_r. Thus, an epipole is the projection, in one camera, of the optical centre of the other camera. The *epipolar plane* is the plane defined by P_W and the optical centres O_l and O_r. The *epipolar lines* ep_l and ep_r are the straight lines of intersection of the epipolar plane with the image planes I_l and I_r. It is the projection in one camera of a ray through the optical centre and image point in the other camera. All epipolar lines intersect at the epipole (except for photo parallel stereo cameras, in this case the epipoles are at infinity)

By calibration of the stereo camera rig we obtain the intrinsic and extrinsic parameters of both cameras. For stereo calibration we can use a similar method as described in section 2.2.1. The intrinsic parameters describe how points are projected onto the imaging planes of both cameras. It includes parameters such as the cameras focal length, principal point and the lens distortion. The extrinsic parameters describe the transformation from the right cameras coordinate system to that of the left one. It is composed of a rotation and a translation. Using this information we can find the epipoles in both images by computing the intersection of the baseline with both imaging planes. If we want to search for the corresponding point of p_l in I_r we know it must lay on the epipolar line ep_r in I_r. To find the epipolar line we compute the intersection of the imaging plane I_r with the epipolar plane defined by p_l, O_l and b_l. So the benefit of camera calibration is that it simplifies the search of corresponding points from a 2D search to a 1D search along the epipolar lines. By epipolar rectification of the images the epipolar lines will become parallel to the image lines. Therefore, we only have to search along the same horizontal scan-line in I_r to find the corresponding point of p_l. Furthermore, we know that the point p_r must lay left of the image coordinates of p_l in I_r. Epipolar rectification first transforms the projection matrices of both cameras to the pin-hole model (un-distortion) where the focal length will be the same for both cameras. The next step is to cancel out the orientation difference between the imaging planes of both cameras making them co-planer. Then the offset between the vertical positions of the focal points is removed. All that remains is a horizontal offset between the optical centres e.g. the baseline. Finally, the transformations used for epipolar rectification can be used to warp the images. This will cause the image to appear as if they were taken by a photo parallel camera system.

2.2.3 Depth estimation & 3D reconstruction

The geometry needed for the stereo reconstruction of image points is explained in this section. It assumes the pin-hole model and that the cameras are in a photo-parallel set-up. Images from real cameras can also be used if they are un-distorted and rectified first. To reconstruct the 3D coordinates of image points we first need to obtain its distance from the camera. Once the distance is obtained we can use formula 2.3 to calculate the other coordinates.

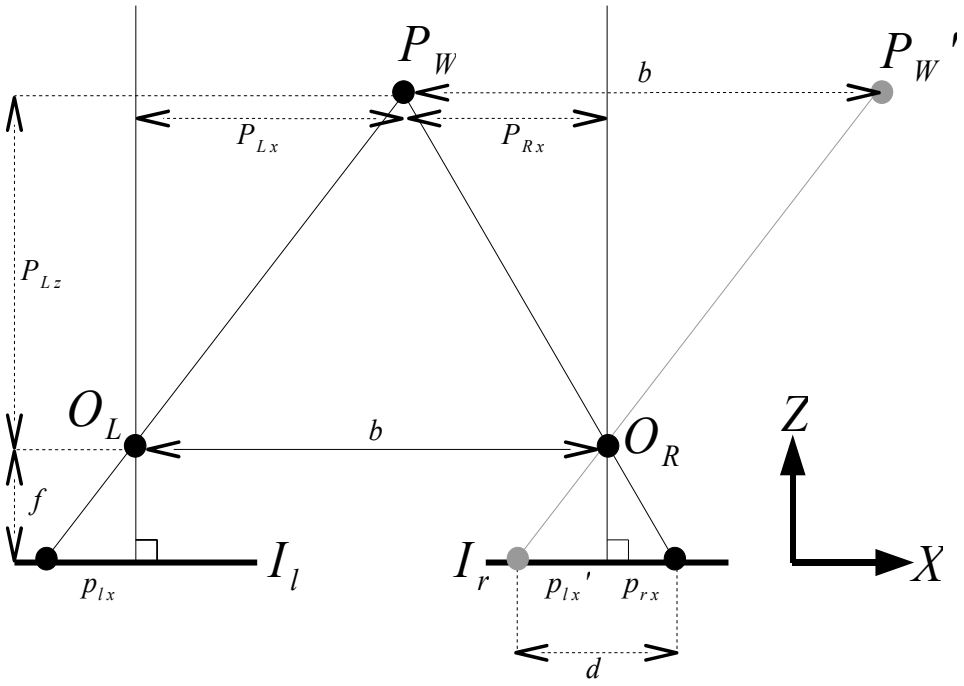


Figure 2.6: Stereo geometry.

First note that distances P_{Lz} and P_{Rz} are equal and that the difference between P_{Lx} and P_{Rx} is equal to b, which leads to:

$$\frac{P_{Lz}}{P_{Lx}-P_{Rx}}=\frac{f}{p_{lx}-p_{rx}}\rightarrow\frac{P_{Lz}}{b}=\frac{f}{d} \qquad (2.7)$$

The depth of $\boldsymbol{P_W}$ relative to O_L can be computed with:

$$P_{lz}=\frac{f\,b}{d} \qquad (2.8)$$

Once all 3D coordinates are calculated using formula 2.3 we can transform the coordinates from the left camera coordinate system to the vehicle coordinate system using:

$$\boldsymbol{P_v}=\mathbf{R_v}\,\boldsymbol{P_{Lc}}+\boldsymbol{T_v} \qquad (2.9)$$

Where $\mathbf{R_v}$ describe the rotation between the left camera frame and vehicle coordinate frame and $\boldsymbol{T_v}$ the translation between them.

2.2.4 Uncertainty in depth estimation

In the previous sections we have assumed that the projections onto the imaging plane can be described by points. In reality imaging planes work with pixels which have a certain size. Pixel size causes a minimum and maximum bound on depth estimation, as can be seen in figure 2.7. The effect of pixel size on depth uncertainty can be minimized using sub-pixel disparity estimation, which is described in section 2.3.14. In the text below we describe the influence of a points distance from the camera, base line width, focal length and pixel size on depth uncertainty.

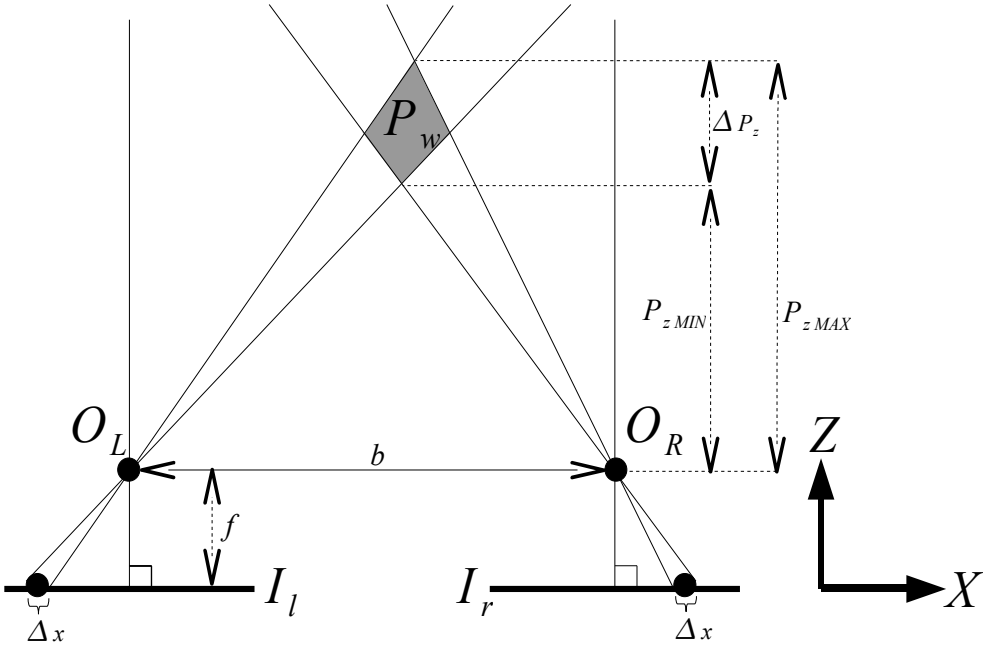


Figure 2.7: Uncertainty in depth.

Given the disparity d between two corresponding points p_l and p_r, the minimum and maximum depth can be computed with:

$$P_{z\,MIN} = \frac{f\,b}{d + \Delta x} \qquad P_{z\,MAX} = \frac{f\,b}{d - \Delta x} \tag{2.10}$$

The influence of the pixel width Δx on the difference between the maximum and the minimum depth ΔP_z can be modelled with:

$$\Delta P_z^{\,2} = \left(\frac{\delta P_z}{\delta p_{lx}} \Delta x \right)^2 + \left(\frac{\delta P_z}{\delta p_{rx}} \Delta x \right)^2$$

$$\frac{\delta P_z}{\delta p_{lx}} = -\frac{f\,b}{d^2} \qquad \frac{\delta P_z}{\delta p_{rx}} = \frac{f\,b}{d^2} \tag{2.11}$$

Finally the influence of P_z on ΔP_z can be computed with:

$$\Delta P_z = \frac{\sqrt{2}\,f\,b\,\Delta x}{d^2} \;\rightarrow\; \Delta P_z = \frac{\sqrt{2}\,\Delta x\,p_z^2}{f\,b} \tag{2.12}$$

2.3 Disparity estimation

As discussed in section 2.2 reconstructing the three dimensional properties of a scene requires finding correspondence points and the disparity between them. In this section we look into the field of stereo algorithms. We will restrict the meaning of the term stereo algorithms to techniques that enable disparity estimation from stereo image pairs taken by a binocular camera system, seen figure 2.1. A stereo image pair consists of one image from the left camera together with one image from the right camera. It is assumed that both images are taken at the same time. Furthermore, both images from the stereo image pair have rectified epipolar geometry according to the intrinsic and extrinsic parameters of the binocular camera system, see section 2.2.2.

The key issue that stereo algorithms try to solve is the construction of a *disparity image* $d(x, y)$. This boils down to finding for each point (x, y) in the image of one camera one corresponding point in the image from the other camera. Once we have two corresponding points from a rectified stereo image pair the disparity value $d(x, y)$ can be obtained. For instance if $I_l(h, w)$ and $I_r(k, w)$are correspondence points in a rectified stereo image pair then $d(h, w)=h-k$. The search for correspondence points is one of the key research topics in machine vision and is known as the correspondence problem. When the stereo image pair is rectified the search complexity is $O(HW^2)$. This is because the epipolar lines are co-parallel and thus we only have to search in one image line for a corresponding point. In practice we do not have to search the whole line in the reference image. Usually a maximum bound is set on the disparity between correspondence points. In this case the search complexity if reduced to $O(HWMax_d)$. Apart from the search space complexity there are other aspects which make the correspondence problem so difficult. Points on image patches with bad signal to noise ratio are usually prone to errors. Bad signal to noise ratio can be caused by the absence of texture. Repetitive textures on the other hand, pose difficulties due to the great similarity between subregions in the texture. Also the difference in camera gain and bias, perspective distortion and occlusions make the search challenging. Because the literature on night-time stereo is very scarce we focus on general disparity estimation. The last section of this chapter, 2.3.16, deals with night-time stereo. For a general overview of recent research into disparity estimation we refer to Brown et al. [8], Scharstein and Szeliski [53]. For a overview and analysis of techniques that can be used for autonomous vehicles we refer to van der Mark and Gavrila [41].

2.3.1 Notation

We will look into stereo algorithms that have as output a *disparity image* $d(x, y)$ with respect to a reference image. Without loss of generality we will take the left image of the stereo image pair to be the reference image I_l and the right image to be the matching image I_r. $d(x, y)$ will give as output a disparity value for pixel $I_l(x, y)$ in the reference image. The output of $d(x, y)$ means that given the position (x, y) in the reference image, the physical point that is associated with $I_l(x, y)$ can be found at position $I_r(x - d(x, y), y)$ in the matching image. At the same time the output of $d(x, y)$ is directly related to the depth of an image point, as discussed earlier in section 2.2.3.

2.3.2 Dense and sparse stereo

A method that tries to estimate a disparity value at every location in the reference image is called *dense stereo.* If the number of locations is limited to certain image features for instance edges, the method is referred to as *sparse stereo.* Sparse stereo was popular during the early days of machine vision when computation power was limited. Nowadays dense stereo has become feasible at real-time frame rates and receives most interest. In the remainder of this thesis we will focus on dense stereo.

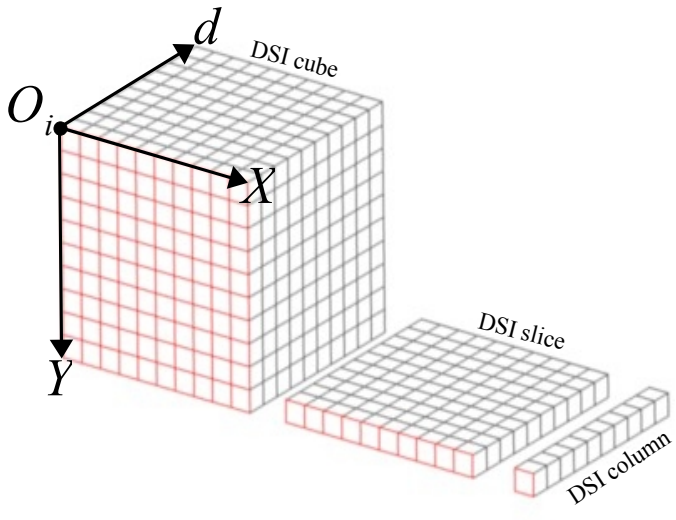


Figure 2.8: Disparity Space Image.

2.3.3 Disparity space image & Disparity estimation

Central to stereo algorithms that produce a disparity map $d(x, y)$ is the notion of a *disparity space image (DSI)* also known as *disparity search space* or *correlation space*. A $DSI(x, y, d)$, see figure 2.8, gives for every position in the reference image and a certain disparity a similarity measure. The similarity measure given by $DSI(x, y, d)$ is the confidence that the point $I_l(x, y)$ and $I_r(x-d, y)$ relate to the same physical point. Estimating disparity using a DSI basically involves three steps. With some methods these steps are clearly separated. Other methods combine steps together and even can use several iterations of them. The first step is to fill the DSI with the output of a single pixel similarity function *S* for every possible value for *x*, *y* and *d*.

$$DSI(x, y, d) = S(x, y, d), \quad x \in [1..\mathrm{H}] \quad y \in [1..\mathrm{W}] \quad d \in [0..\mathrm{Max}_d]$$

The function *S* tells us how much a pixel from the left image is similar to a pixel in the right image. Single pixel similarity is not always distinctive enough. So the next step is to aggregate the output of the similarity function.

$$DSI(x, y, d) = \sum_{k \in K(x, y, d)} DSI(k_x, k_y, k_d), \quad x \in [1..\mathrm{H}] \quad y \in [1..\mathrm{W}] \quad d \in [0..\mathrm{Max}_d]$$

Where $K(x, y, d)$ gives a set of coordinates inside the DSI that can influence the similarity value of $DSI(x, y, d)$. For instance this can be done by using a fronto-parallel 4-connected summing filter e.g. *K(x,y,d) = [(x,y,d), (x-1,y,d), (x+1,y,d), (x,y-1,d), (x,y+1,d)]*. The effect is that neighbouring pixels influence each others similarity value and eventual will influence each others disparity estimate. The third step is to estimate for every point $I_l(x, y)$ the best value for *d* so that the overall disparity map matches the true disparity most likely. This is done by optimizing for *d* over the *DSI*.

$$d = O_d(DSI)$$

A distinction can be made between optimizations methods based on the portion of the DSI they take into account when optimizing the disparity for a given pixel. Local methods optimize the disparity for a pixel solely based on its DSI column. Scan-line methods optimize the disparity for a pixel based on its DSI slice. And global optimization methods optimize the disparity for a pixel based on the entire *DSI* cube. In sections 2.3.4 up to 2.3.9 we will discuss the filling of the DSI with the output of a similarity function. Then in sections 2.3.10 up to 2.3.13 we describe optimization methods to extract the disparity image from the DSI. Finally, section 2.3.14 and 2.3.15 provide post-processing methods and methods to estimate the quality of the found disparity on a pixel by pixel basis.

2.3.4 Similarity computation

In order to find correspondence points in a stereo image pair we must be able to compare candidate points. Comparing candidate point is done by using a similarity measure $S(x, y, d)$. The outcome of this similarity measure tells us how much a given point in the reference image is similar to the point in the matching image. A distinction can be made between similarity measures that operate in the feature domain, Grimson [15], Labayrade and Aubert [29], measures that operate in the frequency domain, Jones and Malik [23], and measures that operate in the intensity domain. The most used intensity based similarity measures will be discussed below.

Intensity based

Over the years, a large collection of intensity based similarity measures have been proposed. For more information and a quantitative comparison between various intensity based similarity measure for computational stereo we refer to Banks and Corke [1], Roma et al. [51]. Most intensity based methods compute the similarity in a rectangular shaped region around the pixel of interest. These methods are referred to as window matching or block matching techniques.

The classical statistical method for determining similarity is Zero Mean Normalized Cross Correlation (ZM-NCC) also known as correlation coefficient. When using ZM-NCC with a correlation window of height $M=(2C_h)+1$ and width $N=(2C_w)+1$. We can compute ZM-NCC in the following manner:

$$ZM-NCC(x,y,d)=\frac{\sum_{w=-C_w}^{w=C_w}\sum_{h=-C_h}^{h=C_h}(I_l(x+w,y+h)-\bar{C}_l)\ (I_r(x+w-d,y+h)-\bar{C}_r)}{\sqrt{\sum_{w=-C_w}^{w=C_w}\sum_{h=-C_h}^{h=C_h}(I_l(x+w,y+h)-\bar{C}_l)^2\ \sum_{w=-C_w}^{w=C_w}\sum_{h=-C_h}^{h=C_h}(I_r(x+w-d,y+h)-\bar{C}_r)^2}} \quad (2.13)$$

$\bar{C}_l$ is the average intensity in a window centred around $I_l(x,y)$ with dimension *MxN*. And $\bar{C}_r$ is the average image intensity in a window centred around $I_r(x-d,y)$ with dimension *MxN*. ZM-NCC normalizes both in the mean and in the variance making it relative insensitive to radiometric bias and gain. Often a computational simplification is used called Normalized Cross Correlation (NCC).

$$NCC(x,y,d)=\frac{\sum_{w=-C_w}^{w=C_w}\sum_{h=-C_h}^{h=C_h}I_l(x+w,y+h)\ \ I_r(x+w-d,y+h)}{\sqrt{\sum_{w=-C_w}^{w=C_w}\sum_{h=-C_h}^{h=C_h}I_l(x+w,y+h)^2\ \sum_{w=-C_w}^{w=C_w}\sum_{h=-C_h}^{h=C_h}I_r(x+w-d,y+h)^2}} \quad (2.14)$$

NCC only normalizes in variance and not in the mean thus only compensating for radiometric gain. We know from the correlation theorem that a correlation between

f and h in the spatial domain is the same as a multiplication in the frequency domain between F^* and H. where F^* is the the complex conjugate of F. And F and H are the Fourier transforms of f and h. While it is not straightforward to apply this for normalized correlation, like ZM-NCC and NCC, approaches exist for efficient computation of ZM-NCC and NCC that make use of this correlation theorem, Lewis [34].

Because of the computational load of ZM-NCC and NCC other intensity based similarity measures are often used for real-time systems. The basis for many of these more efficient intensity based similarity measures are Squared Difference (SD) and Absolute Difference (AD). The use of AD is preferred because it requires less computation and memory.

$$SD(x,y,d) = \left(I_l(x,y) - I_r(x-d,y) \right)^2 \tag{2.15}$$

$$AD(x,y,d) = \left| I_l(x,y) - I_r(x-d,y) \right| \tag{2.16}$$

Using AD and SD the difference of a larger support region can be computed. The most common ones are Summed Squared Difference (SSD) and Summed Absolute Difference (SAD). Also normalized variations of SAD and SSD can be used.

$$SSD(x,y,d) = \sum_{h=-C_h}^{h=C_h} \sum_{w=-C_w}^{w=C_w} SD(x+w, y+h, d) \tag{2.17}$$

$$SAD(x,y,d) = \sum_{h=-C_h}^{h=C_h} \sum_{w=-C_w}^{w=C_w} AD(x+w, y+h, d) \tag{2.18}$$

Window matching measures like SAD and SSD can usually be separated in a single pixel similarity computation step and an aggregation step making them more efficient. The efficiency is gained by reusing and thereby reducing computations. This is achieved by using running block sums and Single Instruction Multiple Data (SIMD) instructions for cost aggregation, van der Mark and Gavrila [41].

One of the latest additions to intensity based similarity measures comes from Birchfield and Tomasi [3]. They propose a similarity method that is proven to be insensitive to image sampling. The measure does not use the intensity values themselves but instead uses a linear interpolated intensity functions surrounding the pixels. They note that the time needed to compute the measure is only 10 percent more that that of SAD. Using this measure it is possible to find the optimal disparity (maximum similarity) with sub-pixel accuracy. Note that sub-pixel accuracy can also be obtained by interpolating the similarity values inside the DSI instead, section 2.3.14.

2.3.5 Pre processing

The problem with un-normalized measures like SSD and SAD is that they are sensitive to radiometric distortion. Doubling the image intensities will double the dissimilarity. To overcome this problem we can filter both images with a Laplacian of Gaussian (LoG) filter, see figure 2.9. The LoG kernel is a zero sum kernel so the response of LoG filtering is zero on patches with equal intensity. Filtering with a LoG kernel makes the matching less sensitive to the intensity values themselves and more sensitive to the relative difference in local intensity values around a given pixel.

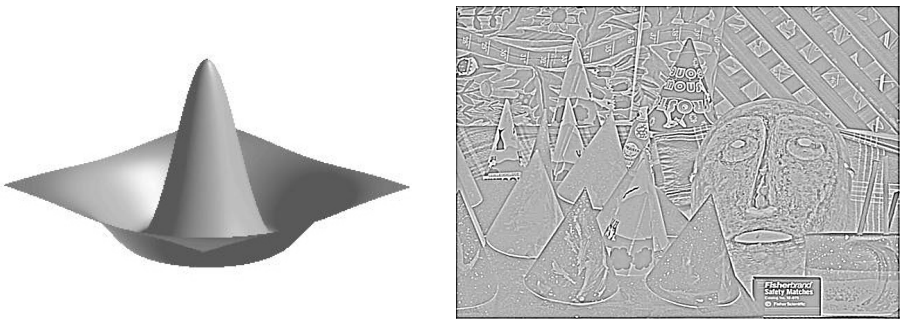


Figure 2.9: LoG kernel (left), LoG filtered image (right), original image figure 2.11.

Pre-processing steps that truly work with the relative ordering of intensity values in a window shaped regions are the Rank and Census transforms, Zabih and Woodfill [65]. The Rank transform of a pixel is defined as the number of pixels in a sub region around the pixel that have a smaller intensity. When an image is rank transformed we can use regular difference metrics like SAD to compute the difference between image regions.

$$Rank(x,y)=\sum_{h=-height}^{h=height}\sum_{w=-width}^{w=width} I(x,y)>I(x+h,y+w) \qquad (2.19)$$

$$Rank\begin{pmatrix}13 & 89 & 34\\ 12 & 34 & 23\\ 11 & 99 & 75\end{pmatrix}=4$$

The effect of the Rank transform as a pre-processing step is the elimination of sensitivity of radiometric gain and bias. Furthermore the authors claim that because the values are much more compressed the measure is more robust against outliers e.g. due to noise. An other benefit of Rank is that its output of a *HxW* window only takes $\log^2(H*W)$ bits. So a 16x16 Rank transformed window can still be encoded using an 8-bit unsigned integer. A drawback of the method is that a lot of discriminative information is lost. This is because the relative ordering of all pixels around a point is encoded in one single value. Zabihand and Woodfill [65] also propose a variation of the rank transform, called the Census transform.

$$Census\begin{pmatrix} 13 & 89 & 34 \\ 12 & 34 & 23 \\ 11 & 99 & 75 \end{pmatrix} = 100101100$$

The Census transform not only captures the number of pixel smaller than the centre pixel but also their location. This location is encoded in a bit string. The Census distance is then computed as the hamming distance (the number of bits that differ) between the Census transform of two regions. A Census transformed window of *HxW* takes *HxW-1* bits. So a 16x16 window would require 255 bits. This points out the main disadvantage of Census, it is very hard to optimize for processing on a standard CPU. In order the compute Census efficiently the use of programmable logic devices such as Field Programmable Gate Array's (FPGA's) are needed.

Banks and Corke [1], compare the performance of Rank and Census matching with those of correlation based difference metrics. Their results indicates that rank and census methods perform comparably to standard metrics and are more robust to radiometric distortion and occlusion. For many of the test scenes, the difference between NCC and census matching was between 5 and 9 percent of the total number of pixels. Unfortunately they did not measure the influence of LoG prepossessing on standard intensity based measures like SAD. Furthermore the performance was based on the number of pixels for which a disparity estimate was found. The actual value of the disparities was not used because the lack of disparity ground truth.

2.3.6 Similarity Aggregation

The quality of the found similarity measure is strongly influenced by the size and shape of the aggregation window. On one hand the aggregation window must be large enough to be distinctive from other windows. On the other hand the aggregation window must be small enough to only cover pixels at equal disparity, so perspective distortions are minimized. This problem is illustrated in figure 2.10. Similarity aggregation tries to capture the real-world relations between pixels. The performance difference between various disparity matching approaches is mainly due to the way they establish this relation between pixels and change the aggregation accordingly. Measures that use a fixed aggregation window like SSD, SAD and NCC are especially sensitive to the chosen window size, see figure 2.11. A large window size is more robust to noise but tends to dilate object edges in the image. On the other hand, a small window size can find depth disparity boundaries very precisely. However, it is more sensitive to noise, especially when image intensity variation is low. So the best window size and shape is a trade off and is based on local image characteristics. In the literature several techniques can be found to estimate the best shape and size of the aggregation window. First, we will list them here and in the next sections we will look into them more closely.

1) Adaptive aggregation windows.
2) Multi-resolution & Multi-scale.
3) Iterative aggregation.

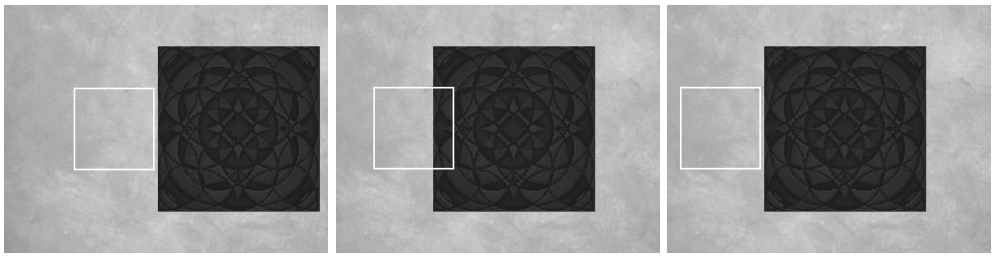


Figure 2.10: Reference window (left) Correct matching window (middle) Chosen matching window (right).

In the reference image we have a marble background with a black textured square in front of it, the reference window is shown as a white square (left). The pixel position associated with the reference window is visible in all images. Due to the projective distortion, the matching window at the correct disparity in the search image contains a part of the foreground object (middle). Because in contrast to the reference window the matching window contains pixels from different objects (and depth) the matching difference will be high. In stead of a clear minimum at the correct position, a faulty matching window that only contains the marble background will result in a better similarity value (right).

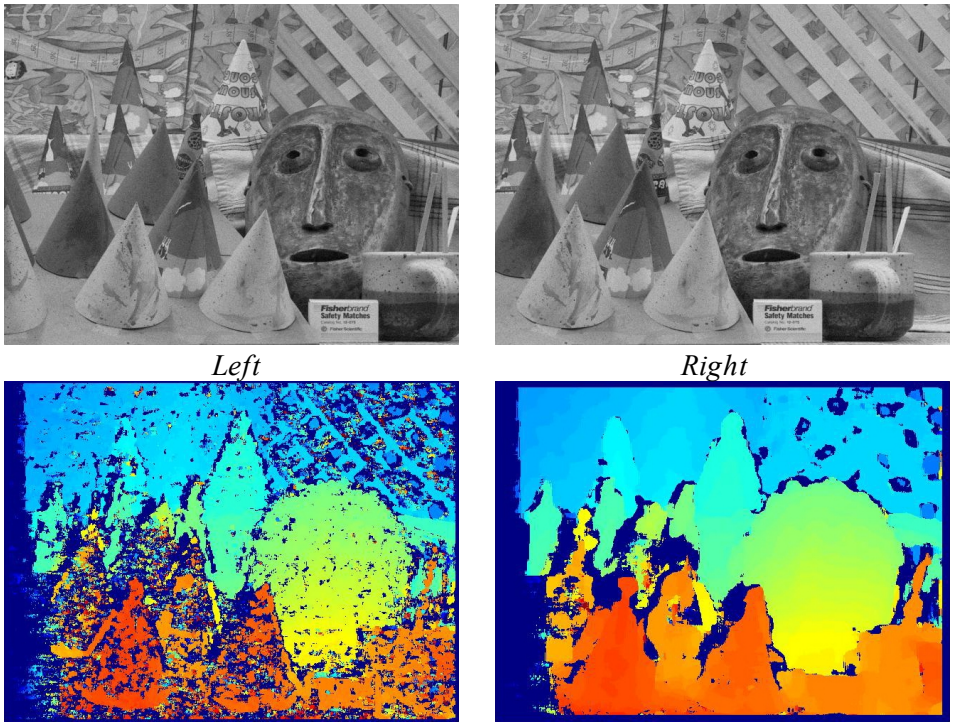


Figure 2.11: Effect of aggregation window size.

2.3.7 Adaptive aggregation windows

In the past years, the focus has been on finding techniques that find the optimal aggregation window size and shape for a given region in both the reference and matching images. The problem with finding the optimal window size and shape is that it depends on the image variance and disparity variance in the window. Unlike image variance, the disparity variance can not be known beforehand. While it is more intuitive to think of cost windows relative to the original stereo image pair, adaptive aggregation methods usually work on the data inside a DSI.

The first structural approach to solve this problem was presented by Okutomi and Kanade [24] [25]. Their approach tries to find the optimal window size and shape by measuring the image variance and estimating the variance in disparity for a given window. Due to the computational load they limit the window shape to be rectangular with controllable height and width. The key to their approach is; modelling uncertainty over the computed disparity. This uncertainty depends on the disparity variation and the image variation within a window. In this manner they can obtain the window size that locally minimizes the uncertainty over the computed disparity. The algorithm works in an iterative fashion starting with an initial disparity estimate and further refining the estimate until convergence. During each iteration and for each pixel the matching widow starts as a 3x3 rectangle. In the next steps the optimal window shape is found by expanding the window in a greedy manner. When for all pixels the window with the lowest uncertainty in its disparity estimate is found the disparities are updated and a new iteration is started.

The method has two disadvantages. Firstly it requires a large amount of computational power. Secondly it only refines an initial disparity estimate. However, the authors state that in some cases they can use an initial estimate for d of 0 when using a coarse-to-fine scheme. The exact conditions for the initial estimate are unknown. Furthermore, it is unclear if the success of the method is mainly from the iterative approach or from their window selection scheme. A limitation of this approach is that it restrict itself to rectangular windows. As the authors state, it would be possible to extend the window growing scheme in a pixel-by-pixel manner. This way arbitrary windows of various shapes and sizes can be formed.

A variable window technique that does facilitates windows of arbitrary shapes and sizes is presented by Boykov et al. [5]. The presented technique forms windows in a pixel by pixel manner. Their technique is based upon a plausibility measure. For every pixel in the reference image they construct a set of disparity values D which are plausible. The construction is based on the intensity of the reference pixel and the intensities of candidate matching pixels. Next, for every plausible disparity values $d \in D$, they grow a window starting at the reference pixel so that the window is the maximum connected set of all neighbouring pixels for which d is plausible.

Finally, they select that *d* for which the grown window is largest. This ensures that the chosen disparity value is most plausible in the neighbourhood of the reference pixel.

The drawback of these methods is that they are all computational expensive. To overcome this problem other methods use a finite set of predefined sub-windows. From this set of sub-windows we can construct a matching window that is most appropriate for the region around a reference pixel. The benefit of this approach is that it can be implemented as an adaptive summing block filter in the DSI. Which makes it possible to use running-block sums for efficient computation of the similarity values.

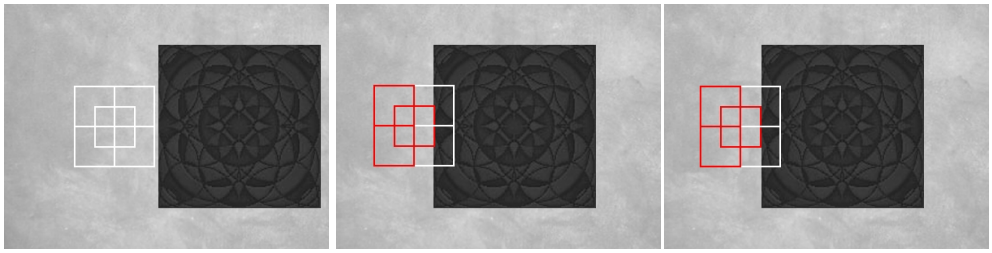


Figure 2.12: Multiple windows.

We have the same situation as in figure 2.10. However now because the matching window can select amongst smaller sub-windows it is able to come closer to the true disparity then when using a fixed window.

Bobick and Intille [4], Fusiello et al. [13], Kang et al. [25] use windows where the matching pixel can be at different locations inside the matching window and not necessarily the centre. Hirschmüller et al. [20], was able to develop a method that runs in real-time. The effect of this approach on the disparity search is illustrated in figures 2.12. For all sub-windows a correlation measure is computed. The final correlation measure is the sum of the three best matching sub-windows (given in red). By using the three best matching sub-windows we can find the disparity more precise as when using the complete matching window.

2.3.8 Multi-resolution & Multi-scale

Multi-resolution and multi-scale have strong resemblance with adaptive windows. Adaptive windows try to find the optimal cost window size and shape for stereo matching. Multi-resolution and multi-scale approaches try to find the optimal matching granularity of the image itself for all pixels in the image. Multi-resolution uses a fixed cost window size and uses multiple copies of the image at various resolutions, which can be visualized as an image pyramid, see figure 2.13. The traditional multi-resolution approach is an iterative coarse-to-fine method, O'Neill and Denos [48], Yang and Yuille [63], Yang et al. [64] and Roma et al.[51]. It first performs a full disparity estimate for the highest level in the image pyramid containing the lowest resolution. It then performs a disparity estimate one level down in the image pyramid where the result of the higher layer guides the disparity estimation process of the current layer. Thus the layers containing the lower resolutions provide the robustness and the layers containing the higher resolutions provide the accuracy. A well known problem with this approach is error propagation from the higher layers towards the lower layers. Because of the fact that the higher levels (lower resolutions) contain less information small objects might be overlooked and occlusions may be missed. These faulty estimates can propagate all the way down to the lowest level. Causing errors that could be avoided if the guidance of the higher layers was ignored.

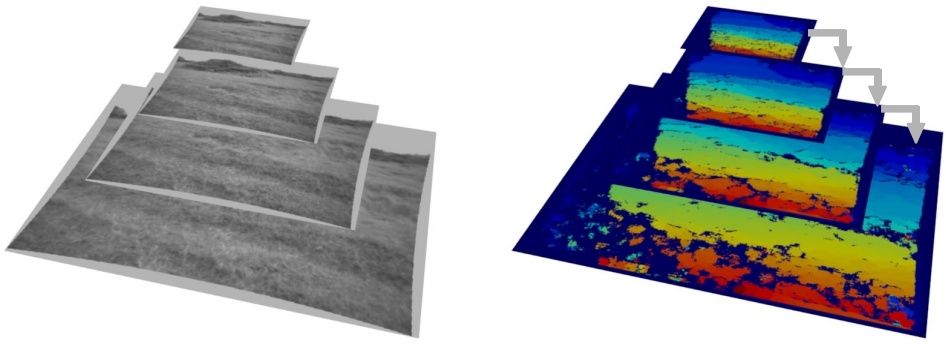


Figure 2.13: Multi-resolution disparity estimation.

Multi-scale approaches, Witkin et al. [62] are very similar to multi-resolution approaches. But instead of using an image-pyramid, multi-scale approaches use a scale-space, Witkin [61], Lindeberg [35], Koenderink [26]. A scale-space can be seen as an image pyramid where the resolution stays the same but each level is filtered with a Gaussian filter. The variance of the Gaussian kernel is increased for higher levels in the pyramid. One of the benefits is that by using a discrete set of Gaussian kernels at various scales a contentious scale-space can be approximated. Also tracking of features is easier because the resolution is equal at all levels. The drawback is that it is usually less efficient than multi-resolution approaches.

2.3.9 Iterative aggregation

Iterative cost aggregation or diffusion is the process of applying a fixed aggregation window on a filled DSI for a number of iterations. The number of iterations can be fixed or can be changed locally on a per pixel basis. For instance we can apply the following 4-connected Gaussian aggregation window (or update rule):

$$DSI(x,y,d) = (1-4\lambda)\,DSI(x,y,d) + \lambda \sum_{(m,n)\in N4} DSI(x-m, y-n, d)$$

(2.19)

Here, λ controls the speed of diffusion. A benefit of such an update rule over traditional window based aggregation is that the influence pixels have on each other decreases according to the distance between them (of course the same can be accomplished by using scale-space approaches or multiplying values inside an aggregation window with a Gaussian kernel). With diffusion an important question is how many iterations to use for any given pixel. Scharstein and Szeliski [54] experimented with three approaches of diffusion. Their first approach is based on a membrane model. Basically it uses the same update rule as above with an extra term to ensure that the new cost value does not deviate to much from the original coast value. Their second approach uses the update rule above with a local stopping criteria based on the validity of the candidate disparity estimate. The disparity validity measure used is based on the cost values inside a DSI column. For any given column they calculate its Winner Margin (WM), which is the normalized difference between the DSI column minimum and the second best minimum (see section 2.3.15). Then they apply the update rule above and again they measure the WM. If the WM decreased, they restore the old value and terminate the diffusion process locally otherwise they continue. Their third approach uses a Bayesian model for disparity matching. Their model consist of two parts. The first part is the prior model that captures the expected disparity smoothness in the scene. The second part is the data model which captures the intensity difference between the pixels in the stereo image pair. Each DSI column is represented with a probability function. Diffusion lets the column probability functions influence each other based on the prior model and the actual intensity differences. From the three approaches the Bayesian approach works remarkably better than the other two approaches. However, parameter choice is crucial especial the ones concerning the prior model which captures the expected disparity smoothness. The local stopping criteria and the membrane model had almost equal performance. Finally Scharstein and Szeliski believed their local stopping criteria approach could be improved using better validity measures instead of Winner Margin.

2.3.10 Disparity estimation & optimization

In the previous sections we have described the filling of a DSI with the output of a similarity measure. The next task is to extract the most representative disparity map from this DSI. This can be regarded as a cost minimization problem. The cost *C* of a candidate disparity map is the sum of its local cost C_l and its global cost C_g, see formula 2.20. The local cost is computed from the similarity values inside the DSI. A low cost means a high level of similarity. The global cost is made up of penalty terms computed from the candidate disparity map. For instance, we can associated high cost with large disparity jumps in the disparity map. Usually penalty terms are only computed for neighbouring pixels in the disparity map. The global term is used to enforce smoothness of the disparity map.

$$C(d)=C_l(d)+C_g(D) \qquad (2.20)$$

$$C_l(d)=\sum_{x=1}^{x=W}\sum_{y=1}^{y=H} DSI(x,y,d(x,y))$$

$$C_g(d)=\sum_{x=1}^{x=W}\sum_{y=1}^{y=H} Penalty(d(x,y),d(x-1,y),d(x+1,y),d(x,y-1),d(x,y+1))$$

The problem of disparity estimation then becomes finding the disparity map *d* with minimal cost from all possible candidates *D*.

$$d=argmin_{d\in D}\ C(d) \qquad (2.21)$$

The set of all possible candidates can be extremely large. For an image with size *HxW* and a maximum allowed disparity of Max_d the search space contains $Max_d{}^{HW}$ possible candidates. It is clear that computing the cost for all possible candidates is not practical. The task of disparity estimation can be seen as an ill-posed problem. We can only solve this ill-posed problem efficiently by limiting the search space and the output of our algorithm. These constraint are based on scene specific knowledge and assumptions about the influence that neighbouring pixels can have on each other. In the next sections we will describe different optimization methods.

2.3.11 Local Optimization

local optimization techniques only use a pixel's DSI column to estimate disparity. The basic thing we can do is to select that disparity with the lowest associated cost in the DSI column i.e. Winner Takes All (WTA). This method only optimizes the local term of formula 2.20 and neglects the global (smoothness) term. Local optimization methods can be computed extremely fast and are in favour for real-time systems. While smoothness is not enforced by the global term it usually is enforced using other techniques. Fore example by using cost-aggregation in the DSI, neighbouring pixels can still have some influence on each others disparity estimate (with scan-line or global optimization methods cost aggregation is usual an integral part of the optimization process itself). Also disparity post-processing, see section 2.3.15, can help to enforce the smoothness constraint.

2.3.12 Scanline optimization

Scanline optimization finds the disparity for a pixel depending on its entire DSI slice. The technique most often used is to represent the DSI slice as a Markov Random Field (MRF). A MRF is basically a 2D extension of a Markov chain, see figure 2.14. With Markov chains each node is only dependent on a limited number of its predecessors. With MRF's a node is only dependent on a limited number of its neighbours. MRF based disparity estimation can be done by finding an optimal path through the MRF using dynamic programming, Cox et al. [10], Bobick and Intille [4]. The low computational load of dynamic programming makes it an option for real-time systems, Kraft and Jonker [28]. We can model a DSI slice with a MRF in the following way. Each pixel in the DSI slice is modelled as a node, so we have $WxMax_d$ nodes per slice. Each node in the MRF has an associated cost $Cost_{w,d}$. Finding the optimal disparity for each pixel in the scan line involves two steps. First a forward cost aggregation step is performed. During this step the cost associated from a pixel can be influenced by a limited set of its neighbours, see figure 2.15. Second a disparity trace back step that finds the lowest cost path through the MRF.

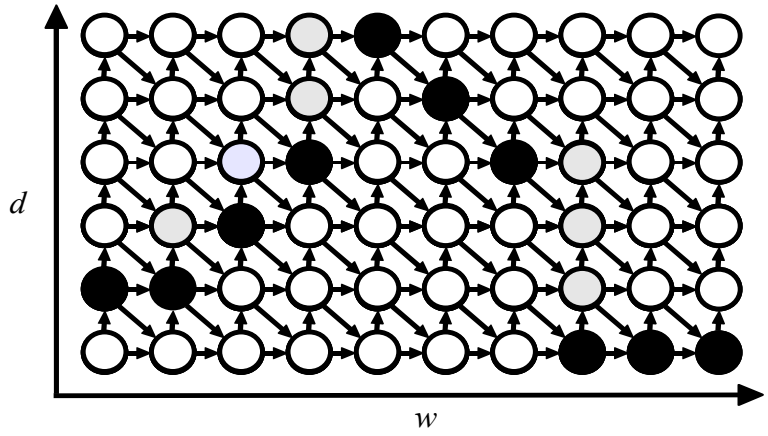


Figure 2.14: DSI based MRF with disparity path (grey and black nodes).

Lets assume we want to compute the disparity for image line *l* and use right-to-left matching. Then $S_{w,d}$ is the dissimilarity between $I_r(w,l)$ and $I_l(w+d,l)$, this value is stored in pixel *(w,d)* in the DSI slice. Also let $C_{discontinuty}$ be the cost associated with a discontinuity and $C_{occlusion}$ the cost of an occlusion. Their actual values are usually heuristically determined. The process starts with a clean MRF with as many nodes as there are pixels in the DSI slice. We start at the most left column *w=1* and set the cost of all nodes to the corresponding similarity values from the DSI slice.

$$Cost_{w,d} = S_{w,d} \quad , w = 1 \quad 0 \le d \le Max_d$$

Then we start visiting all other DSI columns sequentiality. We visit all nodes in a column by column manner, starting at the bottom *d = 0* and ending at the top $d = Max_d$, until we reach the most upper right node. When we visit a node we apply the following update scheme:

$$Cost_{w,d} = Min \begin{bmatrix} Cost_{w-1,d} + S_{w,d} \\ Cost_{w,d+1} + C_{occlusion} \\ Cost_{w-1,d-1} + C_{discontinuity} \end{bmatrix}$$

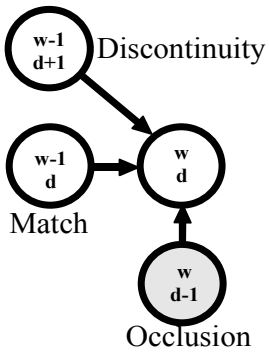


Figure 2.15: Allowed relations between MRF nodes.

We choose the minimum cost value of the three possible previous nodes. Also, the index of the node that supplied the minimal value is stored. Then during the trace back step we simply select that node from the last column with the lowest cost and follow the path using the indices's stored during the cost aggregation step. Bobick and Intille [4], use a similar technique and extend it by using ground control points. Ground control points are MRF nodes which are enforced to be on the path the disparity trace back step takes through the MRF. Ground control points can be high confidence matches. But they can also be edges in the intensity image. A well known artefact of dynamic programming is the streaking effect, see figure 2.16. It can be caused by favouring the match transition over the occlusion transition. This points out the difficulty of choosing the right costs associated with the discontinuity and occlusion transitions. The effect can be minimized by enforcing inter scan-line consistency, Ohta and Kanade [44].

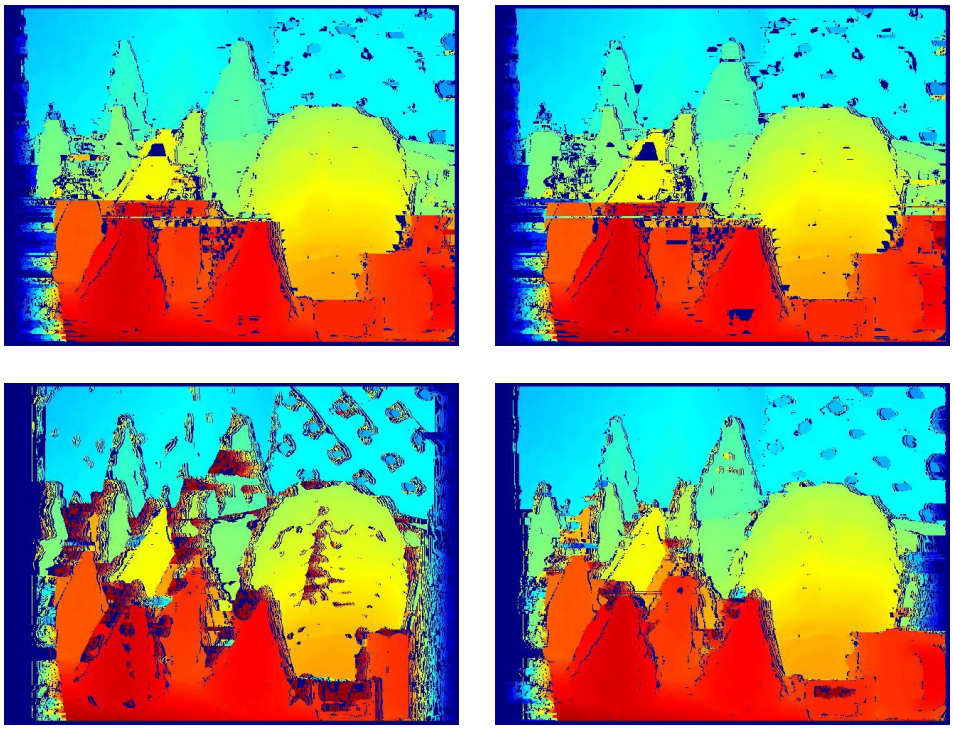


Figure 2.16: Effect of parameter choice on DP (original stereo pair fig. 2.11).

2.3.13 Global optimization

Almost all global optimization methods are based on Markov Random Fields. Now we model each pixel in the disparity map as a node and its possible disparities as node states. This way we can enforce intra and inter scan-line consistency. Finding the optimal disparity map now becomes a task of finding an optimal (state)membrane through the MRF. Usually this is done using Graph-Cut algorithms, Boykov et al. [6] , Kolmogorov and Zabih [27] or belief propagation, Sun et al. [57]. The results obtained with methods using global MRF disparity optimization are among the best performing techniques. The drawback is that the optimization methods involved are computational expensive and are therefore not suitable for real-time systems. While one can argue that the computational burden will pose no problem in a few years time. There is still a problem with the selection of parameters these methods need. These parameters encapsulate prior knowledge about the scene such as its smoothness. For the benchmark images of Scharstein and Szeliski [53] these parameters can be held constant. For a driving vehicle the scene can drastically change, from a flat dessert landscape to a rocky mountain pass for example. This causes the need for changing the parameters according to the properties of the scene. To the best of our knowledge, methods that can estimate the proper parameters settings for global disparity optimization for every possible stereo pair have not been proposed.

2.3.14 Sub-pixel accuracy

As mentioned earlier, sub-pixel disparity estimation can be obtained using two methods. First, we can interpolate the pixels in the image itself and compute the cost using sub-pixel displacements, Birchfield and Tomasi [3]. Secondly, we can interpolate the cost values inside the DSI, see figure 2.17. For local optimization the cost column is shown below. The points are the computed cost values with pixel accuracy. By fitting a curve through these points we can acquire a minimum with sub-pixel accuracy. Usually we first find the minimal cost value with pixel accuracy i.e. *d*. Next we fit a parabolic curve through the point left C_{d-1} and right C_{d+1} of the minimum C_d. Then we find the minimum of this curve with sub-pixel accuracy using formula 2.22. It is common to use a sub-pixel bound which limits the allowed deviation between the pixel accurate and sub-pixel accurate estimates. Using sub-pixel accuracy can reduce depth uncertainty, section 2.2.4.

$$d_{subpixel} = d + \frac{C_{d-1} - C_{d+1}}{2(C_{d-1} - 2C_d + C_{d+1})} \qquad (2.22)$$

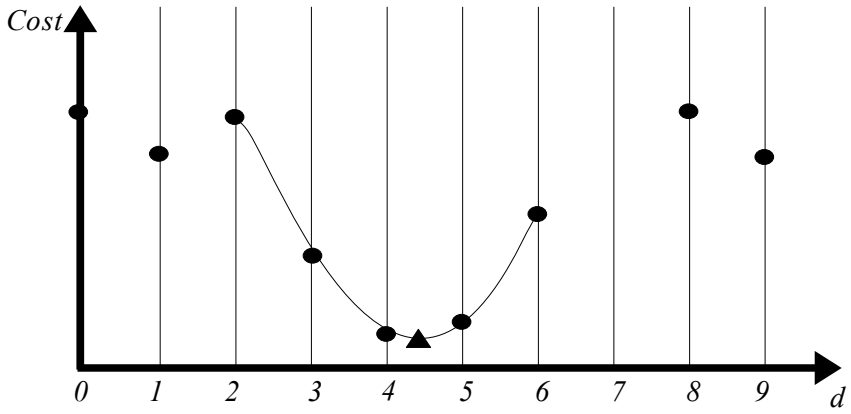


Figure 2.17: Sub-pixel accuracy (triangle), pixel accuracy d = 4.

2.3.15 Quality of disparity Estimate

An important aspect of any stereo algorithm is determining the quality of the found disparity estimates. For this we must construct a measure that reflects how likely it is that the found disparity matches the true disparity. This confidence or validity measure indicates how much we can trust the found estimate. The measure can be used to reject estimates if necessary. However, it also can guide the disparity search itself, Scharstein and Szeliski [54]. Banks and Corke [1] give an overview of used validity measures an compare their effectiveness. Basically, validity measures can be subdivided in three categories. There are measures that are based on the disparity estimates themselves, there are measures that work on the image intensity values and measures that work on the computed cost values. Below we will describe some examples of each category.

disparity based

A common method to reject bad disparity matches is the left-to-right consistency check. What it enforces is that when matching from left to right we must obtain the same disparity output as when matching from right to left. Pixels for which the left-right estimate differs from the right-left estimate are said to be faulty and therefore discarded. Another frequently used method is that of blob filtering Matthies et al. [38]. Blob filtering segments the image. The connectivity between pixels is based on their disparity difference. A pixel will be grouped to one of its neighbours if its four connected and the disparity difference is lower than a certain threshold. This way we obtain image segments in which the disparity is smooth. Next we remove segments with a pixel count smaller than a threshold. Finally disparity smoothness can also be enforced on a per pixel basis. We can use disparity median filtering or set a maximum disparity difference that a pixel can have from its neighbours. With the measures above the confidence measure and the process of disparity rejection are integral parts of each other and can hardly be separated.

Intensity based

Intensity based measures reject disparity estimates for image regions with bad signal to noise ratio. For example image regions with no texture are likely to produce bad matches. We can use the variance in intensity or the difference in the minimum and maximum intensity value as an estimate for the signal to noise ratio. An other metric often used is Moravec's 'Interest Operator' [42].

Cost metric based

The most used validity measures is the winner margin WM_1, Scharstein and Szeliski [54] or related methods WM_2, Hirschmüller [20]. Both are based on the values in the DSI column for a given image location. The winner margin is the normalized difference between the cost values of the best match C_{min} and the second best match C_{min2} in a DSI column.

$$WM_1(x,y) = \frac{C_{min\,2} - C_{min}}{\sum_{d \in D} DSI(x,y,d)} \qquad (2.23)$$

$$WM_2(x,y) = \frac{C_{min\,2} - C_{min}}{C_{min}} \qquad (2.24)$$

$$WM_3(x,y) = \frac{C_{min\,3} - C_{min}}{C_{min}} \qquad (2.25)$$

The problem with WM_1 and WM_2 is that the minimal coast value can be between two pixels. In that case the best and second best match in the DSI column will be vary close to each other, see figure 2.17. This will result in a poor winner margin while the actual found minimum is correct. This is why Mühlmann et al. [43] argue it is better to use the third best match C_{min3} together with the best match, see WM_3. According to Banks and Corke [1], left-to-right consistency checking is most effective on occluded regions. Whereas intensity and cost based measure are more effective at removing estimates from bland image areas. Because ground truth data was not available for their stereo image pairs quantitative analyses were lacking. Unfortunately, apart from the paper by Banks and Corke [1] there is little research into the effectiveness of different validity measures.

2.3.16 Night-time Stereo

The methods discussed so far were all intended for generic stereo images. Unfortunately the published literature on Night-time stereo is very limited, almost all of the work can be traced back to the Jet Propulsion Laboratory of NASA. Owens and Matthies [49] investigated the suitability of passive night vision cameras for stereo vision. CCD cameras equipped with third generation image intensifiers and various cooled and un-cooled thermal imaging cameras where tested. First they looked into several hardware issues such as signal to noise ratio, exposure time and synchronization options. Next the possible use of these cameras for night-time stereo was investigated. Using the cameras, false stereo image pairs of various night-time scenes were created. A false stereo pair consists of two images taken by the same camera from exactly the same location right after each other. One of these images is shifted horizontal. So the disparity is constant throughout the entire image. Their standard disparity estimation technique was applied to the false stereo image pairs. Because the ground truth disparity is known the variation in estimated disparity can be measured. High variation in estimated disparity indicates poor performance. Also the percentage of pixels that passed the left-to-right consistency check was measured. Their results pointed out that only the cooled and un-cooled thermal camera systems reached acceptable performance. The best performance (variation 0.06 and coverage 99.9%) was obtained with a cooled thermal imaging camera. The image intensifiers did not nearly reach the needed level of performance. We have to note that the objective of their research was to investigate the suitability of different cameras and compare them against each other. The performance on the false stereo pairs, hardly is an indication of the true disparity estimation performance. In an other publication, Matthies et al. [39] and Bellutta et al. [2], the comparison between several night-time camera methods was based on obstacle detection. The result were promising but substantial quantitative results were lacking.

Apart from hardware issues such as sensor sensitivity, exposure time and synchronization options there are more fundamental problems for night-time stereo. Using thermal imaging the contrast in the image is based on the relative difference in temperature between various objects in the scene. This relative temperature can change during the night. For instance a rock can have good contrast just after sunset. But as the rock cools through the night it will reach a temperature near that of the surrounding soil. This causes low contrast between the rock and the soil making the rock hard to detect. For other approaches such as active lighting, contrast and exposure time is an important issue. Because the light is emitted from few positions on the vehicle we have strong directional lighting. This directional lighting and the lack of ambient light sources can cause strong highlights on objects which can change according to the camera position. Furthermore because the light is directional we have shadows in regions otherwise lit by ambient light. This contrast between highlights and shadows makes it difficult to have a good signal to noise ratio throughout the entire image. Exposure time is an

important aspect as well. Increasing the exposure time increases the signal to noise ratio but will also cause motion blurring. To the best of our knowledge no existing work on appropriate disparity estimation techniques for low visibility conditions such as during the night have been published.

2.4 Obstacle detection algorithms

So far we have described various methods that enable 3D reconstruction of points seen through a binocular camera system. The next step is to use this 3D information to detect obstacles in front of the vehicle. Obstacle detection (OD) is the task of distinguishing drivable terrain from un-drivable terrain. Most of the existing OD systems rely on stereo vision or LIDAR to estimate the geometrical properties of the terrain in front of the vehicle. Based on the estimated geometrical properties of the terrain we can search for possible obstacles. There is a difference between OD for structured terrain and unstructured terrain. With structured terrain such as highways and urban roads we assume that the terrain is relatively flat, this is called the flat world assumption. Often methods for structured terrain look for the road or ground surface first. Then the height of image points can be compared against the ground plane. Image points with a height that deviates from the ground plane more than a certain threshold will be labelled as an obstacle. With unstructured terrain the flat world assumption is not always possible. Most methods for unstructured terrain therefore first detect obstacles without any ground plane assumption. Once the obstacles have been found, the ground plane can be computed as well. Methods that work for unstructured terrain can also be used for structured terrain. The other way around is not straightforward. In this chapter we will look into OD methods that have been used in unstructured terrain. Most OD approaches found in the literature can be categorized in four classes:

1) Column based slope analysis
2) 3D clustering
3) V-disparity
4) Elevation maps and Voxel maps

In the coming section we will describe each approach in more detail. Detection of an obstacle is only helpful if it is done in time so that the vehicle can stop or change its path. The time a vehicle needs to stop depends on its speed and the friction between its tires and the ground. Formula 2.26, Matthies and Rankin [40], relate the distance D it takes to stop the vehicles to its current speed v.

$$D(v)=\frac{v^2}{2\mu g}+v T_r+B \qquad (2.26)$$

Here μ us the friction coefficient, g is the gravitational acceleration, T_r is the total reaction time and B is a safety bound. For typical off-road operations these values are $\mu=0.65\,m/s^2$, $T_r=0.25\,s$, $B=2\,m$ and of course $g=9.81\,m/s^2$. This formula tells us at which distance we absolutely have to detect an obstacle for save avoidance when driving at a certain speed. For instance when driving at 30 kph the save detection distance is about 10 meters.

2.4.1 Column based analysis

An approach often used for positive obstacle detection in unstructured terrain is image-column based analysis. Three important steps can be distinguished. Firstly, for each image column the slope at each pixel is determined. Then two separate thresholds are applied. One on the measured slope and the other on the height displacement over which the slope was measured. The next step is to filter the obstacle map according to specified obstacle criteria. The effect is that for instance small obstacles, that most likely are false alarms, will be discarded. The advantage of this approach is that is it relatively fast and can run in real-time.

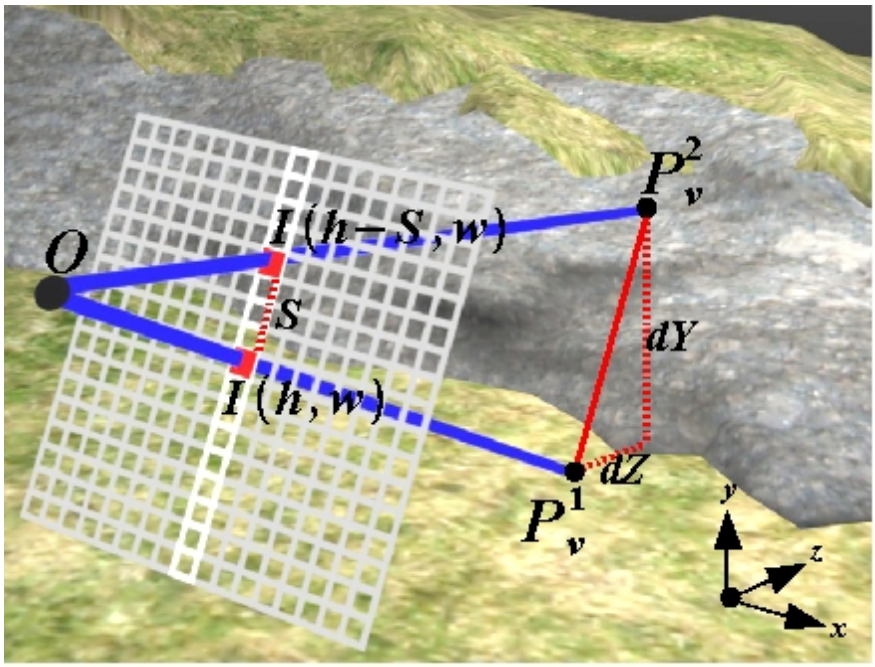


Figure 2.18: Columns based slope estimation.

The terrain slope at a given pixel $I(h,w)$ is defined as the local difference in height dY divided by the difference in depth dZ. To measure dY and dZ we need two points in the scene P_v^1 and P_v^2 given in the vehicle coordinate system, see figure 2.18. The projection of P_v^1 onto the imaging plane is given by $I(h,w)$. The projection of P_v^2onto the imaging plane is given by $I(h-S,w)$. The difference in image height between $I(h,w)$ and $I(h-S,w)$ is called pixel step size and denoted by S. For each pixel in the image we must choose an appropriate S. If we have chosen S we can establish the image points $I(h,w)$ and $I(h-S,w)$. Using the vehicle coordinates P_v^1 and P_v^2 of the chosen pixels $I(h,w)$ and $I(h-S,w)$ we can calculate the terrain slope by:

$$Terrainslope = \frac{P_{Vy}^2 - P_{Vy}^1}{P_{Vz}^2 - P_{Vz}^1} \qquad (2.27)$$

The difference in most column based obstacle detection methods is in the way they choose S.

In this section we will focus on an approach presented by Matthies et al. [37] [38] [39] Their method assumes that positive obstacles resemble a vertical step in height on an other wise flat ground plane. Their algorithm visits every pixel *I(h,w),* for which a depth estimate was found and assumes it is the start of an object of height *H*. They convert *H* to a pixel displacement *S* and use *I(h,w)* and *I(h-S,w)* to compute the difference in estimated height *dY* and depth *dZ*. Because *dZ* and *dY* are highly correlated the decision rule is only based on *dY*. Namely; If the difference in height at the given pixel exceeds a threshold *t*, label the whole section from *I(h,w)* to *I(h-S,w)* as a positive obstacle. Finally the positive obstacle map is filtered with a blob-filter. The connectivity criterion is 4-connected and blobs are rejected if their size in pixels is less than a threshold value. In their early publication, Matthies et al. [37], the conversion between *H* and *S* was done based on a flat world assumption. More precisely, they assumed the depth in each horizontal scan-line is the same. Thus the pixel displacement *S* caused by an obstacle of height *H* is also the same for every horizontal scan-line (where *S* is large at the bottom of the image and small at the top). This way a table was created which gave for each horizontal scan-line one value for *S*. From later publications, Matthies at al [39], Bellutta et al. [2], it seems likely that they extended the algorithm to uses a relative displacement *S* that depends on *H* and the estimated depth of *I(h,w)*.

The problem with column based methods method is that the true slope of a positive object can be greater than the measured slope. This caused by the fact that the plane defined by *O*, *I(h,w)* and *I(h-S,w)* can intersect with the object under various angles. When the surface normal of the object is parallel to the slicing plane the slope will be correctly estimated. In the situation that the surface normal deviates from the slicing plane the estimated slope can be considerable less than the true slope, see figure 2.19. An approach that tries to overcome this shortcoming is discussed in section 2.4.2.

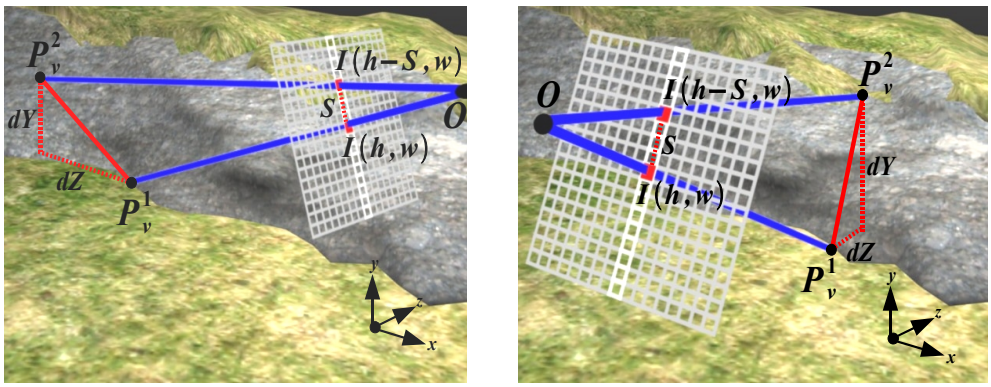


Figure 2.19: Influence of viewing angle on estimated slope. Estimated slope is less than true slope(left), Estimated slope is equal to the true slope (right).

The detection of negative obstacles is inherently more challenging than for positive obstacles. This is caused by the fact that the negative obstacle itself is not always visible. Only a depth discontinuity in the range profile of an image column indicates its presence. To show the intrinsic differences between positive and negative obstacle detection we will look into their geometry. A good indication about an obstacle's detection is the angle θ it creates at the sensor, Matthies and Rankin [40]. When, the distance to the object R is relatively large compared to the height of the sensor above the ground plane H, the angles θ and α will be small. Hence we can use small angle approximation to find their value. For positive obstacles we find $\alpha \approx (H-h)/R$ and $\alpha+\theta_p \approx H/R$ and therefore

$$\theta_p \approx \frac{h}{R} \tag{2.28}$$

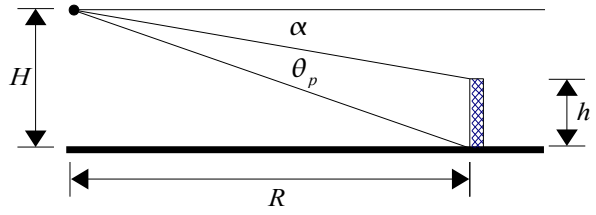


Figure 2.20: Positive obstacle.

For negative obstacles we have $\alpha \approx H/(R+w)$ and $\theta_n \approx H/R$ therefore

$$\theta_n \approx \frac{Hw}{R^2+Rw} \tag{2.29}$$

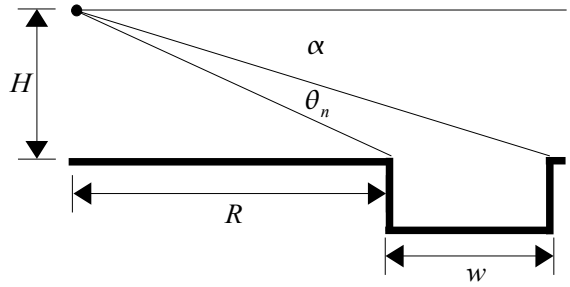


Figure 2.21: Negative obstacle.

This shows that for positive obstacles the angle θ_p decreases with a factor $1/R$ as the range increases. For negative obstacles the angle θ_n decreases with a dominant factor $1/R^2$. This points out that reliable detection of negative obstacles requires significant more image resolution than what is needed for positive obstacles. Narrow field of view (FOV) stereo camera systems mounted on a pan tile unit are often used for the special purpose of negative obstacle detection, Matthies et al. [38] [39] and Bellutta et al. [2]. This allows the system to focus on small parts of the terrain with a high resolution.

We will now describe a negative obstacle detection method proposed by Matthies et al. [38]. Their approach searches for the near-side of a negative obstacle in the following manner. For a given image point P_v^3 they fit a line through points below it in the same image column. In this manner the local ground plane can be obtained. Based on this local ground plane the expected depth of the image point above P_v^3 can be computed. If the measured depth of P_v^4 and the expected depth E_v^4 deviates more than a threshold, P_v^3 is labelled as a negative obstacle.

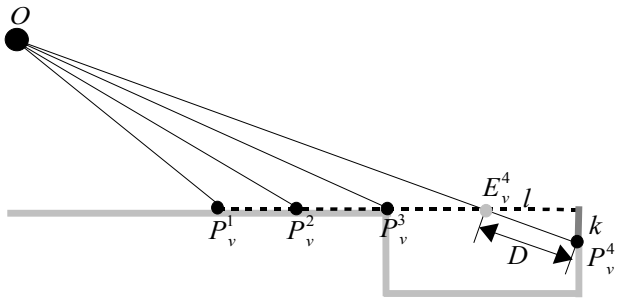


Figure 2.22: Negative obstacle detection.

In a later publication, Matthies et al. [39], they adapted their approach in the following manner. The ground plane is based on a fit though P_v^3 and all image points below it in the same image column. Next they fit a line starting from P_v^3 to the apparent bottom of the ditch. If the angle between the two line segments is greater than a threshold and the width (distance from P_v^3 to P_v^4) of the candidate negative obstacle exceeds a threshold, P_v^3 is labelled. When this is done for all image points the initial negative obstacle map is filtered using a blob filter. Because the candidate negative obstacles are horizontal-line image features the connectivity criterion is 8-connected. The blobs are rejected when their horizontal image width is lower than a particular threshold. Finally in a publication from the same research group, Bellutta et al. [2], an extra criterion was added. Apart from the just described negative obstacle detection they also demand the detection of a vertical edge *k* on the far far side of the negative obstacle. Where the height of *k* must be larger than a depth dependent threshold.

It is generally understood that negative obstacle detection is one of the most challenging tasks for autonomous land vehicles. Geometrical approaches as just described have some fundamental shortcomings. In a recent publication by JPL, Matthies and Rankin [40], negative obstacle detection during night-time conditions is enhanced using thermal imaging. It is based on the fact that negative obstacles at night have characteristic thermal signatures. This is caused by the fact that the bottom of a negative obstacle stays considerable warmer during the night than the surrounding terrain. While the usage of their approach is primarily during the night they state that modelling solar illumination has potential to extend it to day-time conditions.

2.4.2 3D Clustering

A recent method for positive OD uses clustering of pixels that are likely to belong to an obstacle, Talukder et al. [58] and Manduchi et al. [36]. This clustering is based on the 3D coordinates of the pixels. However, regions of interest in the image are exploited to make it computational efficient. The technique is based on so called compatible points. Compatible points are defined as:

Definition 1: Two surface points p1 and p2 are called compatible with each other if they satisfy the following two conditions:
1) their difference in height is larger than H_{min} but smaller than H_{max}
2) the line joining them forms an angle with the horizontal plane larger than θ_{max}.

Definition 2: Two points p1 and p2 belong to the same obstacle if:
1) they are compatible with each other, or
2) there exists a chain of compatible point pairs linking p1 and p2.

Definition one is used to find isolated obstacle points. Where H_{min} and H_{max} are both defined in meters and represent the granularity at which we search for obstacle points. The slope an object must have to be considered an obstacle is defined with θ_{max}. Definition 2 is used to group obstacle points together. In this manner we find isolated obstacles for which we can estimate their height, width or depth. This is useful for discarding false detections and for reasoning about terrain traversability. The naive way to find all obstacle point would be to compare if the conditions for compatibility are met for every point pair. This would require $O(N^2-N)$ operations. Where N is the number of points in the image for which range has been estimated. The ingenuity of their technique lays in the fact how they reduce the search complexity. Given a point together with all compatible points forms two truncated cones in 3D space, as can be seen in fig 2.23. The projection of these cones are two triangles in the image. This means that if we know the depth of the point under investigation we can compute the projection of the two truncated cones which are the two triangles. Thus we only have to search in the regions defined by the two triangles for compatible points. Because the OD considers triangular image areas instead of columns it is less sensitive to the angle between the surface normal and the optical axis. While this is an improvement over column based analysis, also the increase in computational load is significant. Even near real-time frame-rates were not achieved despite a c/c++ implementation on modern computer hardware while using a 320x240 image resolution.

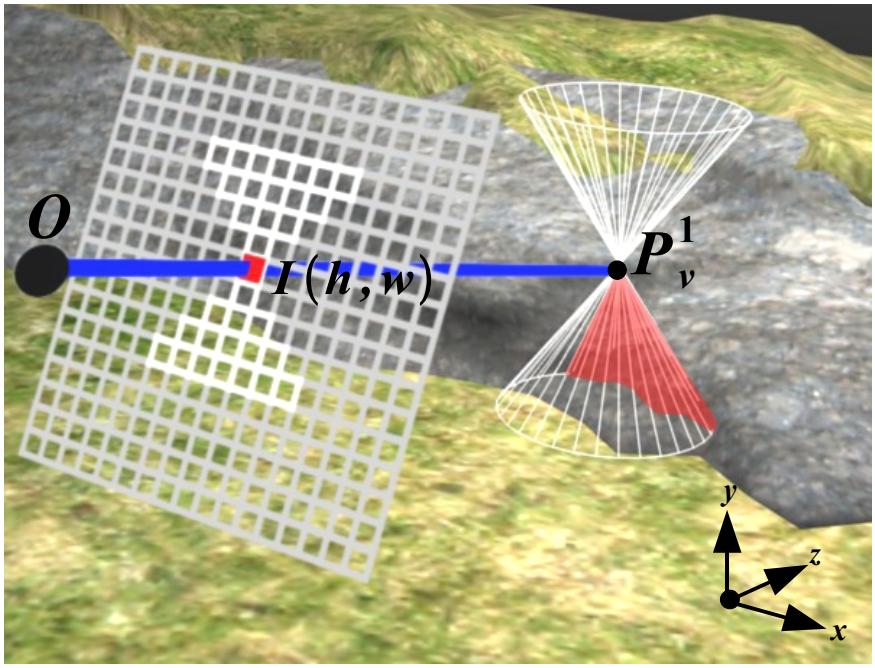


Figure 2.23: Truncated cones approach. Truncated cones are projected to two triangles in the image. red terrain is compatible with P_v^1 .

The assumption is made that the vehicle has zero roll and pitch. Of course this is a very brittle assumption in off-road terrain. The authors mention that when using an IMU to estimate the vehicles roll and pitch the process of forming the truncated triangles can compensate for the vehicle misalignment. However, in their current work they use truncated triangles that are slightly bigger than would be justified by their used thresholds. This way the system is more robust against the vehicle's orientation.

2.4.3 V-Disparity

The V-disparity obstacle detection technique proposed by Labayrade, et al. [30] is based on a V-disparity map, see fig 2.24.c. A V-disparity map basically is a collection of disparity histograms, one for each scan line. A V-disparity map accumulates pixels for every horizontal scan-line and every possible disparity value in the disparity map. Every V-Disparity bin holds the number of pixels coming from that particular line and with that particular disparity. In the V-disparity map 2.24.c below the brightness of a pixel <*d,v*> relates to the amount of pixels that had a disparity *d* at the horizontal image line *v* in the disparity map 2.24.b. A V-disparity map allows us to find dominant features in the disparity map based on their disparity.

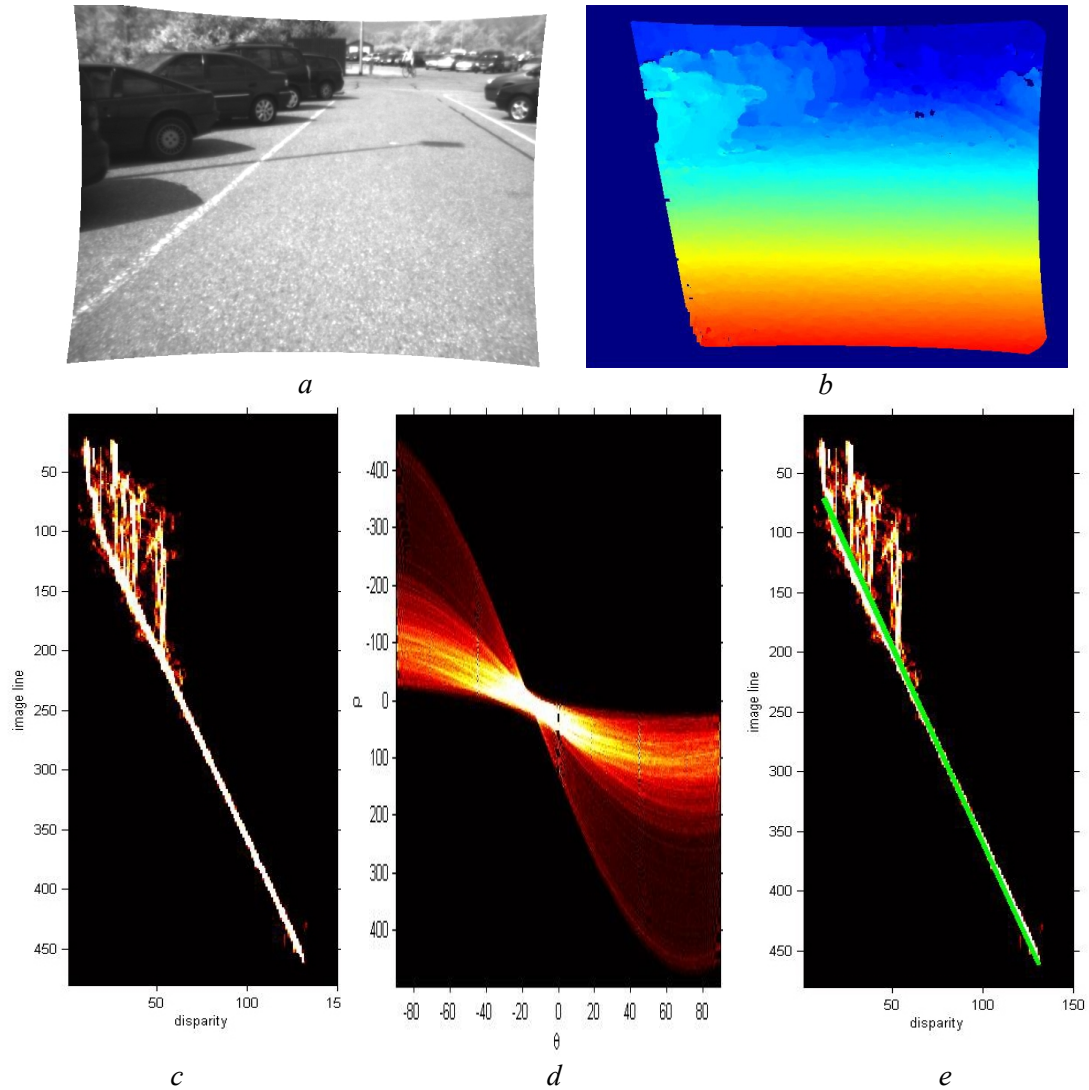


Figure 2.24: Original image (a), Disparity map(b), V-Disparity map(c), Hough transform(d), V-Disparity with extracted ground plane (green) using Hough transform(e).

The construction of a V-disparity map can be done very fast. It requires that every pixel in the disparity is visited once and accumulated in the corresponding bin, this can be done in *O(HW)* operations. The next step is to estimate the ground plane. Labayrade et al. [30] model the ground plane as a succession of parts of planar planes. These planes manifest themselves in the V-disparity map as a piecewise linear curve. This piecewise linear curve can be found using Hough transform, Hough [21]. The search can be speed up and made more reliable by reducing the parameter search space to predefined road orientations. Once the road profile has been found we know for every image column the disparity associated with the road. It is then straightforward to mark pixels that deviate from this road disparity more than a certain threshold. Finally, the marked pixels can be filtered to form consistent patches relating to obstacles in the scene. Work by Broggi et al. [7] shows that it is possible to use the V-disparity method in unstructured terrain. However, the pictures they present come from desert scenes where the ground is relatively flat compared to other outdoor scenes. Research into the applicability of V-Disparity based OD (especially for negative obstacles) under a wide range of terrain types is to our knowledge still an open issue.

2.4.4 Elevation maps & Voxel-based representation

The idea behind OD approaches using elevation maps, Hebert [18], and voxel maps, Lacaze [31], is that the world is subdivided in discrete regions. For elevation maps these are two dimensional squares and for voxel based methods these are three dimensional cubes. We can label each discrete region be it a square or a cube with certain properties. For instance, elevation maps assign each square with an estimated height. In voxel based methods each cube is assigned a label that reflects its solidity. Note that elevation maps and voxel maps are in principal not obstacle detection methods. They only offer a data structure in which a sensed map of the world can be stored. Based on this map we can reason about terrain traversability. Basically this is done by placing a 3D model of our vehicle in the world model and verifying that all four wheels are touching the ground. Because the functionality of these methods is much more than just obstacle detection they also require considerable more computations. A more fundamental challenge is that traditional elevation maps or voxel maps are isometric meaning that each cell in the data structure represents the same size in the world. Sensory input from stereo vision or LIDAR typically has high resolution near the sensor and less resolution further away. This makes using isometric data structures for storing stereo or LIDAR sensory output suboptimal. Considerable research has been dedicated into efficient data structures, Samet [52], and special purpose data structures Lalonde et al. [32]. Another challenge arises from the fact that the world models these methods create are usually accumulated over a lager time span. This causes the need for tracking the movement of the vehicle. When there are imprecisions or uncertainties in the movement of the vehicle according to the world, the world model itself may become inconsistent. Techniques that deal with tracking of a vehicle's movement and at the same time build a world model are referred to as Simultaneous Localization And Mapping (SLAM). SLAM is a popular topic in AI research and has received a great deal of interest, Thrun et al. [59]. However, the topic is not considered in this literature overview.

2.5 Limitations and Solutions

As the figures below illustrate, stereo based geometrical obstacle detection has its limitations. The problems shown are inherent and can only be solved using other techniques that enable us to increase the perceived scene semantics. Objects can be concealed by vegetation like tall grass. While a human might detect small patches of the rock through the grass, the disparity estimation will be biased towards the contour of the vegetation. This will cause the OD system to assume no obstacle is present. On the other hand, vegetation like tall polls of grass can also cause false detections. Again, a human will recognize the tall poll of grass but the OD system is limited to the estimated geometrical slope of the scene. Negative obstacles pose some difficulties by the fact that they are not always visible. Most of the time we can only mark terrain features where negative obstacles are likely. The true nature of the negative obstacle can only be established from close range. Furthermore, negative objects filled with water pose some interesting difficulties by themselves. Several methods exist to increase the knowledge about objects in the scene (in the forth coming section we focus on passive measuring methods and neglect active ones such as LIDAR and RADAR).

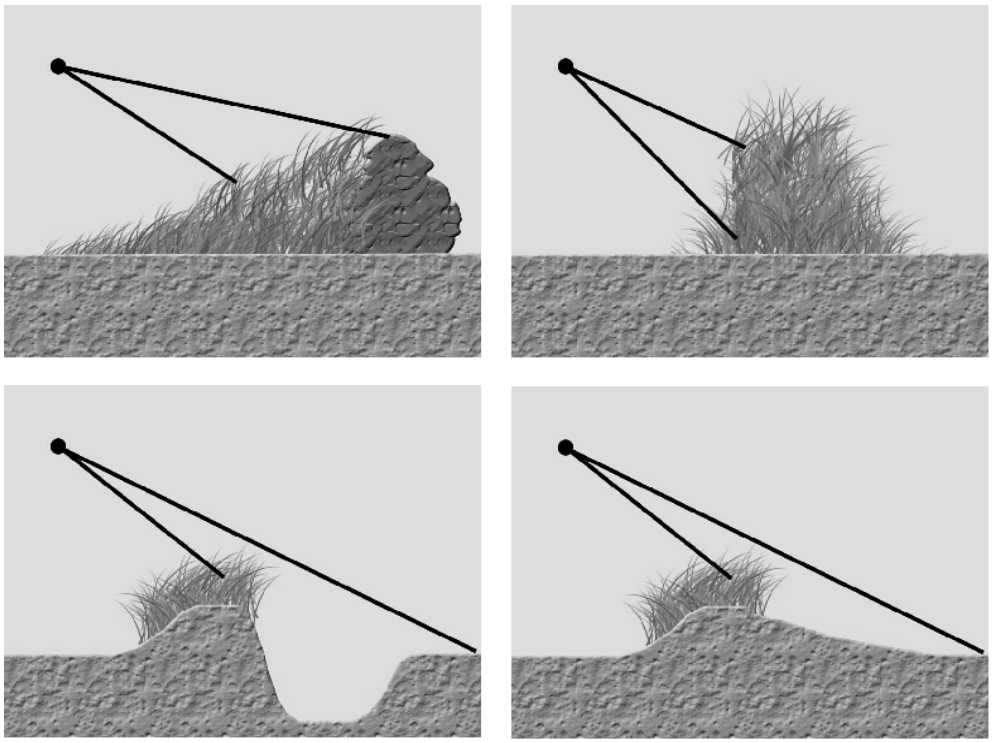


Figure 2.25: Limitations of stereo based OD.

2.5.1 Colour based terrain classification

A method to increase the knowledge about an object is looking at its colour. The colour of an object perceived trough a camera depends on several factors. Aspects such as the reflectiveness properties of the surface, the intensity and the chromatic composition of the light source play an important role. Research shows that robust colour based terrain classification requires invariance to the light source intensity an chromaticity, Jansen et al. [22], Bellutta et al. [2]. In outdoor terrain the chromaticity of the light source is mainly dependent on the weather conditions that influences the sun light. However, the ambient light chromaticity can also vary locally in the scene through the interplay of light beams reflecting of coloured surfaces in the terrain.

The method proposed by Jansen et al. is based on a maximum likelihood classifier using Gaussian Mixture Models (GMMs). First for every image in their train-set a GMM is estimated. These GMMs are used to find sets of images that have the same environmental state. By dividing the train-set set over these states, environmental specific GMMs for different terrain types can be trained. In this manner they can model the different chromatic properties due to environmental effects on different terrain types. A pixel can then be compared against the mixture models of all terrain types under all environmental states using their maximum likelihood classifier. In this manner they can reliably distinguish between terrain types based on its colour under a wide variety of outdoor illumination conditions.

Of course, the drawback of using colour based classification is that it can not be used at night when the scene is lit by a mono chromatic light source, such as near infra-red LEDs, or seen through thermal imaging camera's. It is known that vegetation can have a distinctive response in the near infra-red, Elachi [12]. Matthies et al. [38] suggest that this can be exploited for vegetation classification. However, their proposed method relies on the difference in the reflectiveness of foliage in the red band versus the reflectiveness in the near infra-red band. They measure both by using band pass filters for red light and one for near infra-red light in front of their cameras. A clear distinction in pixel intensity is highly correlated to the pixel being part of foliage. Robust foliage detection using mono chromatic light for autonomous land vehicles is to our knowledge still an open issue.

2.5.2 Texture based terrain classification

A more suitable method during night conditions might be texture based terrain classification. For a overview and comparison of existing texture classification methods we refer to Randen and Husoy [50]. A method that is applicable for real-time implementation is presented by Laws [33]. Laws' approach to texture classification is filtering the image with special texture detection filters. For instance filters for vertical bars, horizontal bars, waves, spots etc. can be used. Next the filter response can be accumulated for a support region and be normalized. Normalisation will make the filter response independent from the local image intensity. Finally for each pixel the texture filter responses can be stored in a vector for subsequent classification.

Texture classification usually suffers from scale and orientation problems. The same object will have a texture at different image scales when seen from different distances. Objects, like grass polls, can have the same texture but with different orientations. Here also the vehicles angle is an important aspect. The scale problem and orientation problem make it difficult to use textures based terrain classification. Unfortunately the ability to detect fine textures such as grass requires large image resolution at intermediate distances. The scale problem can be solved when we know the depth of objects, Olson [46] [47]. Olson constructs a depth dependent scale space by mapping the measured depth of a pixel to a continuous scale parameter. Each pixel is then convolved with a Gaussian kernel of the appropriate scale. To speed up the calculation time the continuous scale space is approximated by convolving with a discrete set of Gaussian kernels and interpolating their results. Scale dependent edge detection can be performed by using a set of discrete derivatives of Gaussian kernels. Similarly we can detect other features such as texture using a discrete set of appropriate kernels at various scales. The orientation problem is more difficult. Even when the vehicles roll is known we still must use texture kernels at various orientations. To the best of our knowledge also reliable texture based terrain classification for autonomous vehicle is still an open issue.

2.6 Research direction

Night-time obstacle detection requires robust techniques for both stereo matching and obstacle detection. The goal of this research is to find, improve or develop suitable methods. In the text below we will give the research directions we have chosen based on our literature study.

In the computational stereo literature the best performing stereo algorithms are based on optimization of a Markov Random Field defined over the values in the DSI. These methods require complex optimization algorithms like graph-cut. The drawback of complex disparity optimization methods is that their results is heavily influenced by several optimization parameters. These parameters depend on the scene visible in both images, and reflect assumptions over the scene prior to stereo matching. Because the terrain in front of a mobile vehicle can display large variations, think of an expanse of grass or a rocky mountain pass, one set of parameters will not suffice. To our knowledge, the problem of obtaining the correct parameter values automatically from only image data has yet to be solved. Therefore there are two arguments against using these more complex optimization methods. First, they are computational intensive and difficult to implement for real-time frame-rates. Secondly, optimization parameters are crucial for good performance and are hard to estimate from the image data alone. Less complex methods like dynamic programming can be implemented in real-time, however there remains the problem of parameter choice (streaking). The just discussed drawbacks of Markov Random Field disparity optimization approaches put them of the list of possible improvements. A well known method to increase the robustness of stereo matching at a modest computational load is the use of image pyramids. Practically all disparity estimation methods that are based on image pyramids, or comparable approaches like scale-space, work in a coarse-to-fine manner. A well known disadvantage of coarse-to-fine approaches is that of error propagation. Errors made at the coarse scale propagate to the finer scales. We think it is more intuitive to work in a fine-to-coarse manner. Basically, our idea is to only use coarse information when needed. This idea is supported by research showing that fine-to-coarse depth disambiguation plays a role in the human visual system Hanspeter et al. [16], Smallman [56]. Retaining or replacing disparity estimates calls for the use of a measure that indicates the correctness of a disparity estimate. Based on this disparity validity measure we keep or reject estimates and replace them by estimates based on coarser information. Therefore, this approach stands or falls by the ability of the validity measure to distinguish between good and bad disparity estimates. The current validity measures do not provide reliable classification between good and bad stereo matches. Therefore finding a validity measure that can be used to distinguish bad from good matches is a key topic for our research.

For obstacle detection in unstructured terrain there are four possible solutions: column based analysis, V-disparity, 3D clustering or using techniques that work with map based representations. Map based representation e.g. elevation maps and voxel maps require tracking of the vehicle's position and pose and building a map representing the terrain around the vehicle. As mentioned earlier these methods are computational intensive and obstacle detection is only a part of their applicability. When using stereo vision and when only obstacle detection is required, we believe that other methods are more suitable. V-disparity methods, Labayrade et al. [30,] are promising, however their usage in unstructured terrain still poses some difficulties. Especially retrieving the parameters of a piecewise linear model of the terrain using Hough transforms [21] is not straightforward. We believe that investigating methods that allow reliable OD using V-disparity estimation is a research topic in itself and is beyond the scope of this study. Another promising technique is presented by Talukder et al. [58]. It overcomes the sliceing plane problem of column based approaches by using efficient 3D clustering. Unfortunately, despite their efficient implementation real-time frame-rates have not been achieved. Based on the observations mentioned above we will choose a column-based approach for measuring the slope of the terrain and thereby detecting obstacles. We think that significant improvements can still be made for column based approaches. For these improvements we will look at some key concepts of the method presented by Talukder et al. [58] and possibly modify them for column based slope analysis. The one fundamental problem that remains is that of the angle between the sliceing plane and the surface normal, see section 2.4.2. Therefore, one of our key research topics will be finding methods that overcome the sliceing plane problem for column based approaches in a computational efficient manner.

Chapter 3

Approach

In this chapter we describe the proposed obstacle detection system and the choices that let to its design. Section 3.1 gives a bird's eye view of the system. In section 3.2 our disparity estimation algorithm will be described. The obstacle detection methods are discussed in section 3.3. Finally, in section 3.4 we describe the test platform and the datasets together with the metrics we used for evaluation.

3.1 System overview

As can be seen in figure 3.1, the system is made up of three parts; rectification, disparity estimation, 3D reconstruction and obstacle detection. The stereo camera system provides a stereo image. This stereo image will be rectified first and then used to create a dense disparity map. The disparity map is used together with the rectified stereo camera's parameters to reconstruct the 3D coordinates for every point in the disparity map. The 3D coordinates are used to estimate terrain slope and find depth discontinuities. Then, positive and negative objects can be extracted by using thresholds on the found slope, depth, height and width of the candidate objects. Rectification and 3D reconstruction are discussed in sections 2.2.2 and 2.2.3. The disparity estimation and the obstacle detection techniques, which are both novel methods, will be described in the next sections.

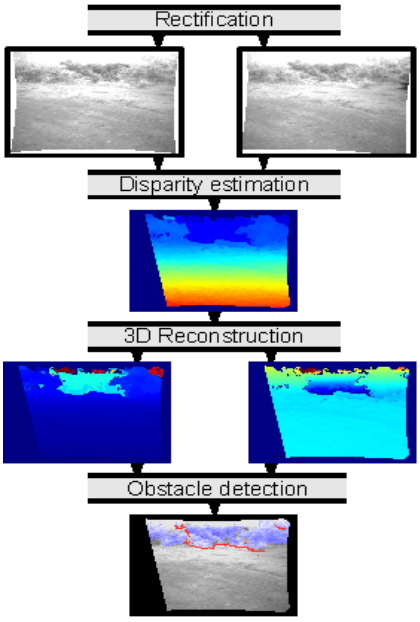


Figure 3.1: System overview.

3.2 Disparity estimation

The developed stereo algorithm uses a fine-to-coarse multi-resolution technique, see figure 3.2. Multi-resolutions approaches make disparity estimation more robust. Furthermore by using our novel fine-to-coarse scheme we do not suffer from error propagation usually associated with coarse-to-fine approaches. The input to the algorithm is a rectified stereo image pair from which a stereo image pyramid is created. Each level in the stereo image pyramid is Rank transformed to compensate for radiometric distortion. Next the algorithm creates a dense disparity map for each level in the stereo image pyramid independently from each other. Then it combines disparity estimates from each level in the disparity pyramid to form one dense disparity estimate. The combining of the different disparity estimates from the different resolutions is based on our confidence measure. This measure reflects the quality of each estimate in the disparity image pyramid. In the text below we briefly describe the disparity estimation process. In the following sections the important steps of the stereo algorithm are discussed more thoroughly. Section 3.2.1 deals with the used single- resolution stereo algorithm. Section 3.2.2 deals with the computation of the confidence measure. In section 3.2.3 we describe how the estimates from the different resolution are combined based on their confidence. Finally in section 3.2.4 we motivate our choices.

Down Sampling
We construct an image pyramid for both stereo images.

Pre-processing
To make the image more robust against noise and radiometric distortion we pre-process the images in the image pyramids using the Rank transform, section 2.3.5.

Single level disparity estimation
For each pyramid level, we estimate a dense disparity map. The single level stereo algorithm is described in section 3.2.1.

Edge strength
Edge strength maps are also computed for each pyramid level. These maps are obtained by convolving the images with derivatives of a Gaussian kernel.

Confidence
The confidence measure is computed using the local deviation in disparity estimates and relating it to the edge strength. The computation of the confidence measure is discussed in section 3.2.2

Up sampling
The resolution of each disparity and confidence pyramid image is restored to the original image size. This facilitates the subsequent confidence based disparity selection.

Confidence based level selection
We select the disparity estimate with the highest confidence in a level by level manner. Disparities from lower levels are favoured over those higher up in the pyramid, because they represent higher resolution estimates. The confidence based level selection is discussed in section 3.2.3

Post processing
Finally, we apply blob filtering on the disparity map followed by median filtering, both are discussed in section 2.3.15.

The core of our disparity estimation technique is the novel disparity confidence estimation technique. We believe that by using our confidence measure it is possible to reliable detect errors in the disparity map and replace them with estimates based on coarser information. This can improve the resulting disparity estimate. Currently, we create an image pyramid first and then compute disparity for all pixels and all levels in the stereo image pyramid before selection. While this allows for proof of concept of the novel confidence based fine-to-coarse approach it is not the most efficient implementation. However, we think it is possible to improve on efficiency using the following scheme. First only perform the disparity estimate at full resolution. Then determine which pixels need estimates from a coarser scale based on their confidence. Next only interpolate those pixels that are needed for the disparity estimation process at the coarser level. Subsequent perform disparity estimation only for those pixels and apply the process again. In this manner redundant computations are avoided. Furthermore, it allows the system to adaptively balance its computational load based on the quality of the images.

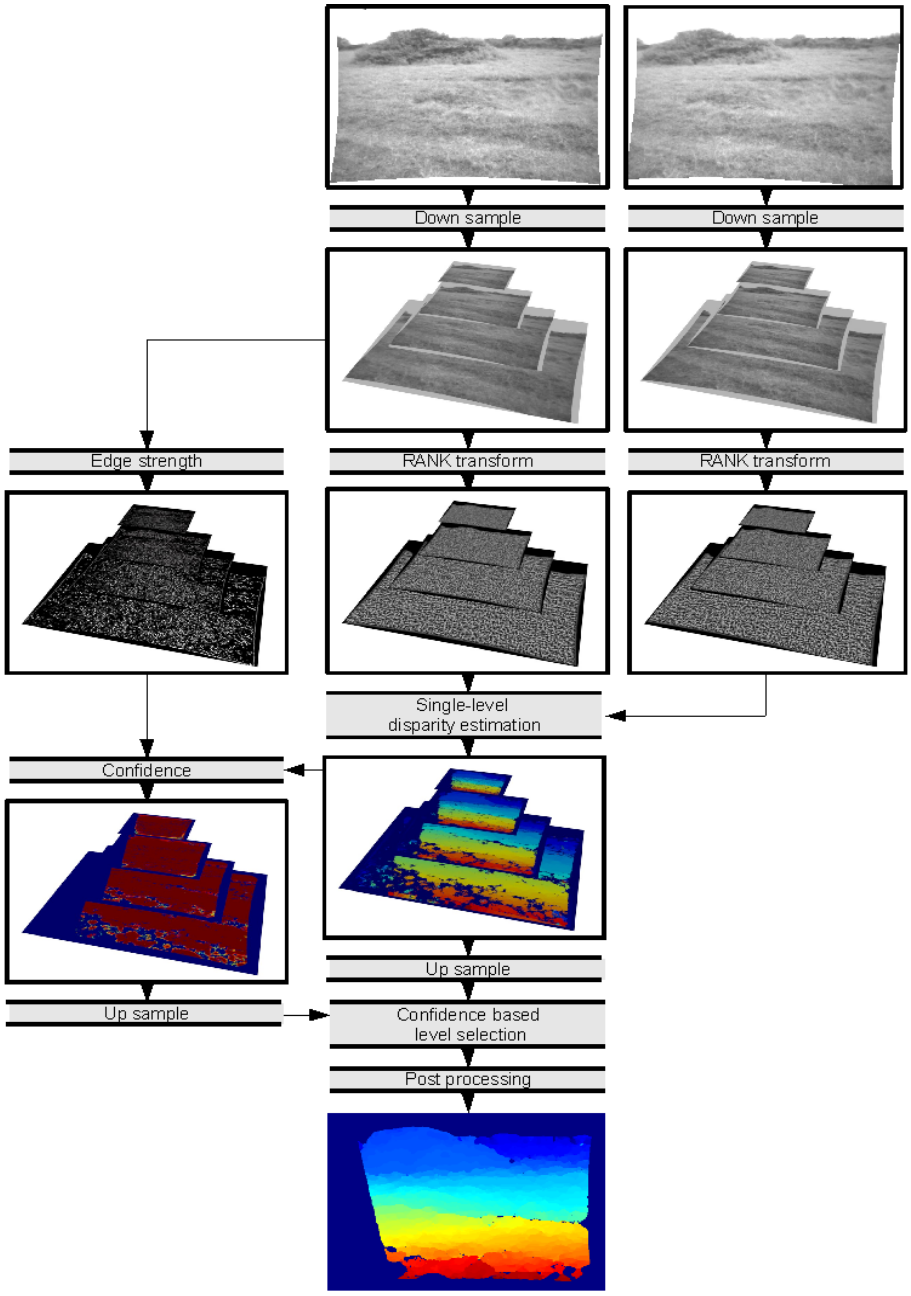


Figure 3.2: Steps in our disparity estimation algorithm.

3.2.1 Single resolution stereo algorithm

For each level in the image pyramid we use a real-time disparity estimation process, van der Mark and Gavrila [41]. We will briefly describe the algorithm using the taxonomy of Scharstein and Szeliski [53].

Cost measure

The used cost measure is the Absolute Difference (AD). The benefits are that it can be computed fast and efficient, both on intensity values and on Rank transformed intensity values. More information on cost measures and pre-processing can be found in section 2.3.4 and 2.3.5 respectively.

Cost aggregation

We used a five sub-window approach for aggregation cost values between pixels. We found that using a sub-window approach reduces object edge dilation in the disparity estimate. Furthermore it also is a fast adaptive window method suitable for real-time implementation. (Note that when we speak of cost window size we mean the total size of the five overlapping sub windows.). Cost aggregation was the topic of section 2.3.6.

Cost minimization

For finding the actual disparity estimate we create a DSI and use the winner takes all optimization scheme with left right consistency checking. Also sub-pixel accuracy is obtained by interpolating the cost values inside the DSI. More information on DSI and cost minimization can be found in section 2.3.10.

Post processing

The disparity estimates are then processed using blob filtering followed by median filtering. Both are discussed in section 2.3.15. Finally the disparities are rescaled to compensate for the reduced image size.

3.2.2 Quality of disparity estimate

The basis for the novel fine-to-coarse approach is our confidence measure that reflects the quality of the found disparity estimate. The matching values within the search range used to find the estimate are often also used to asses its quality. A classic example of this is Winner Margin, see section 2.3.15. This is the normalized difference between the best and second best match. After some initial experiments it was clear that for our purposes DSI based confidence measures like Winner Margin were not performing adequate. This is caused by the fact that these measures ignore the local support for the actual disparity estimate. In other words a disparity estimate surrounded by pixels with the same disparity estimate is more likely to be correct regardless of a low winner margin.

We use the assumption that disparity over object surfaces will be smooth. Large disparity jumps will only occur near object borders or near faulty disparity estimates. These assumptions imply that in a small image region the amount in difference between the disparity estimates is low. It is expected to be high near object borders or faulty estimates. The first step is to measure the disparity deviation in a rectangular shaped window. We experimented with three methods to represent disparity deviation. The first one was standard deviation. The second one, the number of different rounded disparity estimates. Finally we used the average absolute difference relative to the centre pixel of the window. We found the third one most intuitive because it relates the deviation according to the centre pixel. Furthermore it can be computed fast and it behaves linear with the amount of disparity noise. This makes setting threshold values more intuitive.

When the local disparity deviation is computed, the next step is to find pixels with a high deviation which belong to faulty estimates. We do this by subtracting a scaled edge map from the scaled deviation map. The idea behind this is that high deviation can be compensated by a strong edge. In other words, a strong edge likely relates to an object border and can cause high disparity deviation. Also, a high disparity deviation without the presence of a strong edge, and thus object edge, violates our smoothness assumption and has to be faulty. The assumption that strong edges only occur near object borders is not always correct. Strong edges can also occur inside an object. Therefore, such an edge on an object's surface seems to violate our assumption. However, this does not always result in ignoring faulty estimates, because the signal to noise ratio of the intensity edge is high. As a result the disparity estimation process most likely was able to find a good match. The used technique incorporates both the fact that disparity jumps can occur near regions with high edge strength and the fact that regions with low edge strength and thus low image variance are likely to produce bad matches.

The whole process can be summarized as follows. Let *Disp* be the disparity map computed from I_l and I_r which are the left and right images of one particular level in the stereo image pyramid. And let C_{height} and C_{width} be the height and width of the cost window used during the single scale disparity estimation. The first step is to calculate the edge strength in the intensity image. For this we filter I_l with the normalized derivative of a Gaussian kernel. This kernel has strong response near intensity edges. The derivative of the 1D Gaussian kernel is given by:

$$g(x) = -x e^{\frac{-x^2}{2}} \qquad (3.1)$$

Then E_x is the convolution ($M = 3$) of I_l with $h(x)$ in the horizontal image direction.

$$E_x(h,w) = \frac{1}{M} \sum_{m=-M}^{M} I_l(h, w+m) g(m) \qquad (3.2.a)$$

And E_y is the convolution of I_l with $h(x)$ in the vertical image direction.

$$E_y(h,w) = \frac{1}{M} \sum_{m=-M}^{M} I_l(h+m, w) g(m) \qquad (3.2.b)$$

We compute the edge strength by using formula 3.3 with $\Delta h = \lfloor C_{height} 0.5 \rfloor$ and $\Delta w = \lfloor C_{width} 0.5 \rfloor$. The resulting edge strength map is scaled to the unit interval [0..1] by dividing through its highest value.

$$Edgestrength(h,w) = \sum_{h-\Delta h \le m}^{m \le h+\Delta h} \sum_{w-\Delta w \le n}^{n \le w+\Delta w} \sqrt{E_x(m,n)^2 + E_y(m,n)^2} \qquad (3.3)$$

The next step is to compute the local deviation in disparity using formula 3.4 with $\Delta h = \lfloor C_{height} 0.5 \rfloor$ and $\Delta w = \lfloor C_{width} 0.5 \rfloor$. Also the deviation map is rescaled to the unit interval by dividing through its highest value.

$$Dev(h,w) = \frac{1}{(\Delta h \Delta w) - 1} \sum_{h-\Delta h \le m}^{m \le h+\Delta h} \sum_{w-\Delta w \le n}^{n \le w+\Delta w} |Disp(h,w) - Disp(m,n)| \qquad (3.4)$$

Finally, we compute the confidence for every valid pixel in the disparity image by using formula 3.5 (we used $\alpha = 1$ for our experiments).

$$confidence(h,w) = -Dev(h,w) + \alpha\, Edgestrength(h,w) \qquad (3.5)$$

For every pixel without a disparity estimate set the confidence to -1. Finally we can rescale the confidence map so that its values are in the range between 0 and 1.

3.2.3 Confidence based Multi-scale

We now describe the method that combines the disparity estimates from the different levels in the stereo image pyramid. In multi-scale approaches it is common to start at the top level of the stereo image pyramid (lowest resolution) and let the resulting disparity estimate guide the disparity estimation process one level down. The advantage of this approach is that it limits the search space on higher resolution levels. However, errors made at the top level are also propagated to the lowest level. In contrast to the often used coarse-to-fine approaches we use a fine-to-coarse selection scheme. It starts at the highest resolution level and from there works its way to the top of the pyramid. This process is described in more detail in the text below.

Let $Disp^n$ be the disparity map resulting from level n in the stereo image pyramid and $Conf^n$ is its accompanying confidence map. $n=0$ is the lowest level in the stereo image pyramid and $n=N$ is the highest. First all disparity maps and confidence maps are resized to the original resolution. Then the following selection scheme is applied where θ is an appropriate threshold. For a given image point <x,y> in $Disp^0$ check if its confidence $Conf^0(x, y)$ is higher than θ. When it is keep the disparity estimate $Disp^0(x, y)$. If not go one level higher in the image pyramid. Again check if the new disparity estimate $Disp^1(x, y)$ has a confidence $Conf^1(x, y)$ higher than θ. If it is use $Disp^1(x, y)$ as the final estimate. If not go one level higher in the image pyramid and apply the just described selection method again. The scheme is applied until an estimate is found with a confidence higher than θ or if we reached the top of the image pyramid. If no estimate with a confidence higher than θ was found mark it as invalid.

3.2.4 Motivation and Discussion

The field of computational stereo is large and still growing. The question arises if it was necessary to come up with a new disparity estimation process instead of using an existing method. As discussed earlier, literature on stereo algorithms which are applicable for low visibility conditions is limited. Usually, additional hardware is used to increase visibility instead of making the algorithm itself more robust. With our approach we tried to construct a multi-resolution stereo matching method that is robust against low visibility conditions and does not suffer from coarse-to-fine error propagation. The key of our method is our disparity validity measure. Our validity measure balances intensity edges, signal-to-noise ratio and the final disparity smoothness. All of which have already been used separately in other disparity estimation methods before. Bobick and Intille [4] use intensity edges as ground control points that must be visited by the dynamic programming algorithm that create an optimal path through a DSI slice. This way they try to avoid the streaking effect, usually associated with dynamic programming based methods. Signal-to-noise ratio is used by Kanada and Okutomi [24] [45] to estimate the appropriate matching window size. Marovec's interest operator detects image patches with low signal-to-noise ratio and discards their disparity estimates. Disparity smoothness is the basis for many optimization methods and post-processing steps like blob filtering and median filtering. The strength of our validity measure is that it weights several aspects to form one indication about the correctness of a disparity estimate.

3.3 Obstacle detection system

The developed obstacle detection system uses geometrical properties of the scene to distinguish between drivable and un-drivable terrain. Positive obstacle detection is based on the local terrain slope. For an image point $I(h,w)$ the terrain slope can be obtained by looking at the depth and height profile for points in the same image column. By grouping pixels with a terrain slope above a certain threshold we can identify regions that likely belong to positive obstacles. Measuring the height of the grouped points and using a depth dependent threshold further refines our positive obstacle search. Our positive OD approach differs from other approaches in the way we choose the image neighbourhood for terrain slope estimation. Furthermore, instead of using a single slope threshold we use multiple thresholds in a hysteresis process. Grouping of positive obstacle points is not based on image coordinates alone. Their estimated depth is taken into account as well. For negative obstacles, we look at the depth differences between pixels in the same image column. These differences are corrected for the inherent uncertainty in depth estimation. If the uncertainty corrected difference exceeds a threshold, we mark the respective pixel as a possible negative obstacle. Again, negative obstacles are grouped and the clusters are evaluated based on their width using a depth dependent threshold. One of the benefits of our approach is that we can use terrain measurements (*dZ dY*) for detecting both positive and negative obstacles. This reduces the computational load of the system. An overview of the developed OD system is given below. In the coming sections we describe the system in more detail.

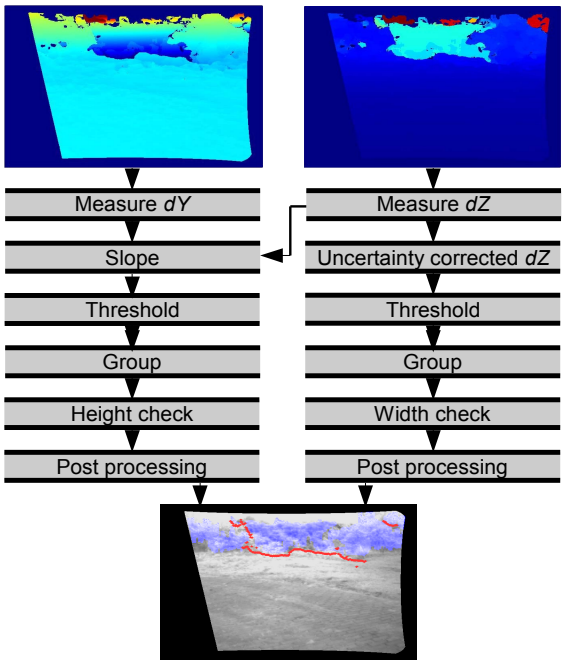


Figure 3.3: Obstacle detection system overview, positive obstacles left column, negative obstacles right column.

3.3.1 Column based slope estimation

As discussed earlier the choice is what pixel step size S to use for the slope measurement. A small pixel step size will measure the slope at a scale to fine for our purposes. A large pixel step size will miss terrain features that are of interest. Using a fixed pixel step size will inevitably cause the problems mentioned above. This forces us to use a pixel step size that is dependent of the depth of an image point. The intuition is that we want to keep the difference in height for the slope measurement constant throughout the scene. By using a larger S for pixels nearby, the slope estimates becomes more robust against small irrelevant terrain features. A smaller S for pixels further away causes the slope estimation not to miss relevant terrain obstacles. Instead of using a fixed pixel step size S we use a fixed height step size S_m given in meters. This insures that we estimate the slope at the same scale throughout the terrain. Using a height step size S_m the pixel step size S can be calculated with formula 3.6. Where f is the focal length of the used camera and Δy is the vertical pixel size, both in meters.

$$S = \left\lceil \frac{f S_m}{(f + P^1_{Vz})\Delta y} \right\rceil \qquad (3.6)$$

The next question is how to apply S. We can let S extend downwards or upwards. The figures below show that the right choice in using a positive or negative pixel step size depends on the local geometry of the obstacle around P^1_v.

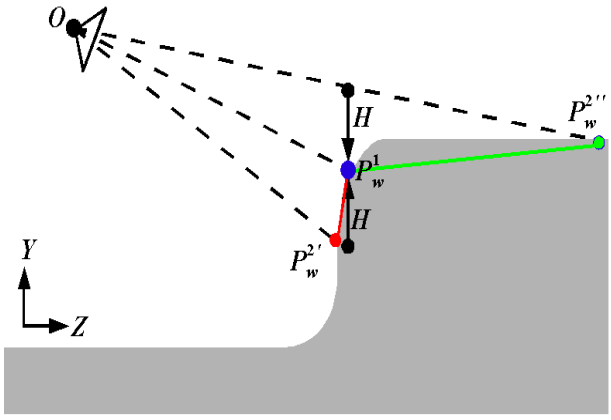


Figure 3.4: Downward estimated slope

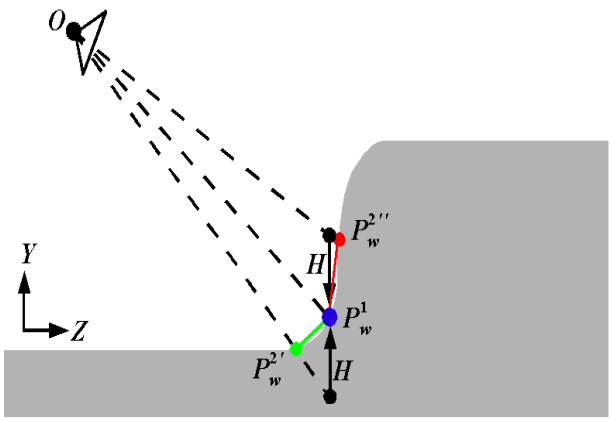


Figure 3.5: Upward estimated slope.

In figure 3.4 the downward slope, using $P_v^{2'}$, will supply the most intuitive value. In the case of figure 3.5 the upward slope, using $P_v^{2''}$, will give the most intuitive value. To automatically choose between the downward and upward slope we use the following scheme. For a pixel $I(h,w)$ calculate the pixel offset S according to its estimated depth using formula 3.6. Then estimate the terrain slope upward using $I(h-S,w)$ with formula 3.7 and downward using $I(h+S,w)$ with formula 3.8. In this way we obtain both a downward and upward estimated slope. Finally, we choose the slope estimate that has the maximum absolute value, formula 3.9.

$$T_u = \frac{P_{Vy}^{2''} - P_{Vy}^{1}}{P_{Vz}^{2''} - P_{Vz}^{1}} \qquad (3.7)$$

$$T_d = \frac{P_{Vy}^{1} - P_{Vy}^{2'}}{P_{Vz}^{1} - P_{Vz}^{2'}} \qquad (3.8)$$

$$Terrainslope = \begin{cases} T_u & if \ \ |T_u| \geqslant |T_d| \\ T_d & if \ \ |T_u| < |T_d| \end{cases} \qquad (3.9)$$

3.3.2 Uncertainty corrected gap estimation

Negative obstacles are detected by looking for depth jumps in the depth profile of an image column. Based on P^1_{vz}, S_m and the height of the camera above the ground Ch we can calculate the expected depth jump EdZ if we assume a flat ground plane (see figure 3.6 and formula 3.10). The expected depth difference EdZ can be compared against the actual found depth difference dZ.

$$EdZ(P_z)=\left(\frac{S_m}{Ch-S_m}\right)P_z \qquad (3.10)$$

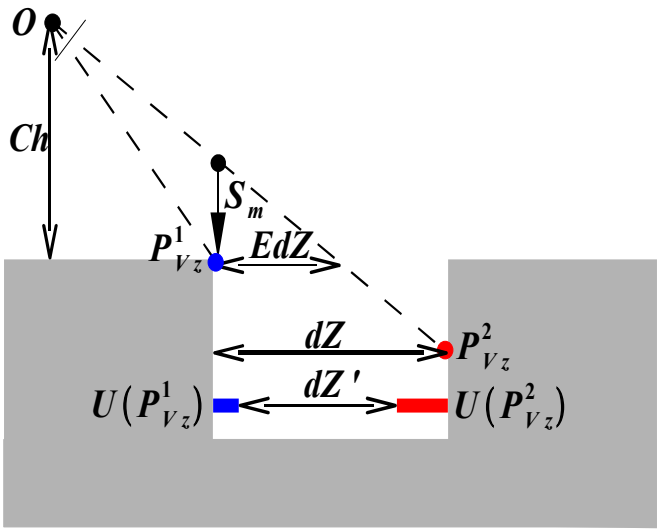


Figure 3.6: Gap estimation.

The role of uncertainty in the depth estimates should also be considered. For robust detection of negative obstacles we want to use the minimum bound on the found dZ. We can do this by using the formula 2.12 from section 2.2.4 which is reprinted below for convenience.

$$U(P_z)=\frac{\sqrt{2}\,\Delta x\,P_z^{\,2}}{f\,b}$$

Where Δx is the horizontal pixel size, f is the focal length of the used camera and b is the baseline width. Using this uncertainty we can compute a minimum bound for dZ using:

$$dZ' = (P^2_{Vz}-\frac{U(P^2_{Vz})}{2})-(P^1_{Vz}+\frac{U(P^1_{Vz})}{2}) \qquad (3.11)$$

For detecting negative obstacles, we can compare the uncertainty corrected depth jump dZ' against the expected depth jump EdZ. For instance, if $\delta \cdot EdZ \leqslant dZ'$ we can mark the pixel as belonging to a negative obstacle. Here δ is an appropriate threshold

3.3.3 Hysteresis based thresholding

Other obstacle detection systems often rely on constant threshold values. We argue that the use of a single threshold is not favourable in practice. Due to the fact that our depth estimation will contain noise, especially during low visibility conditions, the use of hysteresis thresholding is in our opinion a better solution. In a pixel labelling task hysteresis uses thresholds to distinguish between seed pixels and grow pixels. Seed pixels will always be labelled. A grow pixel will only be labelled if there is a *n*-connected path of grow pixels to a seed pixel.

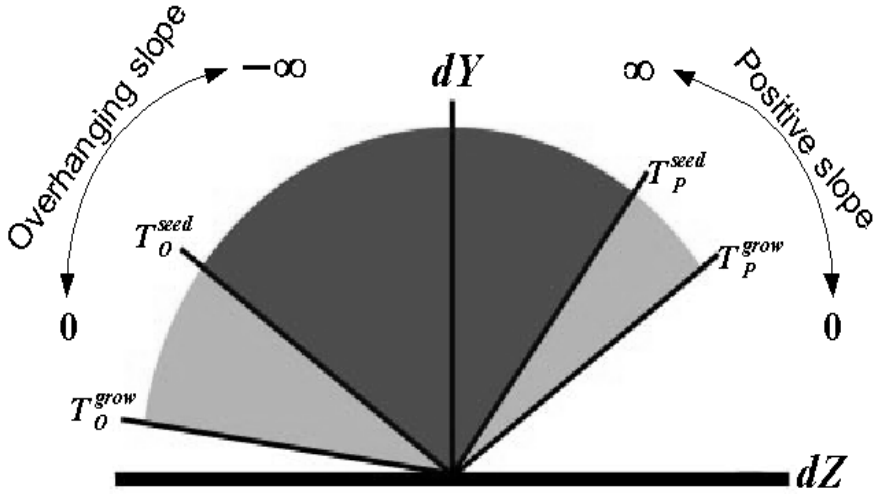


Figure 3.7: Hysteresis based thresholding

For positive obstacle detection, we use hysteresis thresholding in the following way. Every pixel with a positive dY and a slope within the range of $[T_O^{grow} \dots T_O^{seed}) \cup [T_P^{grow} \dots T_P^{seed})$ will be marked as a grow pixel (light grey). And all pixels within the range of $[T_O^{seed} \dots \infty) \cup [T_P^{seed} \dots \infty)$ will be marked as seed pixels (dark grey). For all seed pixels we perform simple morphological opening with a small square kernel (3x3), Gonzalez and Woods [14], to find consistent patches of seed pixels. These seed pixels will be marked as positive obstacles. The grow pixels will only be marked as a positive obstacle if there is a 4-connected path of grow pixels to a seed pixel. These connected grow pixels can be found efficiently using region filling, Gonzalez and Woods [14].

For negative obstacles we apply a similar scheme. Every pixel with a negative dY and a uncertainty corrected depth difference dZ' in the range of $[T_N^{grow} EdZ \dots T_N^{seed} EdZ)$ will be marked as a grow pixel. And all pixels with dZ' in the range of $[T_N^{seed} \cdot EdZ \dots \infty)$ will be marked as seed pixels. Note that for a given pixel the actual threshold depends on the expected depth jump as described in section 3.3.2. Again, the seed pixels are processed with morphological opening and we use region filling to find grow pixels that are connected to seed pixels.

3.3.4 Grouping and Obstacle refinement

We refine our initial (after hysteresis thresholding) obstacle map by measuring the height of positive obstacles and the width of negative obstacles and compare them against depth dependent thresholds. In order to measure these dimensions, we first have to group the initially found object pixels together.

For positive obstacles, grouping of pixels is done by taking both their image coordinates and depth values into account. An obstacle pixel can only be grouped to its 8-connected neighbours. Furthermore, it can only be grouped if the depth difference between the two pixels is smaller than two meters. To prevent obstacle cluttering over the ground plane we perform morphological opening, with a horizontal bar kernel (7x3), on the initial obstacle map. The result of the grouping process is a labelled obstacle map. For each obstacle in the obstacle map we measure its height and compare it to a depth dependent height threshold. To make the measuring of the obstacle height more robust, against outliers in the reconstructed pixel coordinates, we use its projected height in pixels O'_{height} instead of its estimated height O_{height} in meters. The depth dependent height threshold for positive obstacles has the form:

$$T_{height}(O_{depth}) = \alpha_p^{O_{depth}} - \beta_p \qquad (3.12)$$

where α_P controls the steepness of the exponential curve and β_P controls the minimum obstacle height. O_{depth} is the average depth of the obstacle. We can convert this height from meters to pixels using the formula below (Δy denotes the vertical pixel size).

$$T'_{height}(O_{depth}) = \left\lceil \frac{f\, T_{height}(O_{depth})}{\Delta y (f + O_{depth})} \right\rceil \qquad (3.13)$$

Finally a positive obstacle passes the height check when $T'_{height}(O_{depth}) \leq O'_{height}$. For $\beta = 0.6$ and varying α both functions are plotted below. We choose an exponential function because its steepness is easily controlled with one parameter i.e. α_P. Furthermore an exponential curve captures the fact that uncertainty in depth grows non-linearly with the distance as well as the fact that small objects at large distances are less significant for path planning.

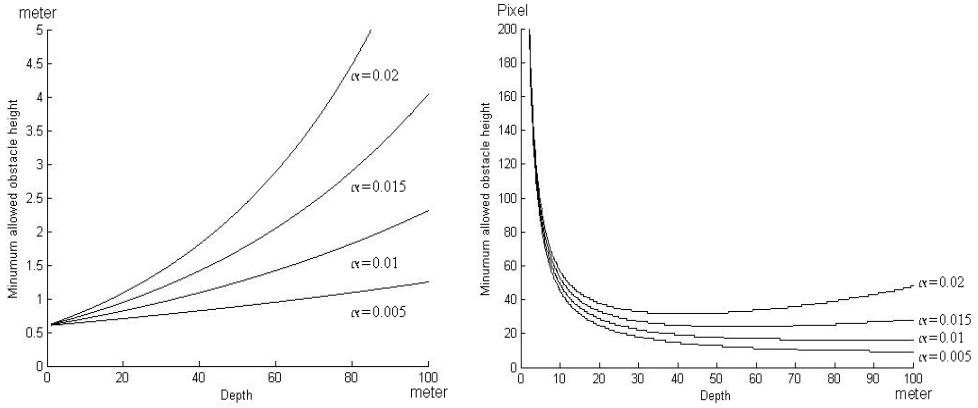


Figure 3.8: Depth dependent height threshold in meters (left) and pixels (right).

For negative obstacles we perform a similar scheme. First the initial negative obstacles are squeezed to one pixel thick lines. For every vertical image column in a negative obstacle we only keep that pixel that is closest to the negative obstacle's edge. We find this pixel by searching for the maximum depth jump between a pixel and is upper neighbour. The search range starts at the lowest obstacle pixel in the image column and ends at the highest obstacle pixel in the image column extended with the pixel step height (used during slope estimation). The next step is to segment the negative obstacle map in separate obstacles. In contrast with positive obstacles, grouping of pixels is only based on image coordinates and not on depth values. A width criteria is used to discard false negative detections. Again the minimum width is calculated with an exponential curve:

$$T_{width}(O_{depth}) = \alpha_n^{O_{depth}} - \beta_n \qquad (3.14)$$

and can be converted to pixels with:

$$T'_{width}(O_{depth}) = \lceil \frac{f\, T_{width}(O_{depth})}{\Delta x (f + O_{depth})} \rceil \qquad (3.15)$$

Finally, a negative obstacle passes when its width in pixels exceeds the depth dependent threshold i.e. $T'_{width}(O_{depth}) \leqslant O'_{width}$.

3.3.5 Post processing

The post processing methods applied mainly increase the visual quality of the obstacle map. Negative obstacle pixels are dilated with a small (9x9) diamond shaped kernel. This causes the one-pixel thick lines to become more visible in the image. As we shall see in section 3.4.5 this has little influence on their evaluation. For positive obstacles we look at their vertical image columns. Every pixel that has a pixel above and below it in the same image column from the same object, will also be marked as belonging to that positive obstacle. Note that positive obstacle segmentation is based on image and depth connectivity. For visualization purposes however every positive object is coloured blue. So obstacles that seem to be connected in the obstacle map might not belong to same obstacle at all.

3.3.6 Motivation and Discussion

Column based OD techniques are often used because of their low computational needs. As we described in section 2.4.1 there are some drawbacks when using these approaches. Instead of using more computational demanding methods like that of Talukder [58] (section 2.4.2), we tried to find efficient improvements over standard column based OD. In the coming text we describe our potential contributions to existing column based methods.

Firstly, we use a depth dependent step-height. From the publications by researchers at JPL, see section 2.4.1, we know that they were using variable step-heights for slope measurement. The size of their step-heights was based on the expected depth of the given image row while assuming a flat surface in front of the vehicle. If they adopted their approach to use step-heights dependent on the actual measured depth on a per pixel basis is unclear from their later publications. However we do know that in work from Talukder [58], the image projection of the truncated cones does depend on the estimated depth on a per pixel basis. Secondly, we use the maximum of the forward and backward estimated slope. We did not find this approach in other publications. However, again a similarity can be drawn between the Talukder approach which uses triangles below and above the point under investigation. This image point is labelled if one of the pixels, in the cones above or below, passes certain thresholds. Thirdly, objects are rejected based on depth dependent height (for positive obstacles) or width (for negative obstacles) thresholds. We use this approach because small obstacles at large depth are less significant for path planning and most likely are false detections. While one can argue this is not the task of the OD system, we believe that it can reduce the false detection rate prior to path planning. Finally, column based slope analysis is sensitive to the angle between the surface normal and the plane defined by focal point and the image column, see section 2.4.1. We use hysteresis thresholding to tackle this problem. Hysteresis thresholding has become a text-book method, Davies [11], to increase classification accuracy. However we did not found it in the literature concerning column based OD.

3.4 Evaluation, system, methods and metrics

In this section, we describe the manner in which we evaluated the developed OD system. First in section 3.4.1 we look into the hardware used during our tests. Section 3.4.2 gives insight in the recorded datasets and the way they are processed to form ground truth. Then in section 3.4.3 we describe some challenges when evaluating obstacle detection systems. And finally in section 3.4.4 and 3.4.5 our methods and metrics used for evaluation the obstacle detection methods will be presented.

3.4.1 Test set-up

The goal of our research is to investigate the suitability of night-time stereo based OD. TNO's RoboJeep see figure 1.1 and 3.9 provides a research platform for autonomous vehicle research. To illuminate the scene in front of the vehicle during the night we use near infra-red Light Emitting Diodes (LED's). Near infra-red is preferred over traditional light bulbs because of reduced signature. The used cameras with the specification below are used to construct a stereo set-up. For keeping track of the vehicle movement and position we used an Inertial Measurement Unit (IMU) and Global Positioning System (GPS). The LIDAR and SONAR also visible on figure 3.9 were not used for this research. Using the described set-up we have a low-cost and low signature solution for night-time conditions. The specification of the used camera and IR lamp are given below.

1) 36 Watt LED near infra-red (880 nm) emitter (Profiline TV6899).
2) Two digital camera's using CMOS imaging chips with a resolution of 640x480 running at 30 fps with automatic gain control between 62 dB and 110 (Aglaia INKA NSC LM9618) .

Figure 3.9: Test set-up.

3.4.2 Dataset and labelling

Using the system described in section 3.4.1 we recorded over 40 GB of day and night-time images. From these recordings we manually selected 140 daytime images (the day-time dataset) and 140 night-time images (the night-time dataset). Time synchronization was used to associate GPS data to each recorded camera image. By matching the GPS data and by visual inspection, we selected for every frame in the daytime dataset a comparable frame in the night-time dataset.

Figure 3.10: Test terrain, 1 big gap 2 concrete wall, 3 Rock pile, 4 Holes, 5 Sand plane, 6 Trunks.

All 140 images from both our daytime and night-time dataset have been manually labelled. The labelling is pixel based and consist out of four classes. The *positive* obstacle class is coloured blue, the *negative* obstacle class given is coloured red, the *drivable* class is coloured green and the *ignore* class is coloured black. The labelling of positive obstacles is straightforward. We simply labelled each pixel of all objects in the scene that we considerer a positive object. Note that we do not only label that part of the object that has a slope above a certain threshold. Instead we label the whole object starting from the local ground plane to the top. For negative objects we choose to not only label the exact edge but also a region above and below it. The reason for this is that it is not realistic to demand the OD algorithm to find the precise location of negative object edges with pixel precision, especial during the night at greater distances. The real thickness of the labelled edge will depend on the depth of the negative object in the picture.

The greater the depth the thicker the labelled edge. Much effort has been put in to make sure that the thickness of the labelled edges is comparable between day and night-time images. The ignore class is used for objects or group of objects at large distances. By labelling a pixel with the ignore class it is excluded from evaluation. Of course this class is used with great restraint. Everything left in the image is labelled as drivable. In the next section we briefly present our dataset.

1 Big gap sequence

This sequence contains 40 images both during the day and night taken while the vehicle was driving to a approximately 6 meter wide and 4 meter deep gap. The rectified images are shown below together with their ground truth label maps.

Figure 3.11.a: Day-time frame 20.

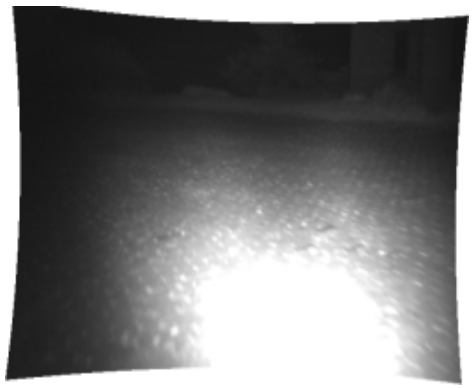


Figure 3.11.b: Night-time frame 20.

Figure 3.11.c: Day-time label 20.

Figure 3.11.d: Night-time label 20.

2 Concrete wall sequence

This set contains 20 day and night images taken while the vehicle was driving over an expanse of tall grass past a concrete wall.

Figure 3.12.a Day-time frame 4.

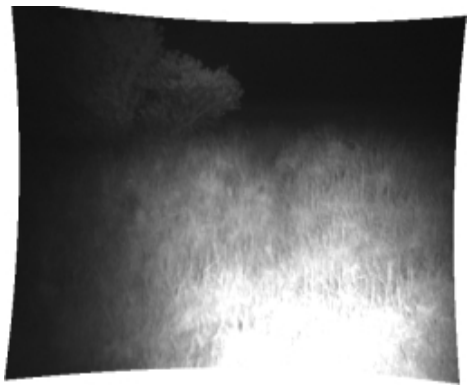


Figure 3.12.b: Night-time frame 4.

2 Rock pile sequence

This set contains 40 day and 40 night images taken while the vehicle was driving towards a pile of rocks.

Figure 3.13.a: Day-time frame 38.

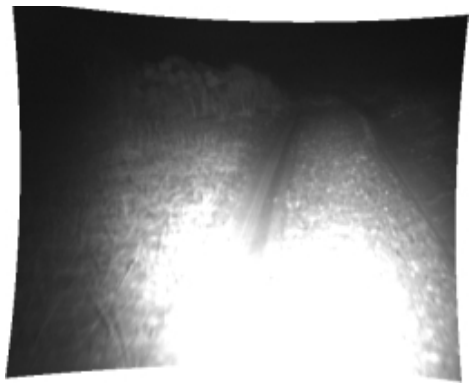


Figure 3.13.b: Night-time frame 38.

4 Holes sequence

This set contains 20 day and 20 night images taken while the vehicle was driving through an expanse of grass covered with traversable holes.

Figure 3.14.a: Day-time frame 18.

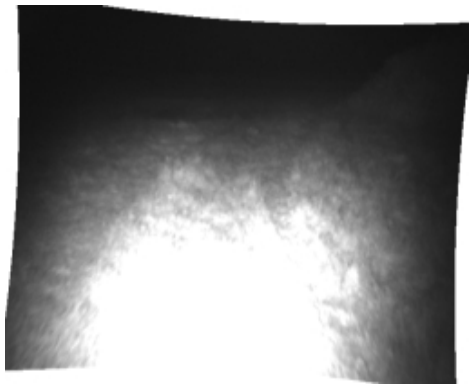


figure 3.14.b: Night-time frame 18.

5 Sand plane sequence

This set contains 10 day and 10 night images taken while the vehicle was driving over a sand plane towards a sand dune.

Figure 3.15.a: Day-time frame 1.

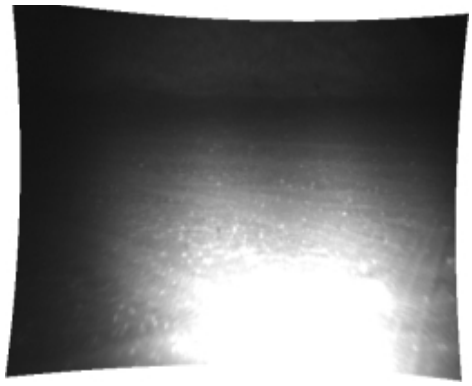


Figure 3.15.b: Night-time frame 1.

6 Trunks sequence

This set contains 10 day and 10 night images taken while the vehicle was driving towards a pile of trunks.

Figure 3.16.a: Day-time frame 1.

Figure 3.16.b: Night-time frame 1.

3.4.3 Evaluation of obstacle detection

We need a metric that can be used for absolute and relative comparison between OD systems. With absolute we mean it should be able to give insight in how well the system would perform perform under realistic conditions. For our application, this is the ability to navigate the vehicle safely through rough terrain. With relative we mean the ability to fairly compare different systems with each other. A reliable and efficient way to evaluate the system would be to couple it to a path planning simulator. The simulator renders photo-realistic stereo imagery used by the system to compute the obstacle maps. The path planner can then find a path around the found obstacles. The found path and the optimal path can then be compared, for instance based on travelling distance. Using such a simulator is within the reach of modern technology. However, it would require substantial effort to construct such a system. Especially realistic modelling of camera influences, rough terrain and vehicle dynamics will require considerable attention. An other approach is to use real-world images. In this manner the data itself is as realistic as possible; however, evaluation becomes more challenging. A straightforward scheme would be to use evaluation based on a pixel based classification problem. The figures below illustrate that pixel-based evaluation has some great disadvantages.

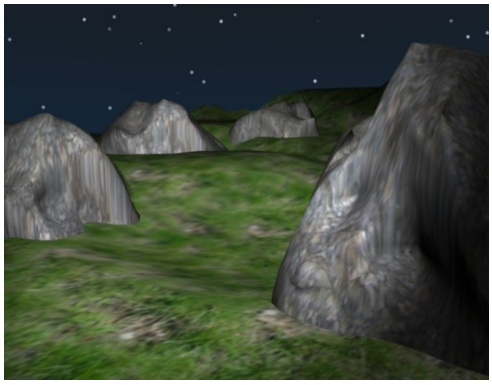


Figure 3.17: Scene.

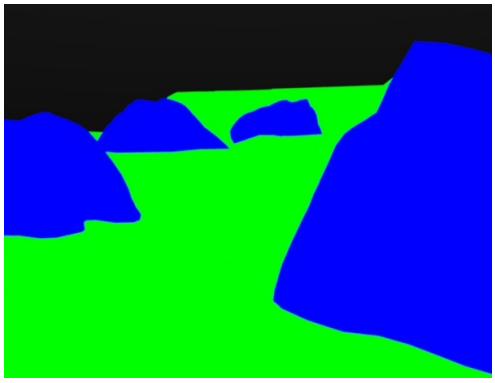


Figure 3.18: Ground truth.

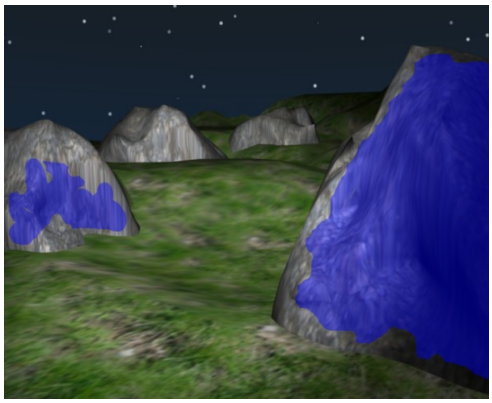


Figure 3.19: Obstacle map system A

System A: true positive rate = 0.68

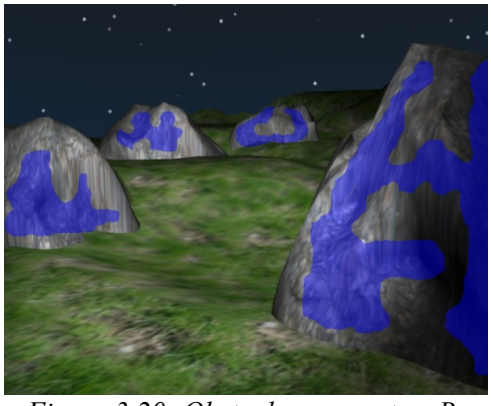


Figure 3.20: Obstacle map system B.

System B: true positive rate = 0.49

The question is which system performed best. System *A* has a true positive rate of 0.68. System *B* has a true positive rate of 0.49. Based on these measures we could conclude that system *A* out performs system *B*. However, system *B* might be more appropriate because it can detect obstacles further away from the vehicle. This simple example illustrates why we think it is more appropriate to use the surface that pixels represent instead of the pixels themselves. The problem with this approach is that the surface which a pixel represent is based on its depth. And while the ground truth class of a pixel is available its ground truth depth is not. This forces us to use the estimated depth together with the ground truth labelling to compute the ground truth surface of a pixel. The details of our evaluation method are described in the next sections.

3.4.4 Positive obstacle evaluation

We propose a positive obstacle evaluation method not based on the pixels themselves but on the surface that those pixels represent. In this section, we describe the quantities used to evaluate our system. We make a distinction between vertical pixels (positive obstacles) and horizontal pixels (drivable terrain). Figure 3.21 and 3.22 show the (gross) simplifications of camera geometry and terrain geometry we used for calculating the surfaces of vertical and horizontal pixels. We note that the purpose of the coming formulas is solely to provide quantities for evaluating our OD system in an efficient manner.

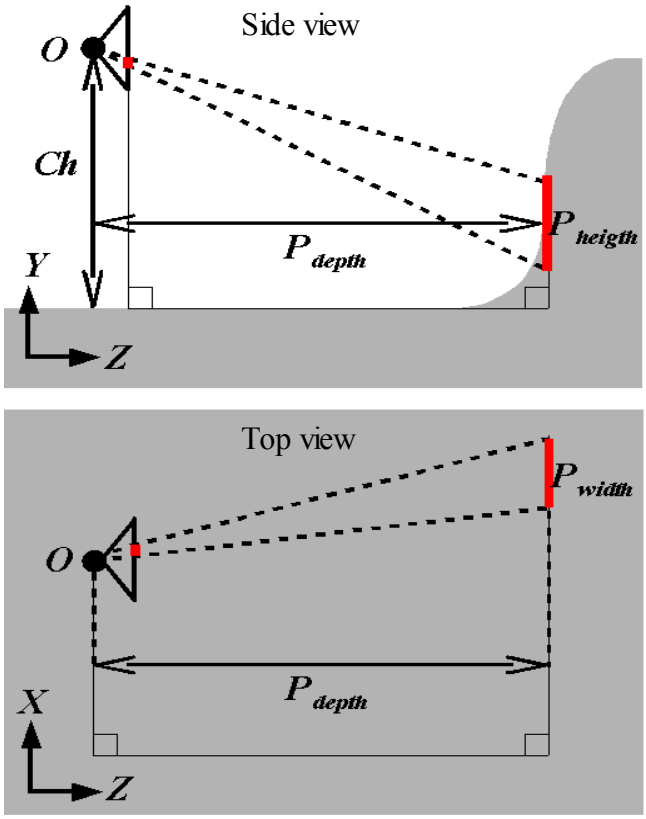


Figure 3.21: Vertical surface

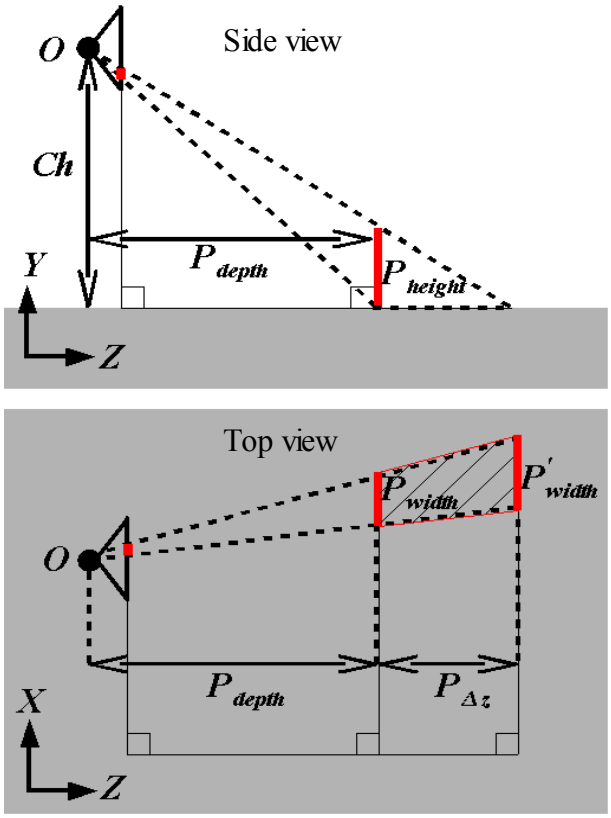


Figure 3.22: Horizontal surface.

We compute the surface of a vertical pixel using formula 3.16, 3.17 and 3.18. The pixel's estimated depth P_{Vz} is used to compute its real-world height P_{height} and width P_{width}.

$$S_{vertical}(P) = P_{height} P_{width} \tag{3.16}$$

$$P_{height} = P_{Vz} \frac{\Delta y}{f} \tag{3.17}$$

$$P_{width} = P_{Vz} \frac{\Delta x}{f} \tag{3.18}$$

Here Δx is the horizontal pixel size and Δy is the vertical pixel size. The horizontal surface $S_{horizontal}$ is computed with:

$$S_{horizontal}(P) = P_{width} P_{\Delta z} + \frac{(P'_{width} - P_{width}) P_{\Delta z}}{2} \tag{3.19}$$

where P'_{width} and $P_{\Delta z}$ can be computed with:

$$P'_{width} = (P_{Vz} + P_{\Delta z}) \frac{\Delta x}{f} \tag{3.20}$$

$$P_{\Delta z} = P_{Vz} \cdot \frac{P_{height}}{Ch - P_{height}} \tag{3.21}$$

Let D be a set of pixels then their total horizontal surface can be computed with:

$$Horizonatal(D)=\sum_{P\in D} S_{horizonal}(P) \qquad (3.22)$$

And the their total vertical surface with:

$$Vertical(D)=\sum_{P\in D} S_{vertical}(P) \qquad (3.23)$$

We now describe how we use these formulas to compute our performance measures. Let I^n be the nth image in the dataset, G^n is its ground truth obstacle map, D^n is the computed depth map and O^n is its computed obstacle map. These maps are used to fetch the ground truth label of a pixel, $P_{ground\ truth}$, its obstacle classification label, P_{label} and depth P_{Vz}. The value for a pixel's label can be *Positive*, *Negative*, *Drivable* or *Ignore*. Now let C^n be the set of pixels with a estimated depth larger than *MinD* and smaller than *MaxD* in the nth image.

$$C^n=\{P \mid MinD<P_{Vz}<MaxD, P\in I^n\}$$

Changing the value for *MinD* and *MaxD* allows us to measure performance at different distances from the vehicle. $D^n_{ground\ truth\ drivable}$ is the set of pixels that were labelled as drivable in the ground truth image,

$$D^n_{ground\ truth\ drivable}=\{P \mid P_{ground\ truth}=Drivable \wedge P\in C^n\}$$

and $D^n_{ground\ truth\ positive}$ is the set of pixels that were labelled as positive obstacle in the ground truth image.

$$D^n_{ground\ truth\ positive}=\{P \mid P_{ground\ truth}=Positive \wedge P\in C^n\}$$

$D^n_{true\ positive}$ are the pixels in image I^n that were correctly classified as positive obstacle pixels,

$$D^n_{true\ positive}=\{P \mid P_{ground\ truth}=Positive \wedge P_{label}=Positive\ ,\ P\in C^n\}$$

and $D^n_{false\ positive}$ are the pixels that were wrongly classified as positive obstacle pixels.

$$D^n_{false\ positive}=\{P \mid P_{ground\ truth}=Drivable \wedge P_{label}=Positive\ ,\ P\in C^n\}$$

These sets can be used to find the true positive and false positive rates over a set of N images as follows:

$$True\ Positive\ Rate=\frac{\sum_{1<n<N} Vertical(D^n_{true\ positive})}{\sum_{1<n<N} Vertical(D^n_{grount\ truth\ positive})} \qquad (3.24)$$

$$False\ Positive\ Rate=\frac{\sum_{1<n<N} Horizontal(D^n_{false\ positive})}{\sum_{1<n<N} Horizontal(D^n_{grount\ truth\ drivable})} \qquad (3.25)$$

The ground truth surfaces are computed according to a fixed set of day-time depth maps together with the ground truth obstacle maps. The depth maps are computed using our disparity estimation method discussed in section 3.2. We choose these depth maps because they have the highest percentage of valid depth estimates.

3.4.5 Negative obstacle evaluation

To use a surface based evaluation approach to get inside into the real-world applicability of an OD method requires a large evaluation dataset. Otherwise depth errors made can influence the ground truth surfaces significantly. Unfortunately our dataset only contains one sequence with a single negative object. This is why we considered using another method to measure the true negative rate of our system. The true negative detection rate is based on the percentage of an obstacle's length that is correctly classified. Here, we measure the obstacle length in pixels. For a given image I^n in our big-gap sequence we measure the length of the negative obstacle in pixels L^n based on the ground truth map G^n. Then we find all pixels in the computed obstacle map O^n that were correctly classified as negative obstacle i.e.

$$D^n_{true\ negative} = \{ P \mid P_{ground\ truth} = Negative \wedge P_{label} = Negative\ ,\ P \in I^n \}$$

From this set we only take one pixel per image column:

$$L^n_{true\ negative} = \{ P \mid Unique(P, D^n_{true\ negative}), P \in D^n_{true\ negative} \}$$

$$Unique(P, G) = (\forall P' : P' \in G \wedge P'_{width} = P_{width} : P'_{heigth} < P_{heigth})$$

Then the true negative rate over a set of N images is defined as:

$$True\ Negative\ Rate = \frac{1}{N} \sum_{1<n<N} \frac{L^n_{true\ negative}}{L^n} \qquad (3.26)$$

Note that this is a pretty straightforward method for evaluating negative OD performance. We use it because we have only one sequence with a single negative obstacle. Furthermore we roughly know the ground truth distance to the negative obstacle in every frame. This allows us to get insight in the negative OD performance at various distances. This in contrast to positive obstacles were the ground truth distances are not always known. For the false negative rate we use a similar approach as for the false positive rate. However before we compute the horizontal surface we dilate the false detections with a rectangular kernel. This is done because false negative detections are small lines only a few pixels in length. By dilating these pixels, the measure reflects the negative effect on the vehicles expected performance more realistic. Thus let O'^n be the obstacle map that is formed by dilating the negative obstacle pixels in O^n with a rectangular kernel. Dilation only has effect on pixels labelled as drivable in the ground truth map G^n. Then P'_{label} is the label associated with pixel P in the dilated obstacle map O'^n. The false negative rate over the whole dataset is then given by:

$$False\ Negative\ Rate = \frac{\sum_{1<n<N} Horizontal(D^n_{false\ negative})}{\sum_{1<n<N} Horizontal(D^n_{grount\ truth\ drivable})} \qquad (3.27)$$

where $D^n_{false\ negative}$ are the dilated pixels that were wrongly classified as negative obstacle pixels i.e.

$$D^n_{false\ negative} = \{ P \mid P_{ground\ truth} = Drivable \wedge P'_{label} = Negative\ ,\ p \in C^n \}$$

Chapter 4

Results

In this chapter we present the results of our research. Our disparity estimation and obstacle detection techniques have been compared against existing methods. We evaluated for both day and night-time conditions using a wide range of parameter settings. In the first section of this chapter we focus on depth estimation. We will look into the depth coverage, depth uncertainty, depth dilation and the performance of our novel disparity validity measure. The second section deals with obstacle detection. We investigate the true detection rate and compare it against the false detection rate for positive and negative obstacles during day and night conditions. Furthermore, we evaluated obstacle detection using a maximum obstacle distance of 5, 10, 25, 35 and 50 meter from the vehicle.

4.1 Depth estimation

In this section we describe the depth estimation results using the novel multi-scale technique and compare it to a single-scale method. The single scale method uses a 5-window approach, see section 2.3.7. Each of the five windows has a size of 5x5 pixels and the five windows together form a window of 11x11 pixels. We use the sum of the three best matching windows to compute the similarity value. Furthermore, we use left-right consistency checking with a maximum allowed difference of ten pixels. Optimisation is done using the winner takes all (WTA) approach with sub-pixel accuracy. A sub-pixel bound of 1 pixel is used. The disparity map is filtered using a blobfilter with 0.5 disparity threshold and minimal blob size of 5 pixels. The novel multi-scale approach uses a single scale approach, with the just described parameters, for each of its stereo pyramid levels. The size reduction between the pyramid levels is 65% using bi-linear interpolation. The base level of the stereo pyramid uses a left-to-right check tolerance of 10 pixels. For each successive level this tolerance is reduced with 2 pixels. After the disparity estimation is completed, the disparity map is used to reconstruct the 3D coordinates of all points with a valid disparity estimate. The variable parameters are the type of preprocessing step and the base resolution. We experimented with LoG and RANK preprocessing and measured their effect for the day-time and night-time datasets. For the single scale approach we tested the performance using the full (640x480) and half (320x240) resolution.

4.1.1 Depth coverage

We first present the depth coverage of our tested methods. The depth coverage is the percentage of pixels which are not rejected from the final disparity map. Rejection is based on the left-to-right consistency check and disparity blob filtering. For our multi-resolution approach disparity estimates are also rejected based on their disparity confidence. We evaluated over our 140 day-time and 140 night-time images. In the table below the different parameter configurations for our disparity estimation methods are given. Figure 4.1 presents their depth coverage results.

Test NR.	Test name	Dataset	Base resolution	Preprocessing	Pyramid levels
1	**Day Multi RANK FS**	**Day**	640x480	Rank transform	4
2	**Day Multi LoG FS**	**Day**	640x480	LoG convolution	4
3	**Day Single RANK FS**	**Day**	640x480	Rank transform	none
4	**Day Single LoG FS**	**Day**	640x480	LoG convolution	none
5	**Day Single RANK HS**	**Day**	320x240	Rank transform	none
6	**Day Single LoG HS**	**Day**	320x240	LoG convolution	none
7	**IR Multi RANK FS**	**IR**	640x480	Rank transform	4
8	**IR Multi LoG FS**	**IR**	640x480	LoG convolution	4
9	**IR Single RANK FS**	**IR**	640x480	Rank transform	none
10	**IR Single LoG FS**	**IR**	640x480	LoG convolution	none
11	**IR Single RANK HS**	**IR**	320x240	Rank transform	none
12	**IR Single LoG HS**	**IR**	320x240	LoG convolution	none

Table 4.1: Disparity test settings.

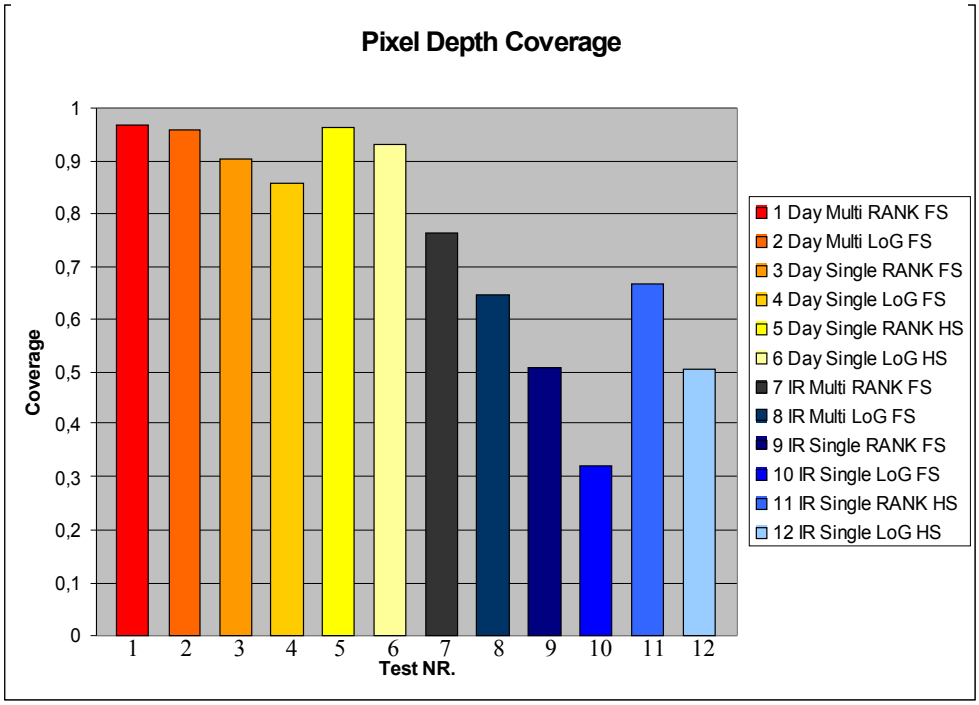


Figure 4.1: Pixel depth coverage.

The first thing to note is the positive influence of applying the Rank transform compared to LoG filtering as a preprocessing step. Especially during night time conditions the Rank transform shows an increase of at least 10 percent compared to LoG filtering. While LoG filtering is frequently used in approaches from the literature, this test suggest Rank transform might be more appropriate. Another interesting aspect is the performance gain during night conditions due to the novel multi-scale method. Again we see a 10 percent increase in depth coverage between the multi-scale and single-scale approaches. By using the single-scale approach at full resolution we do not reach an acceptable coverage level during night-time conditions. Most likely this is due to the fact that the five 5x5 matching windows do not contain enough distinctive intensity variation. If the single-scale approach is applied to half resolution images then only by using Rank preprocessing we achieve an acceptable coverage level. Below we show night-time disparity maps created with the various approaches.

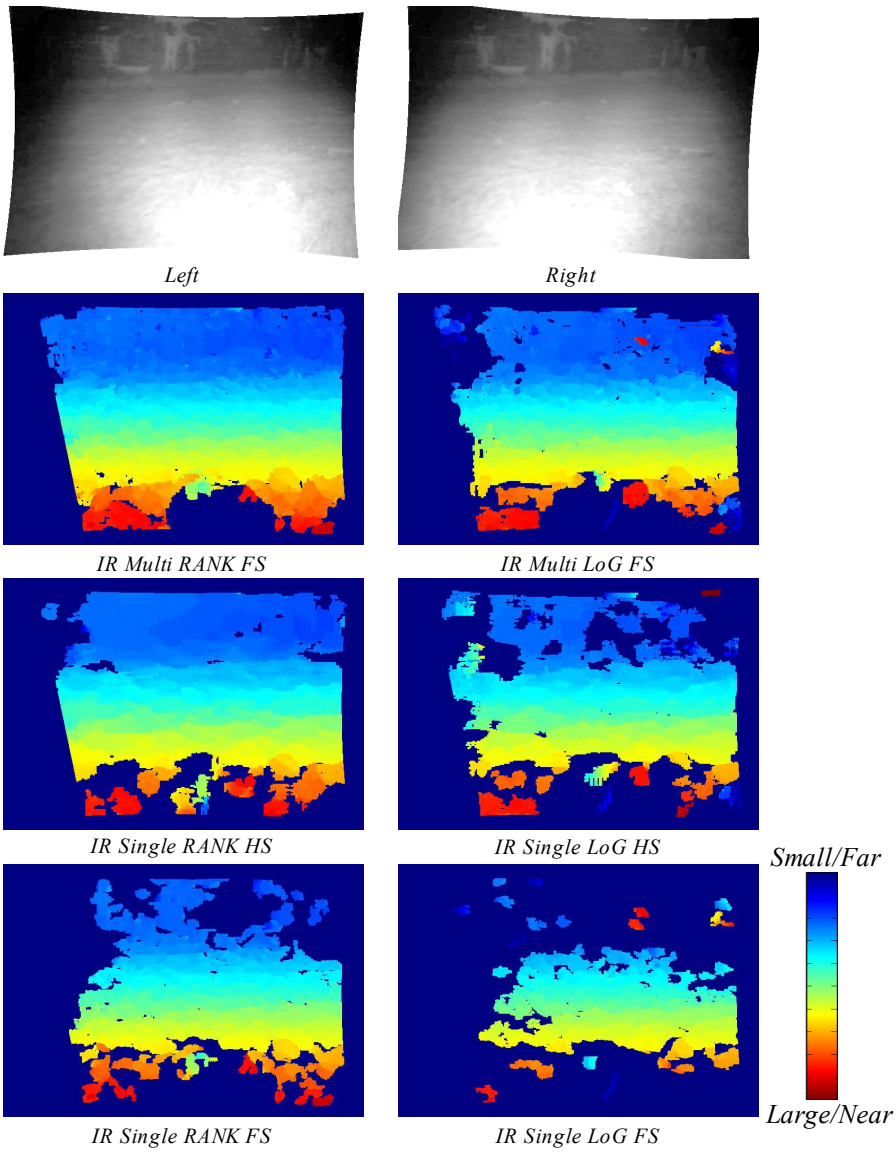


figure 4.2: Night-time disparity maps for a concrete wall.

4.1.2 Depth Uncertainty

Another aspect of our multi-scale approach is the possible decrease in depth uncertainty due to the use of higher resolutions. As discussed in section 2.2.4 depth uncertainty is a function of baseline width, focal length and the disparity error size (i.e. pixel size). Using half of the initial resolution effectively doubles the pixel size and thus increases depth uncertainty. In figure 4.3 we present the influence of the multi-scale approach on depth uncertainty. The benefits of the multi-scale approach is that it will take estimates from different resolutions. Some estimates will originate from the base of the image pyramid (highest resolution) having minimum uncertainty. Other estimates will originate from lower resolutions having more uncertainty. The question arises how much each pyramid level contributes to the disparity map. For several configurations we plotted the percentage of pixels coming from the four different levels in the stereo image pyramid. Again we evaluated over 140 day-time and 140 night-time images

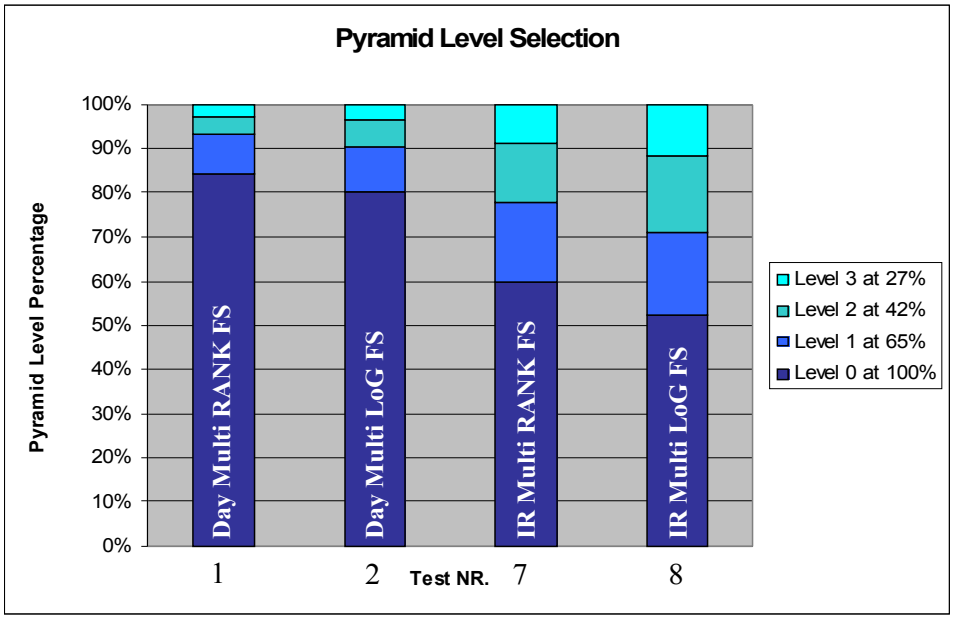


Figure 4.3: Pyramid level selection.

During day-time conditions the majority of the estimates come from the base of the stereo image pyramid (maximal resolution) and consequently have minimal depth uncertainty. During night-time conditions the base of the stereo image pyramid contributes 60% for Rank preprocessing and 52% for LoG preprocessing. This is consistent with the coverage levels of the multi-scale and single-scale configurations at full resolution, see figure 4.1. The benefits of our fine-to-coarse method is that it allows us to increase the depth coverage of the single-scale full resolution approach to an acceptable level (50% to 86%). Also single-scale half resolution approaches reach acceptable depth coverage levels. However with the multi-scale approach 60% of its estimates come from the base of the image pyramid. Consequently, the overall depth uncertainty is much less when using our multi-scale approach then when using a single-scale approach at half resolution.

4.1.3 Depth dilation

The figures below illustrate the depth dilation effects on day and night-time images. Depth is mapped from dark blue (near) to light blue (far). No colour indicates that no reliable depth estimate was found. The scene consist of a large ditch with some bushes in front of it. The far side of the ditch can be seen in the middle of the image.

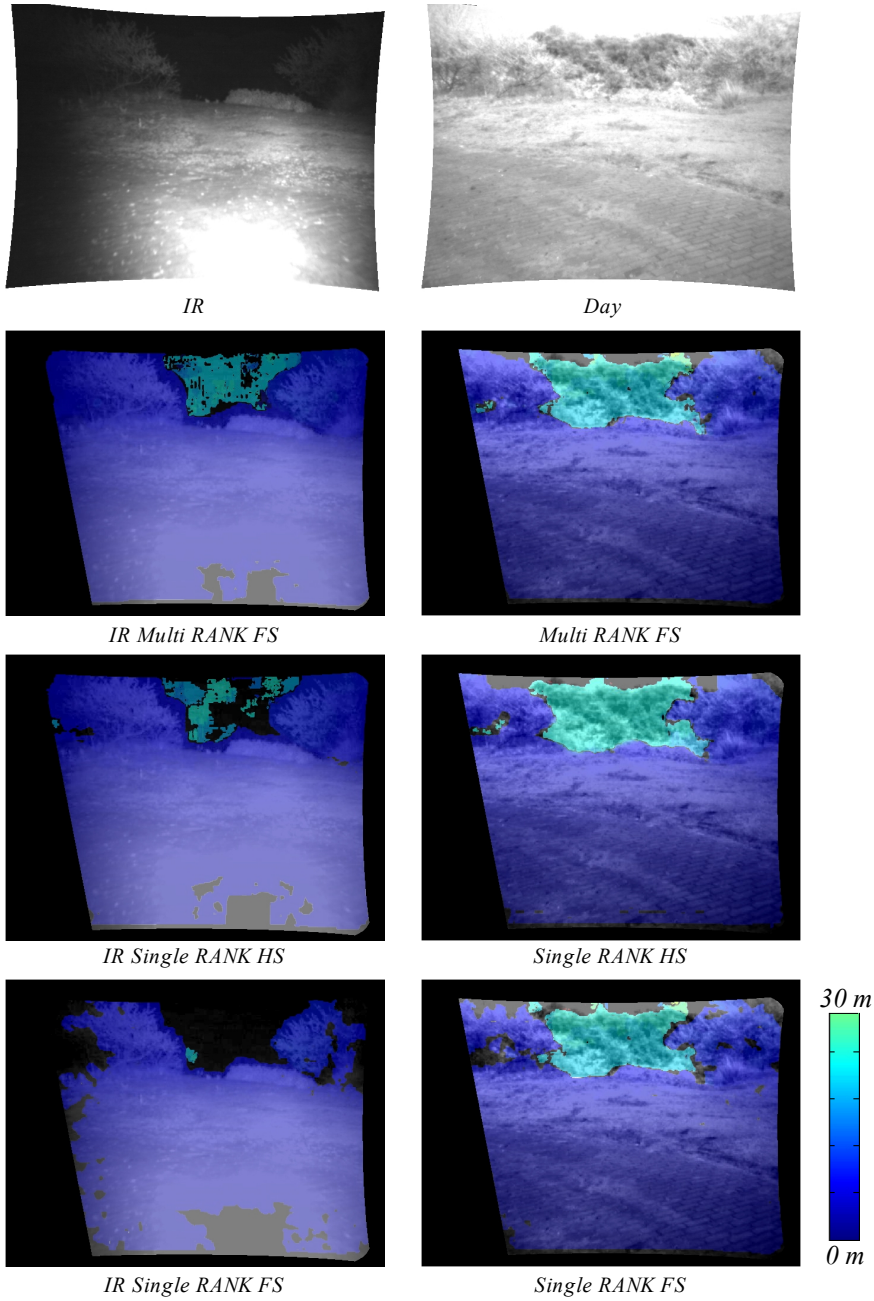


Figure 4.4: Depth dilation effects.

We can see depth dilation errors near objects edges are more severe for night-time images. Regardless of disparity estimation methods used, the dilation is significant. The effect is most likely caused by using directional lighting in a combination with window based matching. Objects near to the vehicle receive more light and thus have more intensity variation. Objects further away receive less light and therefore will have less intensity variation. As discussed in section 2.3.6. block matching based disparity estimation is biased towards the depth of objects that have the most distinctive intensity variation. During day-time conditions the effect of the light source on intensity variations between objects far and near to the vehicle can be neglected. We observed that during day-time conditions the depth of the far side of the ditch is sometimes favoured over the depth of the near side of the ditch. During night-time conditions the bushes in front of the ditch clearly have more intensity variation than the far side of the ditch. This causes the disparity estimation algorithm to favour the disparity of the bushes. From our observations is seems that objects appear larger during the night than they do during the day. The effect of the novel multi-scale method is also clearly visible in the images. While it can boost the depth coverage to an acceptable level it will also increase the depth dilation errors. This is a fundamental problem when using lower resolutions for block matching.

4.1.2 Disparity validity measures

Our fine-to-coarse disparity estimation method is based on the ability to distinguish between good and bad matches. In this section we compare traditional validity measures, see section 2.3.15, against our novel validity measure that is based on local disparity deviation and edge strength in the images, see section 3.2.2. We investigate their suitability as input for a threshold process. For this we performed disparity estimation on the Cone benchmark dataset [53] for which the ground truth disparity is exactly known. To make the process more challenging we added Gaussian white noise with zero mean and a variance of $0.75 \cdot 10^{-3}$ (image intensities are between 0 and 1), see figure 4.6. We used the same methods as discussed in section 3.2.1. However no post processing was applied, except for the left-to-right consistency check. ROC curves were plotted for several threshold values in the range of 0 up to 1. The ROC curves plot the true negative rate (bad pixels discarded correctly) against the false negative rate (good pixels discarded wrongly). For a given test we apply a threshold on the disparity validity measure and thereby divide the estimates in good and bad matches. The estimates that are labelled as bad are compared against the ground truth. If the difference between the disparity estimate and the ground truth disparity is equal or larger than 20% the estimate is regarded as faulty and its removal is assumed to be a correct choice. If the difference was smaller than 20% the estimate was correct and thus a good estimate was discarded wrongly. In this manner we can measure the true negative and false negative rates for different threshold values, see figure 4.5.

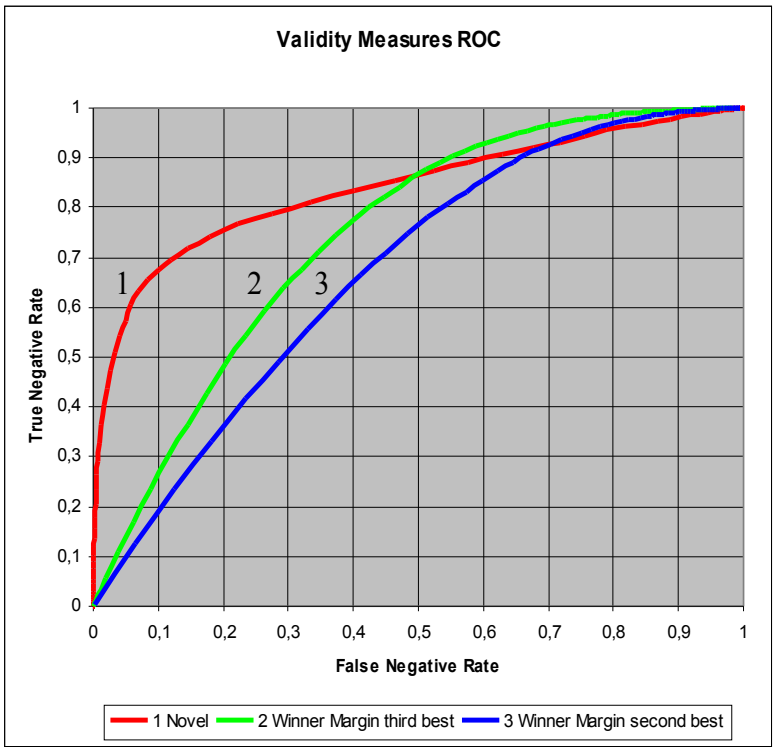


Figure 4.5:Validity based thresholds.

While we tested only on one image and used a disparity tolerance of 20%, the performance difference is unmistakable. Clearly, the novel validity measure is a better indication for the correctness of the disparity estimate. In figure 4.6 up to 4.9 , we see the effect of applying a threshold that discards 70% of the bad matches for the different validity measures. The validity maps are plotted for the range 0 bad (blue) to 1 good (red). The difference between the different validity measures is clearly visible. As can be seen, the cost based approaches have more difficulty distinguishing between bad and good matches. Using these measures in our fine-to-coarse disparity estimation approach would not achieve acceptable performance. With our novel validity measure we can make a more reliable distinction between bad and good matches. This ability is the corner stone of our fine-to-coarse disparity estimation method. It should be noted that cost based validity measures are widely used. However, our results show that their performance is not optimal.

Figure 4.6: Original with noise (left), Unfiltered disparity estimate (right).

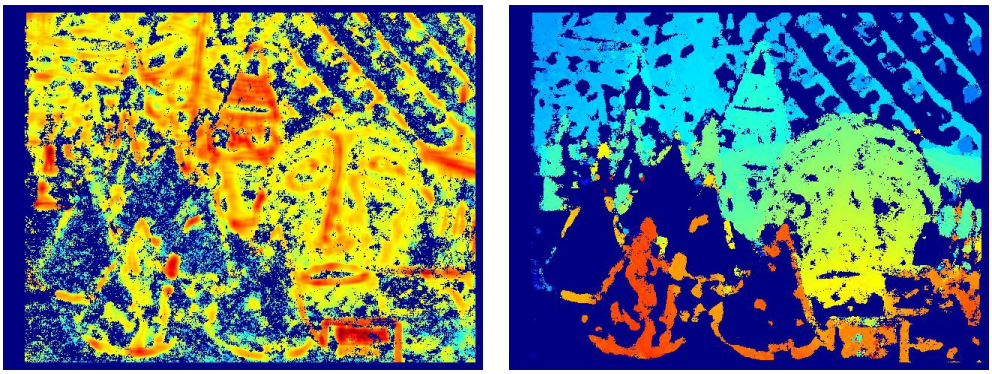


Figure 4.7: Novel validity measure (right), Filtered disparity map at 70% error reduction (right).

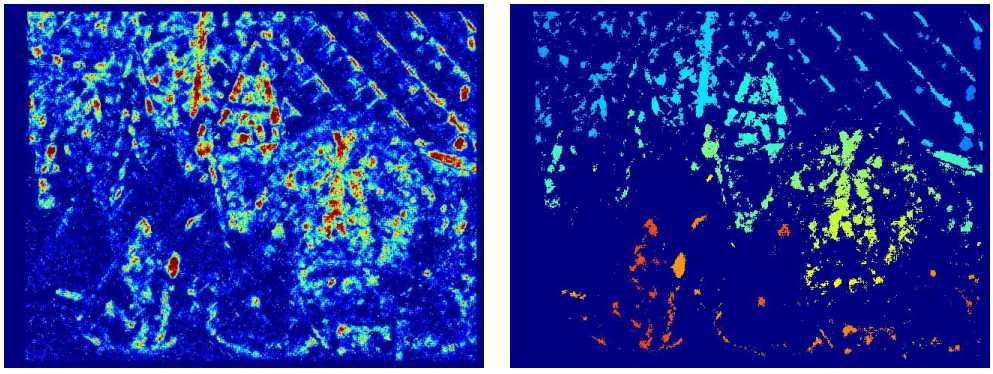


Figure 4.8: Winner Margin third best (left), Filtered disparity map at 70% error reduction (right).

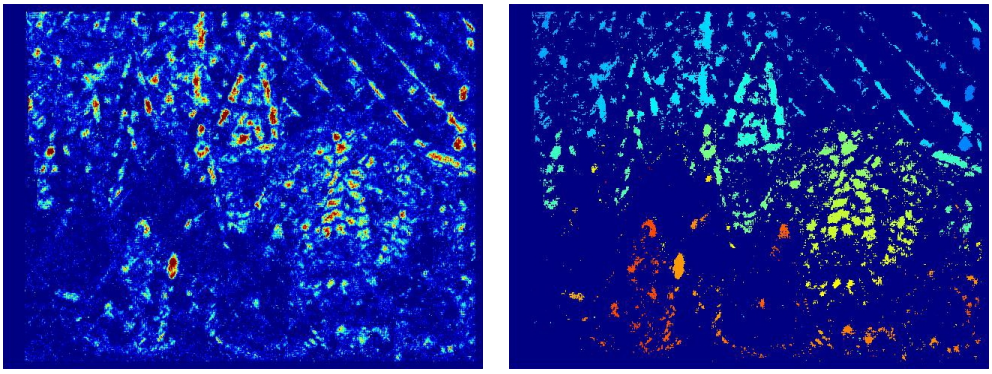


Figure 4.9: Winner Margin second best (left), Filtered disparity map at 70% error reduction (right).

4.2 Obstacle detection

In this section we examine the influence of different disparity estimation methods and OD parameters on OD performance. We evaluated our obstacle detection method as described in section 3.4. Day- and night-time results for positive and negative obstacles will be compared against each other. And while the results are mainly quantitative we also try to show the real-world applicability of the different approaches.

4.2.1 Parameter summary

In this section we summarize the OD parameters that are used for evaluating the system. In total we have experimented with 40 different OD configurations. We first show the parameters that were fixed during all test runs.

$\alpha_P = 0.10$ Controls the steepness of the exponential height threshold curve for positive obstacles.

$\beta_P = 0.6$ Controls minimum obstacle height ($1-\beta_P$) for positive obstacles.

$\alpha_N = 0.15$ Controls the steepness of the exponential width threshold curve for negative obstacles.

$\beta_N = 0.6$ Controls minimum obstacle width ($1-\beta_N$) for negative obstacles.

For more information regarding this parameters we refer to section 3.3.4.

For the step height, see section 3.3.1, we experimented with four different values: $S^{meters} = [0{,}15 \;\; 0{,}30 \;\; 0{,}45 \;\; 0{,}65]$. For each step height we also investigated the influence of the hysteresis thresholding using 10 different configurations for positive and negative obstacles, see table 4.2. For positive obstacles we start with a regular threshold (seed and grow threshold are the same) and then for each test we lower the grow threshold. For negative obstacles we do the same but also experiment with increasing the seed threshold. The hysteresis parameter configurations are given below.

Conf.	T_O^{seed}	T_O^{grow}	T_P^{seed}	T_P^{grow}	T_N^{grow}	T_N^{seed}
1	-0.10	-0.10	0.75	0.75	1.25	1.5
2	-0.10	-0.10	0.75	0.65	1.5	1.5
3	-0.10	-0.10	0.75	0.45	1.5	2
4	-0.10	-0.10	0.75	0.30	2	2
5	-0.10	-0.10	0.75	0.15	2	3
6	-0.10	-0.10	0.75	0.10	2.5	3
7	-0.10	-0.10	0.75	0.05	3	3
8	-0.05	-0.05	0.75	0.10	2	4
9	-0.05	-0.05	0.75	0.05	3	4
10	-0.05	-0.10	0.85	0.5	4	4

Table 4.2: Hysteresis test settings.

To efficiently plot the performance of each configuration we used the following testing scheme. For all tests we start with a step height of 0.15 and run experiments with all 10 hysteresis configurations in table 4.11. Next we use a step height of 0.30 and again run experiments with all 10 hysteresis configurations. And so on until we used a step height of 0.65 with hysteresis configuration 10. In total we have 40 different OD parameter configurations. For each setting we plot a marker in a ROC curve. The first parameter setting is plotted with a black maker. All consecutive parameters are plotted using markers with decreasing grey levels. The obtained plots do not allow for precise comparison between several configurations. However, they do show the general influence of different step heights and hysteresis thresholds. As an example we plotted the ROC curve in figure 4.10.

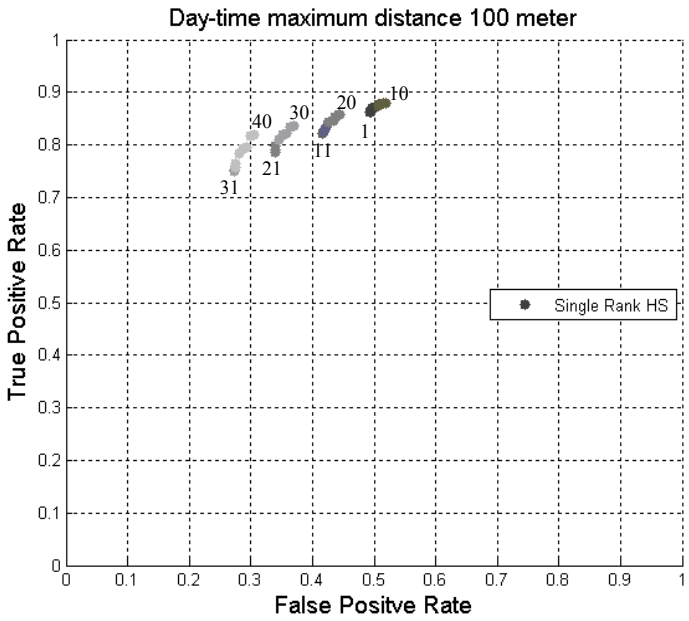


Figure 4.10: Example ROC curve.
Marker 1 up to 10: Step height = 0.15m
Marker 11 up to 20: Step height = 0.30m
Marker 21 up to 30: Step height = 0.45m
Marker 31 up to 40: Step height = 0.60m
For the ten markers in each set the hysteresis threshold values can be found in table 4.2.

We emphasize that any hard conclusions on the OD performance can not be made from this image due to the high depth uncertainty at 100 m. We only show this figure to illustrate the effect of the OD parameters and how the coming plots can be interpreted. What we effectively see is that by using a greater step-size we reduce the false detection rate of positive obstacles. However, the true detection rate will be lowered as well. By using hysteresis thresholding we can increase the true detection rate at a modest increase of the false detection rate (see configuration 31 up to 40). This makes hysteresis a powerful tool to boost the performance when using larger step heights. The effect of hysteresis thresholding and different step-heights will be made more clear in the coming sections.

4.2.2 Positive OD day-time

In this set of tests we used our day-time data. The figures below show ROC curves for different disparity estimation and positive OD configurations. We evaluated for terrain up to 50 meter away from the vehicle. Keeping in mind that estimated depth is used to realize this 50 meter range. Nevertheless, the results are encouraging. The ROC curves primarily allow comparison between methods. Conclusions about how the vehicle would perform using one such method is harder. First, we might not have to reach a true detection rate of 100%. If some small patches are not detected does not mean the path planner will be influenced by it. Similarly a true detection rate of 90% does not mean that 10 out of hundred objects are missed. It does indicate that on average 90% of an obstacle's surface is correctly classified. Secondly, zero false detection rates might not be necessary. In our experience false positive detections pop up for a period of one or two frames and do not reappear at the same place consistently. Based on our experience, we assume that save and efficient operation of a vehicle requires a true positive rate of at least 0.85 and a false positive rate less than 0.1 for our performance measures. Thus we demand that on average 85% of an obstacle's surface is labelled correctly. and that at least 90% of the drivable surface is available for path planning.

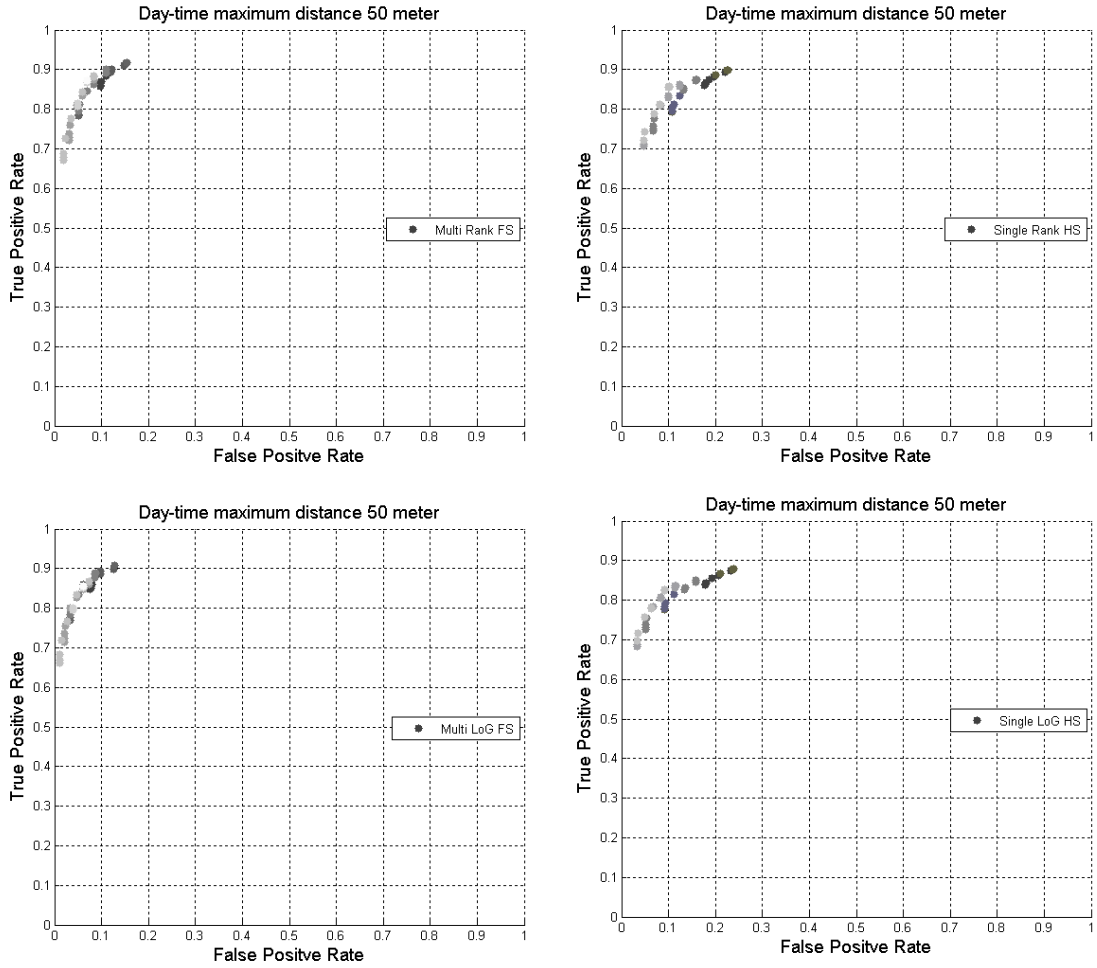


Figure 4.11: Positive OD day-time.

The influence of the different disparity estimation configurations on daytime OD performance is minimal. The multi-scale method has a slight better false detection rate than the single-scale method. Most likely this is due to the higher depth accuracy near obstacle boundaries achieved by using the full-resolution images. In the curves from figure 4.11 we again see the same effect of step height and hysteresis thresholding on positive OD as in figure 4.10. Effectively, a higher step height reduces false detections but also lowers the true detection rate. The loss in true detection rate can be compensated by appropriate hysteresis thresholding. The possible effect of using different step-heights and hysteresis settings are discussed in the next sections.

4.2.3 Effect of Step-height

In the plots from figure 4.11 we already saw that the smaller step heights (darker dots) have significant more false detections than the larger step heights (whiter dots). In figure 4.12 we illustrate the effect of using different step-heights for positive OD detection. Pixels classified as a positive obstacle are coloured blue. Clearly, small step heights produce many false alarms on the grass polls in the terrain. Increasing the step height increases the coarseness at which we look at the terrain and thereby reduces false detections. Eventually the step-height becomes too large and (part of) obstacles will be missed.

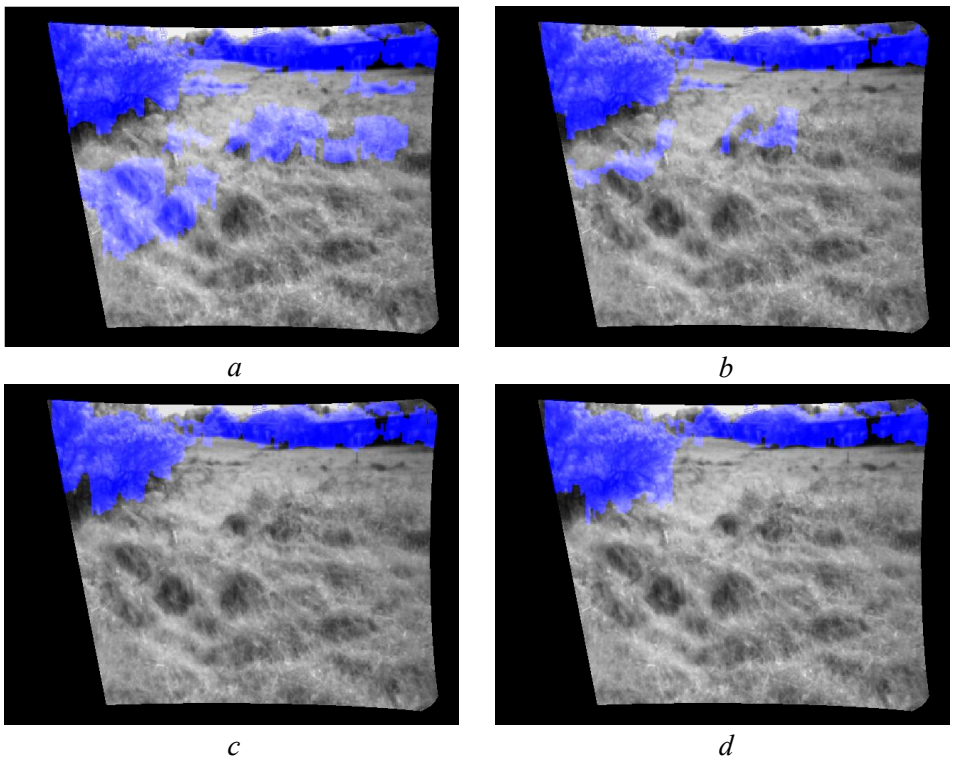
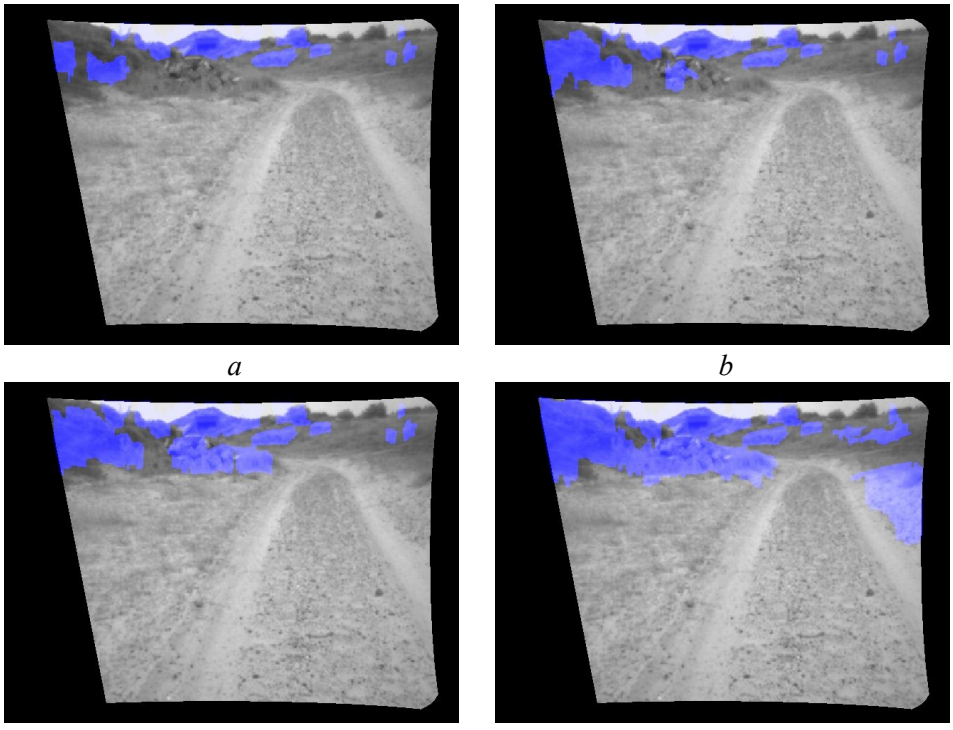


Figure 4.12: Effect of different step heights. 0.15 m(a), 0.30 m(b), 0.45 m(c), 0.60 m(d).

4.2.4 Hysteresis slope thresholding

The graphs in figure 4.11 also show that using a smaller step height can reduce the true detection rate significantly. Small obstacles are the cause of this because their edges are not detected. A powerful tool to boost the performance when larger step heights are used, is hysteresis thresholding. The figures below illustrate the effect of hysteresis slope thresholding. Initially, the object patches that exceed the regular slope threshold are too small to pass the object size thresholds (fig. 4.13.a). Therefore the object (pile of rocks left of the road) is not be detected. When we use hysteresis thresholding and start lowering the grow threshold these object patches will grow in size and pass the object size threshold, (fig. 4.13.b, 4.13.c). If we lower the grow threshold too much we will get false detections(fig. 4.13.d).



a *b*

c *d*

Figure 4.13: Effect of hysteresis thresholding, 0.75-0.75(a) 0.75-0.25(b) 0.75-0.15(c) 0.75-0.05(d).

If we look at the ROC curves in figure 4.11 we see that hysteresis can boost the true positive detection rate from 70% up to 90% while increasing the false positive detection rate by 10%. Improving true detection rates while keeping false detections at an acceptable level is crucial for vehicle performance. Later on, we will see that this is even more important during night-time conditions.

One of the fundamental problems of column based OD is the issue of the angle between the plane, defined by the camera's focal point and an image column, and the surface normal of an obstacle. When this angle is large the detected slope can be significant less than the true slope, see section 2.4.1. Talukder [58] tries to overcome this problem by using 3D point clustering in triangular image regions. In figure 4.14 we show that hysteresis thresholding can also be a powerful tool to overcome this problem. First, we show the original image 4.16.a with its estimated depth map 4.16.b and slope map 4.16.c. Slope is plotted in the range blue(flat) to red(steep). The slope map is computed with the method discussed in section 3.3.1 using a step-height of 0.45 m Due to the viewing angle effect, the slope estimated for the brushwood left of the road is less than its true slope. By using a regular threshold the brushwood will not be consistently classified as a positive obstacle, 4.16.d. However by applying hysteresis thresholding the obstacle map will be become more consistent over the obstacles in the scene, 4.16.e and 4.16.f.

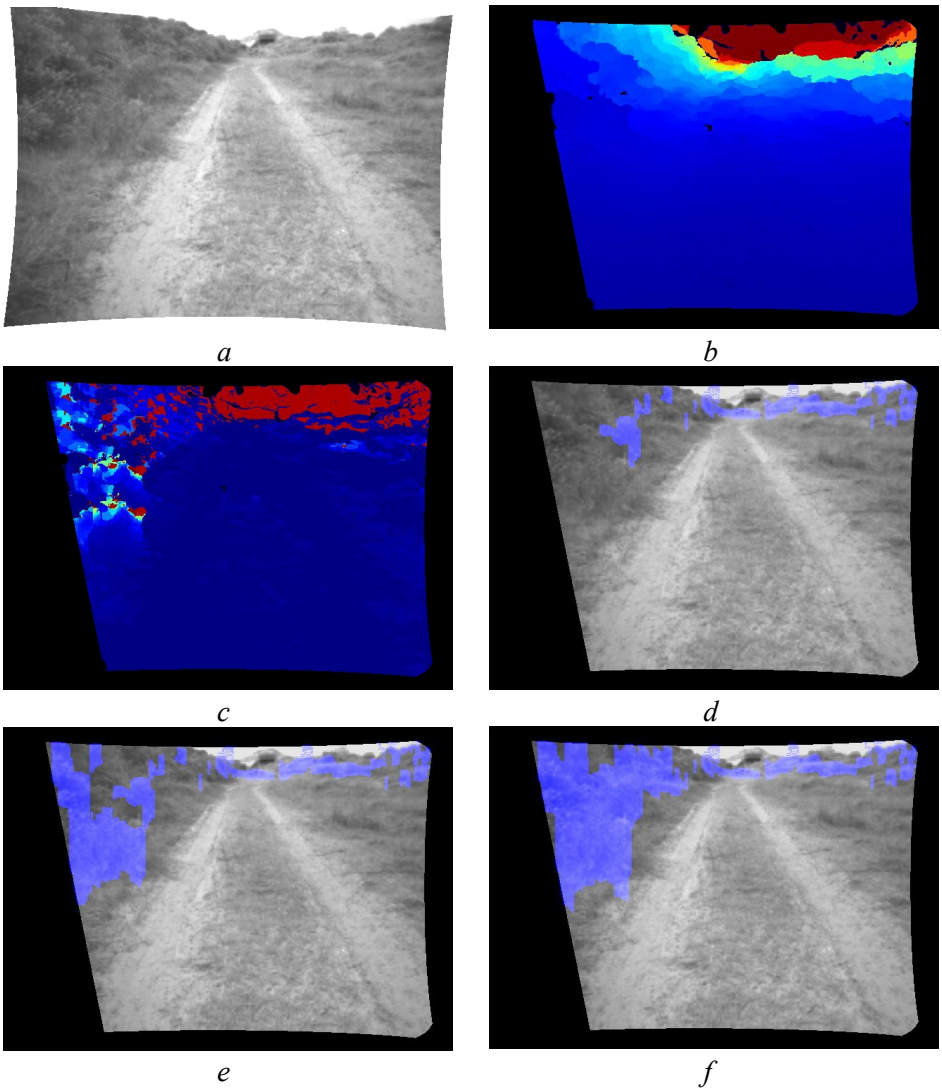


Figure 4.14: Original image (a), computed depth map (b), computed slope map(c), Regular threshold 0.75 (d), hysteresis thresholding [0.5 0.75] (e) and [0.25 0.75] (f).

4.2.3 Positive OD night-time

We now present positive OD performance on our night-time dataset. Figure 4.15 shows results obtained when using Rank preprocessing. Figure 4.16 shows the result when using LoG preprocessing. It can be seen that performance is unacceptable if all distances up to 50 m are considered. However the ROC plots for 35 and 25 m show good performance for those distances.

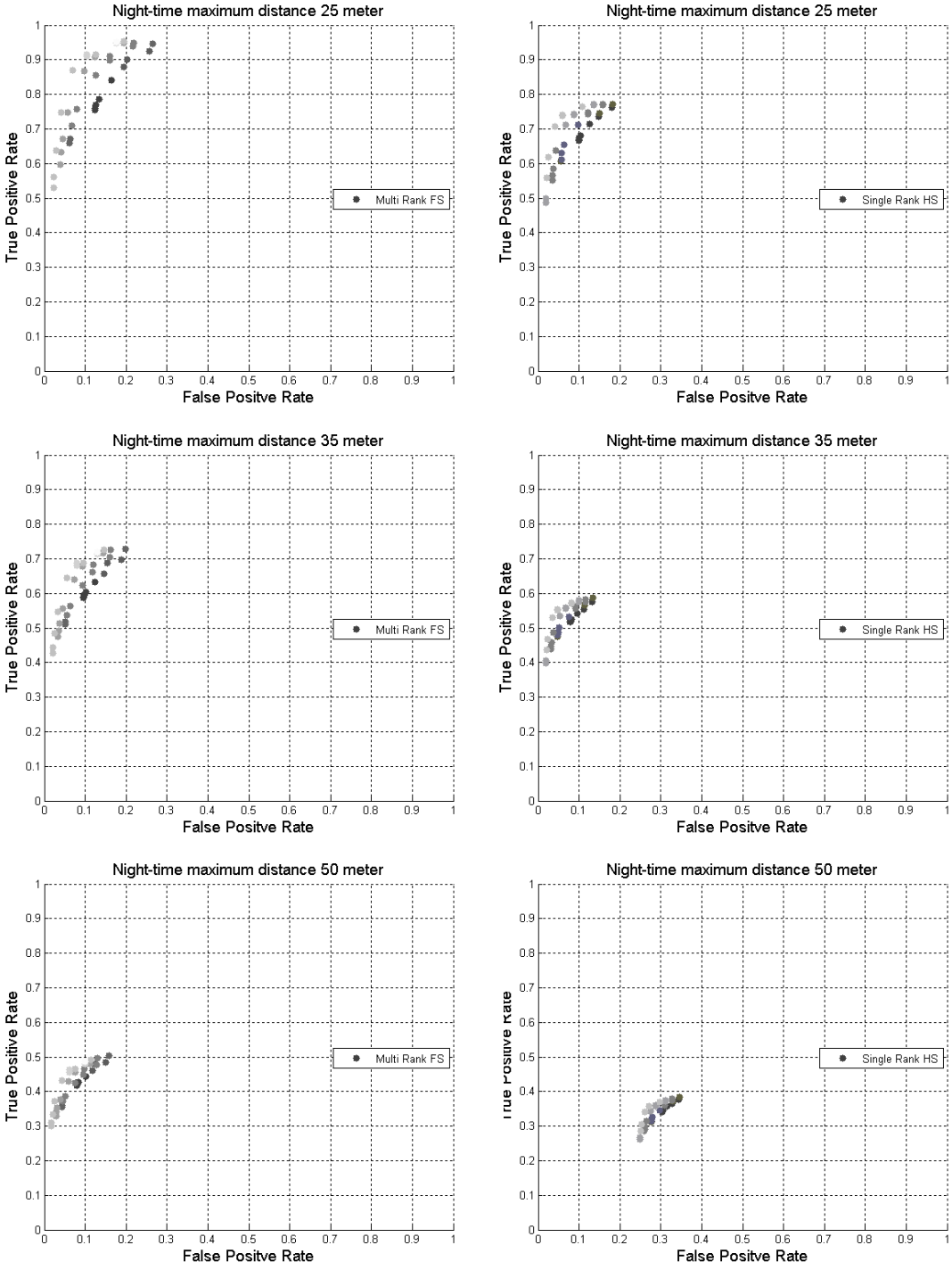


Figure 4.15: Positive OD performance night-time using Rank.

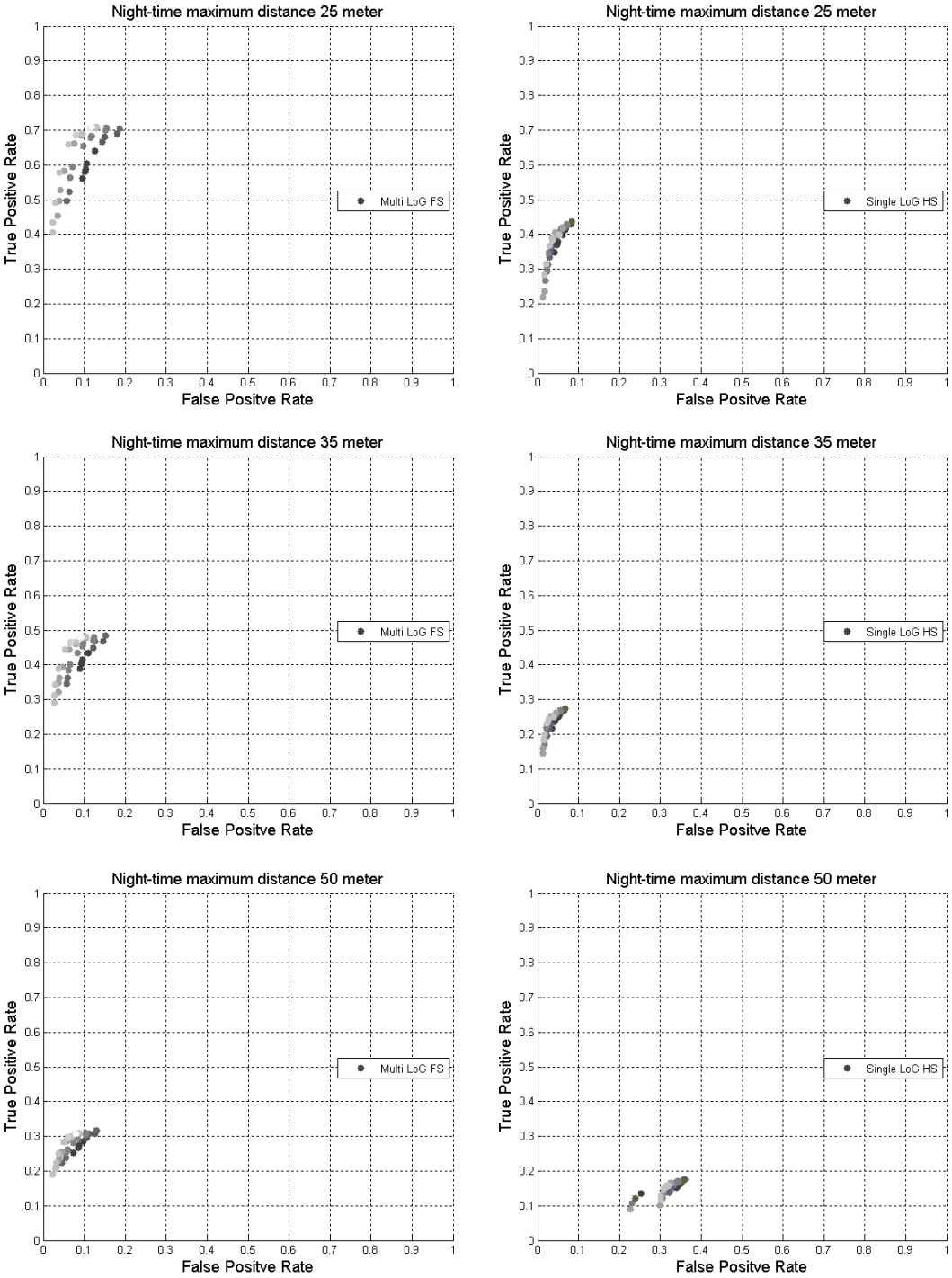
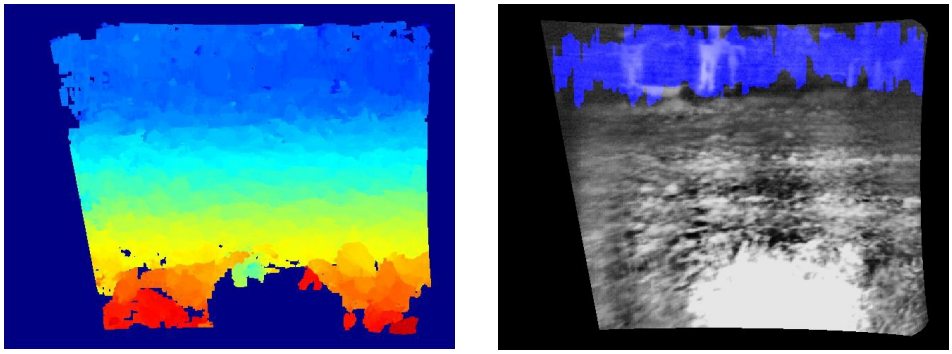


Figure 4.16: Positive OD night-time LoG.

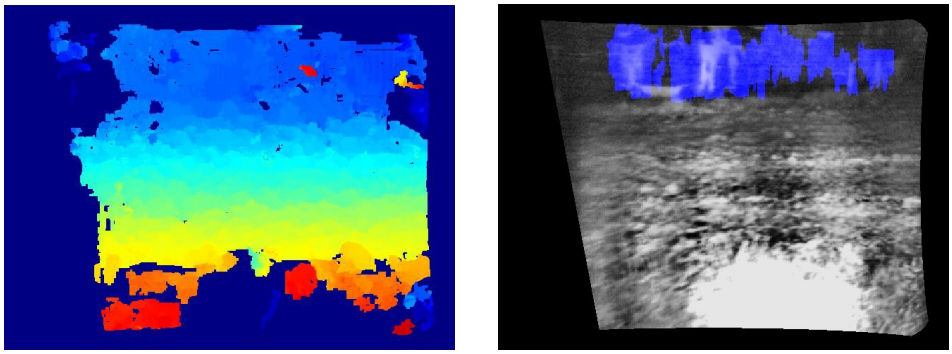
In the plots on the previous pages we see the influence of the tested depth estimation techniques on night-time positive OD. There is a clear difference between the true detection rate of the single-scale and multi-scale configurations. Also the choice between LoG and Rank preprocessing has a clear impact on positive OD performance. The performance gain between depth estimation configurations in true detection rate is about the same as the gain in depth coverage for the given configurations. This indicates that the extra depth information found using the multi-scale approaches has sufficient quality to increase the detection of positive obstacles. We observe that only the configuration that uses Rank preprocessing and our multi-scale approach reaches acceptable (0.85, 0.1) performance levels for obstacles up to 25 m.

The influence of step-height and hysteresis thresholding is consistent with previous results. However, now their actual value becomes more crucial. The tests using small step-heights have considerably higher false detection rates. Also the influence of hysteresis thresholding becomes important during low visibility conditions. While it is hard to observe in the plots, a large step height e.g. *0.45* meter and modest hysteresis thresholding e.g. *seed <= 0.75 grow <= 0.30* boosts performance to acceptable (0.85, 0.1) levels.

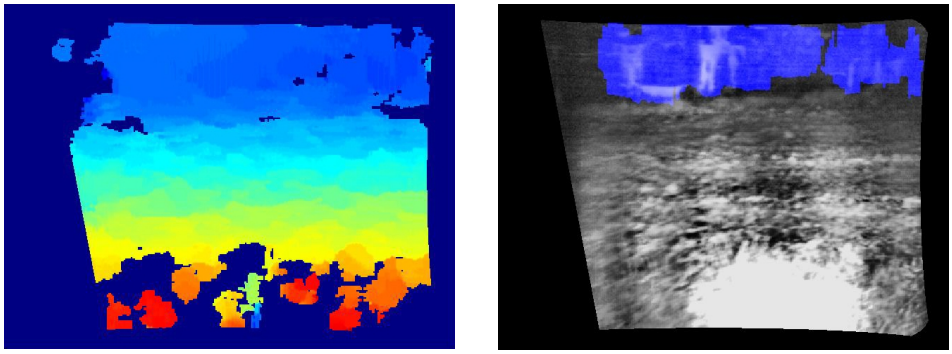
Figure 4.17 shows the night-time OD results for various depth estimation configurations. It is the same stereo image pair as used in figure 4.2. What these images demonstrate is that increased depth coverage alone does not increase OD performance. Depth consistency over the obstacle is also an important aspect. This is because positive OD is based on grouping of pixels at equal depth and measuring the size of the found candidate objects. If an obstacle is subdivided into smaller sub-objects, due to errors or inconsistent depths, these sub-objects might not pass the obstacle size thresholds. In section 4.1.3 we have seen that significant dilation errors near object borders can occur. We believe that this does not have to have a negative effect on positive OD performance as such. It mainly will increase the false detection rate. In the plots in figure 4.15 we can see, that given the right parameters, false detections can be kept to an acceptable level.



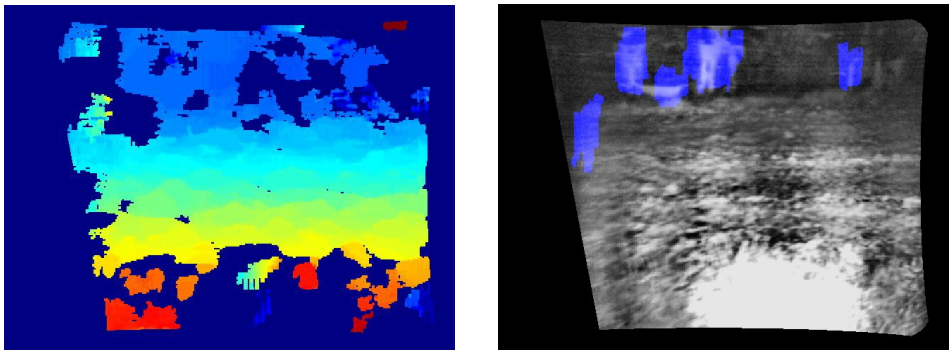
IR Multi RANK FS *OD Result*



IR Multi LoG FS *OD Result*



IR Single RANK HS *OD Result*



IR Single LoG HS *OD Result*

Figure 4.17: Night-time positive OD results, disparity maps (left), obstacle maps(right).

4.2.4 Negative OD day-time

Now we turn our attention to negative obstacle detection. In contrast to the good results on positive obstacles, negative obstacles pose more challenges, as can be seen in the plots below. While we get adequate performance for small distances up to 5m, the false detection rate becomes unacceptable at larger distances .

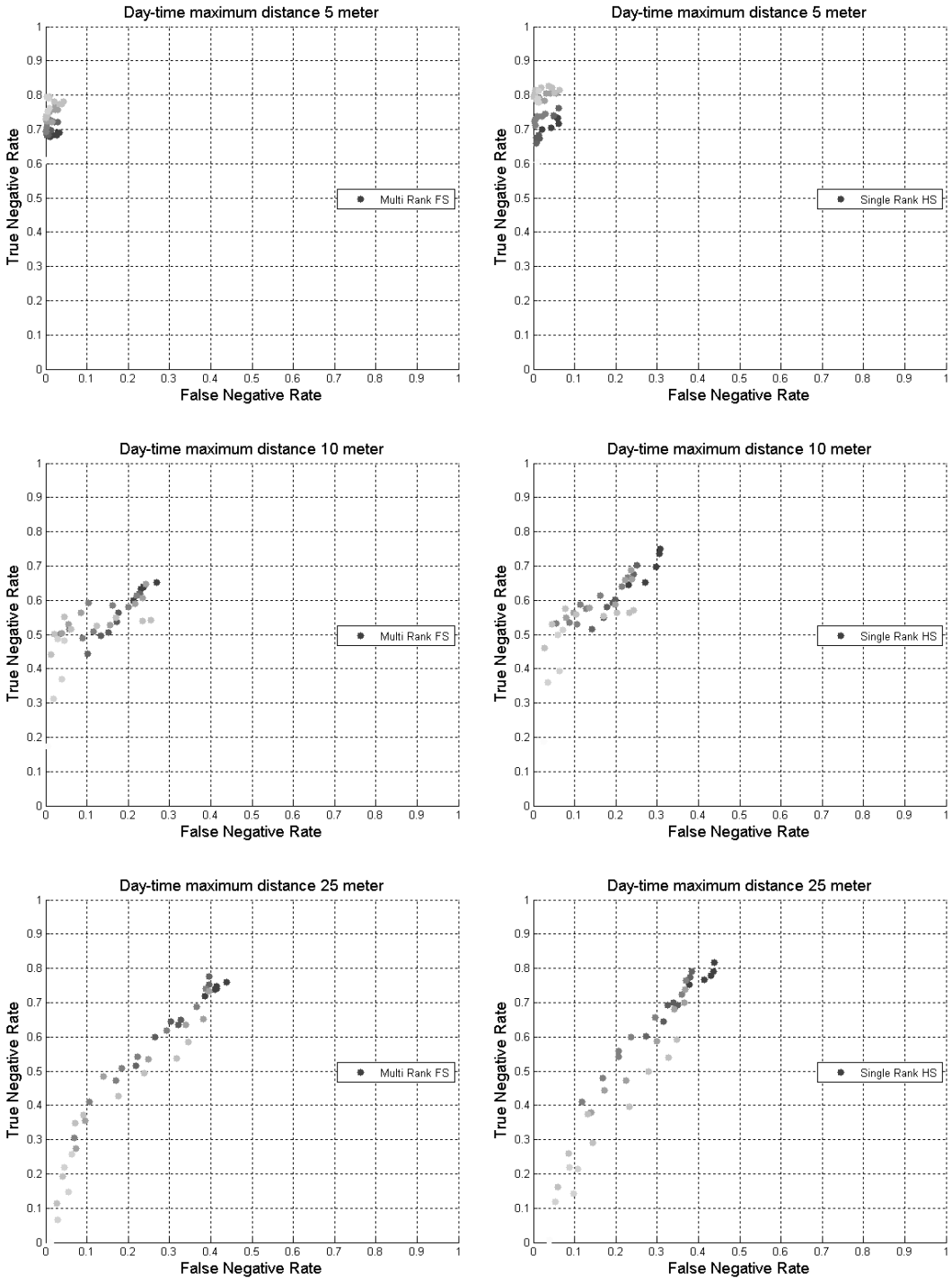


Figure 4.18: Negative OD day-time.

Before we start analysing the results we first note the following aspects. Our dataset only contains one negative obstacle that the vehicle cannot traverse, see figure 4.19. This negative obstacle is relatively easy to detect for the following reasons. Firstly, the distance between the far-side and near-side is about 6 meters. Furthermore, the far-side is higher than the near-side and can be seen clearly even from large distances. Finally, the ditch is a few meters deep. Because of these facts we believe the true negative detection rates are not as representative as those of positive obstacle detection. Nevertheless, interesting observations can be made from these plots.

The graphs on the previous page and figure 4.19 clearly show the dilemma of choosing an appropriate step-height when detecting negative obstacles. Small step-heights allow detection of negative obstacles at greater depths. Notice, that the darkest markers, from configurations using a small step height, obtain the best true-detection rate for lager distances. However, due to the ambiguous nature of negative obstacles a small step-height will also cause a false detection rate that is unacceptable. Larger step-heights have a considerable smaller false-detection rates for larger distance. For the range of 5 m they perform significantly better. Clearly, workable true negative detection rates can also be reached for larger distances. The main difficulty however is the false negative detection rate that can be unacceptably high.

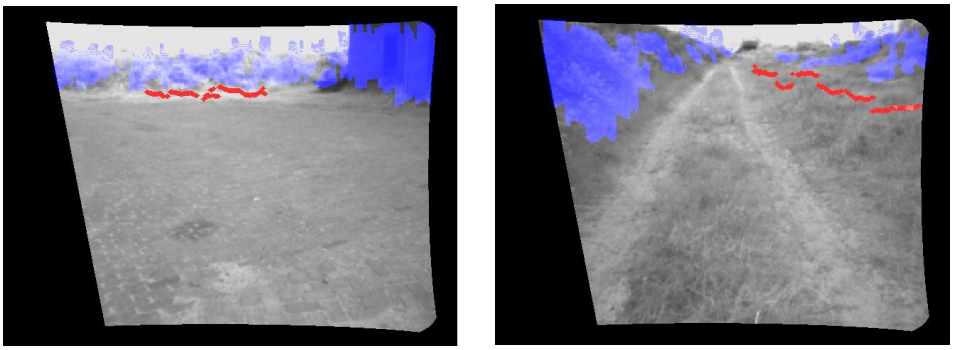


Figure 4.19: Problem of parameter choice. Using a step height of 0.45 meter and a hysteresis threshold of [2.0 3.0] we can detect the negative obstacle (left). However, false detections can appear e.g. on top of grass polls (right).

4.2.5 Negative OD night-time

We now present the negative OD result during night-time conditions. Figure 4.20 shows the quantitative results obtained using our night-time dataset. In figure 4.21 and 4.22 we show obstacle maps generated during day- and night-time conditions for the negative obstacle in our dataset.

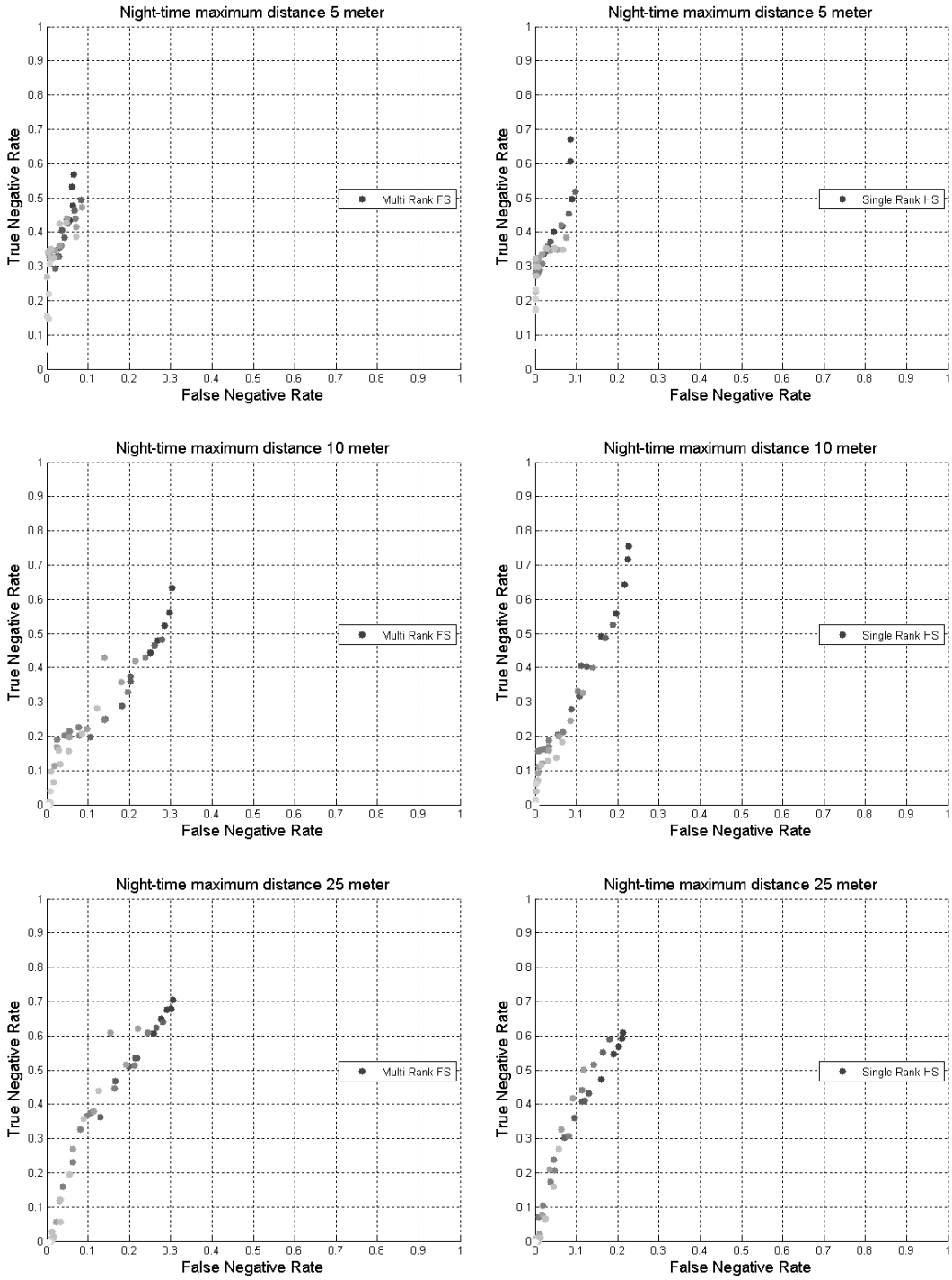


Figure 4.20: Negative OD night-time.

For nearby distances we observe a drop in true detection rate. Furthermore, now the smaller step-heights seem to achieve better performance. For the single negative obstacle in our dataset, depth dilation plays an important role, see figure 4.22. As discussed in section 4.1.3 the bushes on the near-side of the gap will appear larger during the night. This will cause the near side of the ditch to appear smaller in width, effectively leaving less room for negative obstacle detection. In most cases the object will be detected. However, the width of the object is less than during day-time conditions. Because the lack in negative obstacle detection is compensated by the increase in positive obstacle size, the effect on the operation of the vehicle can be minimal (for this particular negative obstacle). Furthermore, it seems that the position of the negative obstacle is located less precise during night-time conditions. It is shifted upwards (see fig. 4.22 at 10 m) by the depth dilation effect on the negative obstacle's near-edge. When it is shifted downwards (see fig. 4.21) it is usually caused by the lack of depth estimates at the negative obstacle's near-edge. As with day-time conditions, acceptable true detection rates during the night can be achieved for this negative obstacle. Again it is the false detection rate that makes the real-world usability troublesome.

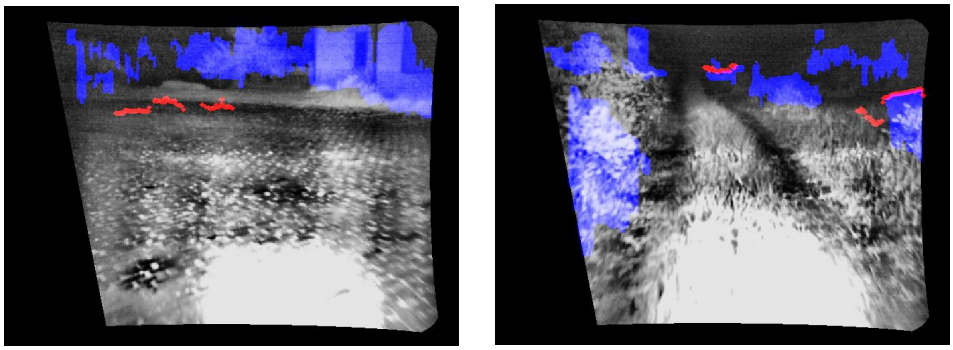


Figure 4.21:Problem of parameter choice. Using a step height of 0.45 meter and a hysteresis threshold of [2.0 3.0] we can detect the negative obstacle (left). However, false detections can appear e.g. on top of grass polls (right).

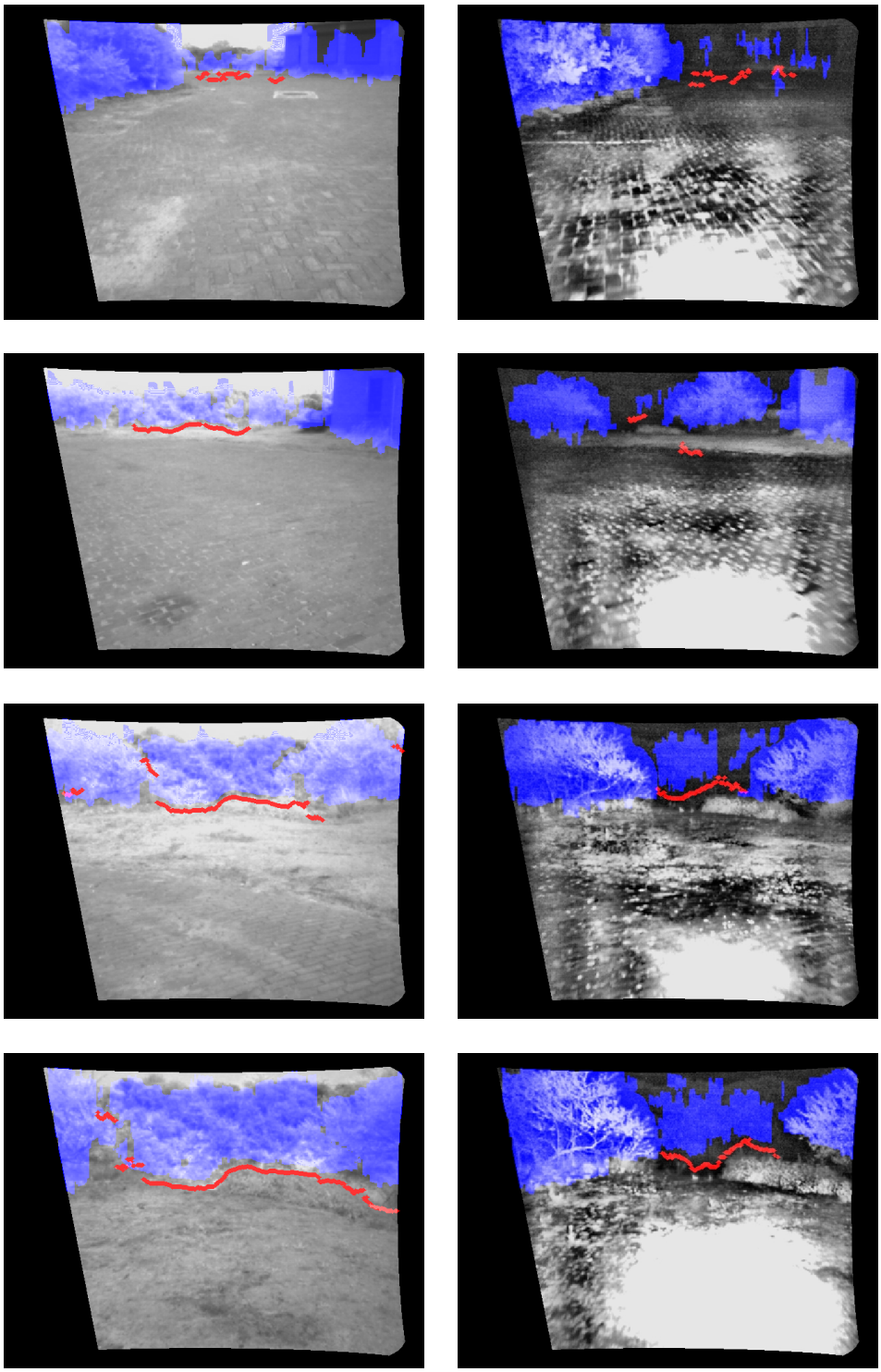


Figure 4.22: Result with OD configuration 5 using Rank Multi-scale (from top to bottom: 50m 30m 10m 5m).

Chapter 5

Discussion & Conclusion

This research has shown that it is possible to reliably detect positive obstacles during day and night conditions using stereo vision. We have quantitatively evaluated the performance of our system over a large real-world dataset. Obstacle detection (OD) at night requires robust depth estimation which we achieve by our fine-to-coarse disparity estimation procedure. Our fine-to-coarse approach does not suffer from traditional coarse-to-fine error propagation and is based on the ability to make a distinction between good and bad stereo matches. An important contribution is our disparity validity measure. Which, in contrast to traditional cost-based measures is able to reliably distinguish between good and bad stereo matches. The detection of positive and negative obstacles can be done robustly by depth dependent slope measurement and obstacle thresholds. The application of hysteresis thresholding is a powerful tool for boosting OD performance without significant increase in the computational load. In the next sections we will discuss our results in more detail.

5.1 Depth estimation

Our novel fine-to-coarse disparity estimation processes makes it possible to achieve acceptable high-quality disparity coverage levels during day and night conditions. The performance has been measured over a large real-world dataset showing an increase in depth coverage and decrease in depth uncertainty. We have shown that it can be used during the night when the scene is illuminated by a 36 Watt IR spotlight. Of course, results obtained rely heavily on the intensity of the IR-emitters. Increasing their wattages would improve performance. However, the goal of this research is to investigate the performance during low visibility conditions. The use of a single light source with limited output is typical for low visibility conditions. Currently we have no implementation of the novel system that runs in real-time. However, we believe that the novel method can run real-time due to the following reasons. First, the multi-resolution approach is based on a single-resolution approach that has been proven to run real-time. The overhead is only based on the fine-to-coarse level selection, bi-linear interpolation and computation of the novel disparity validity measure, which can be performed fast. Apart from the fast fine-to-coarse level selection, there are no computational dependencies between the levels in the stereo image pyramid making parallelization a powerful option. Also the ability to adaptively balance the computational load based on the visibility conditions make it appropriate for use in autonomous systems.

Our tests show that cost-based disparity validity measures are poor predictors of the correctness of a disparity estimate. Our novel validity measure that balances intensity edges, signal-to-noise ratio and the final disparity smoothness outperforms these approaches The advantage of our method is that it combines intensity and disparity information into one value in an intuitive way. Firstly, it is based on the assumptions that disparity on an object's surface is smooth. Secondly, object borders are likely to cause disparity jumps. And thirdly, object borders often result in intensity edges which represent a positive signal to noise ratio. The strength of our validity measure is that it considers all these assumptions to form one indication about the correctness of a disparity estimate.

LoG filtering is an often used pre-processing step. This research points out that in the case of depth estimation for autonomous land vehicles Rank preprocessing is the better option. While Census might even work better, a Rank transform window of 16x16 can still be represented by 8 bits. This allows every disparity estimation system that uses 8-bit grey-scale as input to incorporate the Rank transform. While Census is more challenging to implement for real-time processing.

5.2 Obstacle Detection

To our knowledge this is the first research effort that investigates obstacle detection on a large real-world data-set in a true quantitative way. The results show that positive obstacle detection during low visibility conditions requires reliable depth estimation. Our novel multi-resolution stereo matching approach can provide this robustness through acceptable depth coverage and low depth uncertainty.

The positive obstacle detection method itself poses little fundamental problems. The only factor on its performance is the quality of the estimated disparity. Trivially, if no reliable disparity estimated is found, no obstacles can be detected. This aspect is inherently linked with geometrical based obstacle detection. The benefits of our approach is its ability to increase true detection rates without increasing false detection rates. For this we use depth dependent slope measurement and hysteresis thresholding. When the angle between, the plane defined by the optical axis and the image column, with the surface normal is large. Hysteresis thresholding can be a powerful tool to increase the performance Which makes it an alternative for more computational demanding approaches like that of Talukder [58]. Increasing performance would most likely require 3D reasoning about terrain traversability. Also extracting more semantic knowledge about the terrain and its objects is an interesting prospect.

We have shown that our method can detect negative obstacles at modest range during day and night conditions without the use of narrow FOV cameras. Furthermore, it does not rely on line fitting as used by Matthies et al. [38]. Instead it looks for suspicious (uncertainty corrected) depth jumps, which can be done fast and efficient. Having said that, we must accept the fundamental drawbacks of geometrical based approaches for negative obstacle detection. The main difficulty we encountered is unacceptable false detection rates. We found that on flat terrain the false detections are minimal. However, on bumpy terrain with for example stretches of grass polls false detections occur often. Based on the pure geometrical input, the algorithm can not distinguish between a true negative obstacle and a false detection. Discarding false detections most likely will require obtaining more contextual information of the local and global terrain as well as learned knowledge. We also note that unfortunately our results were obtained on one abundantly clear negative object. The performance on less obvious negative obstacles is not comparable and most likely not as good.

5.3 Evaluation

This research makes a first step towards an efficient and reliable OD evaluation benchmarking environment for the research community. We quantitatively evaluated day and night-time performance for different OD approaches on a large real-world dataset. In our opinion this is the only manner in which conclusions can be made about the performance of the wide range of existing OD methods. Because the lack of existing benchmark datasets and evaluation methods, substantial work was dedicated to setting up a benchmark environment. In the future we hope to improve both our datasets and our evaluation methods to come to an even more reliable and effective benchmarking environment for the OD research community.

Most researchers present qualitative results in the form of object label maps. However, to truly compare different techniques amongst each other using different configuration and settings requires quantitative evaluation. Initially we experimented with pixel based labelling and classification. As discussed in section 3.4.3 this evaluation method favours systems which only classify large object close to the vehicle. As an alternative we tried object based labelling and classification. The main problem with this scheme is the labelling of objects in the scene due to cluttering. Labelling every bush over a stretch of undergrowth is a daunting task. Furthermore labelling the whole stretch of undergrowth as one single object will not give good insight into the performance. Finally we used our surface based evaluation. It allows for efficient labelling only using three classes (positive obstacle, negative obstacle, drivable terrain and ignore). Also surface based performance measures give a good indication of the performance differences between methods.

Chapter 6

Future work

To reach the performance needed to effectively use autonomous land vehicles for a wide range of scenarios still requires substantial research. In this chapter we present some possible topics for further research that are related to our research described in this thesis.

6.1 Disparity estimation & Information Fusion

As we have described in section 2.5 disparity is only one source of information for an autonomous vehicle. Also, other image modalities like colour and texture grasp valuable information about the scene and its objects. Most likely we will need all these image modalities to form an environmental model that is reliable and comprehensive enough for effective autonomous vehicle operation. Often disparity estimation, colour based classification and texture based classification are seen as separate problems. We argue that a lot of performance can be gained by exchanging information at early stages when solving these (traditionally separated) problems. For instance texture based terrain classification requires using filters at various scales and orientations. As research shows, Olson [47], the distance to an object can be used to select the appropriate texture-filter scale. This reduces the amount of filters that have to be used, which simplifies and enhances the task of texture recognition. On the other hand incorporating texture information in the disparity estimation process can increase the quality of the disparity map, see figure 6.1. Here we used texture to segment the intensity image. This segmentation together with the disparity map is used to enhance the disparity estimation. Especially the performance gain on vertical poles is interesting. During further research we would like to investigate the potential of using texture segmentation to enhance disparity estimation results. Also using depth information to enhance texture based obstacle classification (e.g. distinguishing between a rock or a grass poll) is promising.

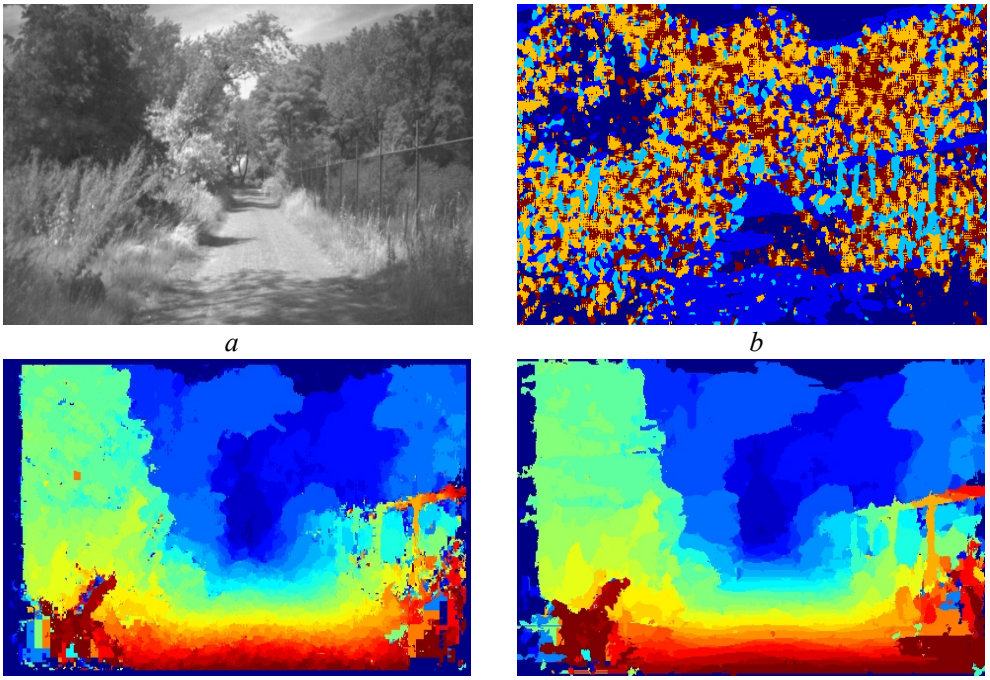


Figure 6.1: Using texture segmentation to improve disparity estimate. Left image of stereo image pair (a) and its texture segmentation (b), disparity map (c), texture enhanced disparity map (d).

6.2 Obstacle detection

Significant more research has to be dedicated to negative obstacle detection. To facilitate this we will have to record images from a larger variety of real-world negative obstacles. It is likely that using narrow FOV cameras is needed for detecting negative obstacles at large distances. We will probably have to accept that the true traversability of some negative obstacles can only be obtained at close distance. For positive obstacles we hope to lower the false detection rate by looking at other image modalities such as colour and texture. Also methods that fit a ground plane after the obstacles have been found can enhance OD performance. Methods that obtain and weight information based on more then one frame such as SLAM are also promising for off-road OD. Given our benchmark environment it would be interesting to measure the performance of other OD approaches. Especially the approach proposed by Talukder et al. [58] and V-disparity methods like that of Broggi et al. [7] are interesting. To our knowledge comparing a wide range of OD methods in a quantitative way on a large real-world dataset has not been done before. Making such a comparison using our benchmark environment would be a great help to the OD research community.

6.3 Evaluation

The main problem of our surface based evaluation is the lack of surface ground truth. We experimented using estimated depth to compute the surface of all positive objects and non-objects in the scene. The estimated depth of object pixels influence their estimated surface. Structural bias in depth estimates can skew the results. The correctness of our performance measuring is based on the assumptions that if bias occurs it will be on average the same for day and night conditions and the same for different depth estimation configurations. If this is not the case the OD comparison between various approaches is difficult. If for instance LoG filtering has a structural negative bias compared to Rank transforming then this would skew the performance measures in favour for Rank preprocessing. This is due the fact that pixel at closer depth represents less surface. We tried to get insight into the quality and possible biases of the actual depth estimates. We did this by using the GPS read-out accompanying the frames and satellite imagery of our test terrain. Unfortunately this was not successful. Mainly due the imprecision in GPS coordinates from both the GPS system and the satellite imagery. While we have no indication that their exists a structural bias between methods, the only way to truly investigate the possible bias in depth estimates requires additional testing. Also from the aspect of obstacle depth precision, measuring depth biases is an interesting topic.

Bibliography

[1] J. Banks and P. Corke, "Quantitative Evaluation of Matching Methods and Validity Measures for Stereo Vision", *The International Journal of Robotics Research*, Vol. 20, Issue 7, pp 512-532, 2001.

[2] P. Bellutta, R. Manduchi, L. Matthies, K. Owens, A. Rankin, "Terrain perception for DEMO III ", *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 326-331, 2000.

[3] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, Issue 4, pp. 401-406, 1998.

[4] A.F. Bobick, S.S.Intille, "Large occlusion stereo", *International Journal of Computer Vision*, Vol. 1, pp. 7-55, 1999.

[5] Y. Boykov, O. Veksler, and R. Zabih, "A variable window approach to early vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, Issue 12, pp 1283-1294, 1998.

[6] Y. Boykov, O. Veksler, and R. Zabih. "Fast approximate energy minimization via graph cuts", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, Issue 11, pp. 1222–1239, 2001.

[7] A. Broggi, C. Caraffi, R. Isabella F. Grisleri, P. Grisleri, "Obstacle Detection with Stereo Vision for Off-Road Vehicle Navigation", *Proceedings of the IEEE International Workshop on Machine Vision for Intelligent Vehicles*, pp. 65, 2005.

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

[9] L. Cohen, L. Vinet, P.T. Sander, A. Gagalowicz, "Hierarchical region based stereo matching", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 416-421, 1989.

[10] I. J. Cox, S. L. Hingorani, S. B. Rao, B. M. Maggs. "A maximum likelihood stereo algorithm". *Computer Vision and Image Understanding,* Vol. 63, Issue 3, pp. 542–567, 1996.

[11] E.R. Davies, "*Machine Vision Theory, Algorithms, Practicalities*", Elsevier, Third edition, 2005.

[12] C Elachi, "*Introduction to the physics and techniques of remote sensing*", Wiley New York, 1987.

[13] A. Fusiello, V. Roberto, and E. Trucco, "Symmetric stereo with multiple windowing", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 14, Issue 8, pp. 1053-1066, 2000.

[14] R.C. Gonzalez, R.E. Woods, "*Digital Image Processing*", Prentice Hall, Second edition, 2001.

[15] W. E. L. Grimson, "Computational experiments with a feature based stereo algorithm", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 7, Issue 1, pp. 17-34, 1985.

[16] A. Hanspeter, H.A. Mallot, S. Gillner, P.A. Arndt, "Is correspondence search in human stereo vision a coarse-to-fine process ?", *Biological cybernetics*, Vol. 74, Issue 2, pp 95-106, 1996.

[17] R. Hartley,A. Zisserman, "*Multiple View Geometry in Computer Vision*", Second edition, Cambridge University Press, 2004.

[18] M. Hebert, N. Vandapel, S. Keller, and R.R. Donamukkala, "Evaluation and comparison of terrain classification techniques from LADAR data for autonomous navigation", *Army Science Conference*, Orlando, 2002.

[19] J. Heikkilä, O, Silvén, "A Four-step Camera Calibration Procedure with Implicit Image Correction", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 1106-1112, 1997.

[20] H. Hirschmüller, P. R. Innocent, J. Garibaldi, "Real-Time Correlation- Based Stereo Vision with Reduced Border Errors", *International Journal of Computer Vision* ,Vol. 47, pp. 229-246, 2002.

[21] P.V.C. Hough, "Method and means for recognising complex patterns", US Patent 3069654, 1962.

[22] P. Jansen, W. van der Mark, J.C. van den Heuvel, F.C.A Groen, "Colour based off-road environment and terrain type classification ", *Proceedings IEEE Conference on Intelligent Transportation Systems*, pp. 216- 221, 2005.

[23] D. G. Jones and J. Malik,"A computational framework for determining stereo correspondence from a set of linear spatial filters", *European Conference On Computer Vision*, pp. 395–410, 1992.

[24] T. Kanade, M. Okutomi, “A stereo matching algorithm with an adaptive window: theory and experiment”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, Issue 9, pp 920-932, 1994.

[25] S. B. Kang, R. Szeliski, and J. Chai, “Handling occlusions in dense multi-view stereo”, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 103-110, 2001.

[26] J. Koenderink, "The structure of images", *Biological Cybernetics*, Vol. 50, pp 363–370, 1984.

[27] V. Kolmogorov and R. Zabih, “Computing visual correspondence with occlusions using graph cuts”, *ICCV*, volume II, pages 508–515, 2001.

[28] G. Kraft and P.P. Jonker, “Real-Time Stereo with Dense Output by a SIMD Computed Dynamic Programming Algorithm”, *International Conference on Parallel and Distributed Processing Techniques and Applications*, Vol. III, pp. 1031-1036 , 2002.

[29] R. Labayrade and D. Aubert, “In-vehicle obstacle detection and characterization by stereovision”, *Proceedings of the International Workshop on In-Vehicle Cognitive Computer Vision*, 2003.

[30] R. Labayrade, D. Aubert, and J.-P. Tarel, “Real time obstacle detection on non flat road geometry through V-disparity representation”, *IEEE Intelligent Vehicles Symposium, Versailles*, pp. 646–651, June 2002.

[31] A. Lacaze, K. Murphy, and M. DelGiorno, “Autonomous mobility for the DEMO III Experimental Unmanned Vehicle”, *Association for Unmanned Vehicle Systems–Unmanned Vehicle 2002*.

[32] JF. Lalonde, N. Vandapel, M Hebert,”Data Structure for efficient Processing in 3-D”, *Robotics: Science and Systems 1*, 2005.

[33] K.I. Laws, “Rapid texture identification”, *SPIE Image processing for missile guidance*, Vol. 238, pp. 376-380, 1980.

[34] J. P. Lewis, “Fast normalized cross-correlation,” *Vision Interface*, pp. 120-123, 1995.

[35] T. Lindeberg, “*Scale-Space Theory in Computer Vision*”, Kluwer Academic Publishers, 1994.

[36] R. Manduchi, A. Castano, A. Talukder, L. Matthies, “Obstacle Detection and Terrain Classification for Autonomous Off-road Navigation”*, Autonomous Robots* ,Vol. 18, Issue 1, pp. 81-102, 2005.

[37] L. Matthies, P. Grandjean, “Stochastic Performance Modeling and Evaluation of Obstacle Detectability with Imaging Range Sensors”, *IEEE Transactions on Robotics and Automation, Special Issue on Perception-based Real World Navigation*, Vol. 10, Issue 6, pp. 783-792, 1994.

[38] L. Matthies, A. Kelly, T. Litwin, G. Tharp, "Obstacle Detection for Unmanned Ground Vehicles: A Progress Report", *Robotics research 7*, Springer-Verlag.

[39] L. Matthies, T. Litwin, K. Owens, A. Rankin, K. Murphy, D. Coombs, J. Gilsinn, T. Hong, S. Legowik, M. Nashman, B. Yoshimi, "Performance Evaluation of UGV Obstacle Detection with CCD/FLIR Stereo Vision and LADAR", *IEEE ISIC / CIRA / ISAS Joint Conference*, 1998.

[40] L. Matthies, A. Rankin,”Negative Obstacle Detection by Thermal Signature”, *IEEE Conference on Intelligent Robots and Systems*, Vol. 1, pp. 906-913, 2003.

[41] W. van der Mark, D.M. Gavrila, “Real-time dense stereo for intelligent vehicles“, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7, Issue 1, pp. 38- 50, 2006.

[42] H. Moravec, “Toward automatic visual obstacle avoidance,” *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pp. 584–590, August 1977.

[43] K. Mühlmann, D. Maier, J. Hesser, R. M¨anner, “Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation”, *International Journal of Computer Vision,* Vol. 47, Nr. 1-3, pp. 79-88, April - June 2002.

[44] Y. Ohta and T. Kanade. Stereo by intra- and interscanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 7 Issue 2, pp. 139–154, 1985.

[45] M. Okutomi and T. Kanade, “A locally adaptive window for signal matching”, *International Journal of Computer Vision*, Vol. 7, Issue 2, pp 143-162, 1992.

[46] C.G. Olson, ”Variable-scale smoothing and edge detection guided by stereoscopy”, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* pp. 80-85, 1998.

[47] C.G. Olson, ”Adaptive-scale filtering and feature detection using range data“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, Issue 9, pp 983-991, 2000.

[48] M. O'Neill, M. Denos, "Automated system for coarse-to-fine pyramidal area correlation stereo matching", *Image and Vision Computing*, Vol. 14, Issue 3, pp. 225-236, 1996,

[49] K. Owens, L. Matthies, "Passive Night Vision Sensor Comparison for Unmanned Ground Vehicle Stereo Vision Navigation", *IEEE International Conference on Robotics & Automation*, Vol. 1, pp. 122-131, 2000.

[50] T. Randen, J.H. Husoy, "Filtering for texture classification: a comparative study", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, Issue 4, pp. 291-310, 1999.

[51] N. Roma, J. Santos-Victor, J. Tomé, "A comparative analysis of cross-correlation matching algorithms using a pyramidal resolution approach", *Empirical Evaluation Methods in Computer Vision*, pp 117-142, 2002.

[52] H. Samet, "*The Design and Analysis of Spatial Data Structures*", Addison-Wesley, 1989.

[53] D. Scharstein, R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms", *International Journal of Computer Vision,* Vol. 47, Issue 1-3, pp. 7-42, 2002.

[54] D. Scharstein and R. Szeliski. "Stereo matching with nonlinear diffusion", *International Journal of Computer Vision*, Vol. 28, Issue2, pp 155–174, 1998.

[55] D. Scharstein, "Matching images by comparing their gradient fields", *IEEE International Conference On Pattern Recognition*, Vol. 1, pp 572–575, 1994.

[56] H.S. Smallman, "Fine-to-coarse Scale Disambiguation in Stereopsis", *Vision Research*, Vol. 35, Issue 8, pp 1047-1060, 1995.

[57] J. Sun, H.Y. Shum, N. N. Zheng. "Stereo matching using belief propagation", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 25, Issue 7, pp. 787-800, 2003.

[58] A. Talukder, R. Manduchi, A. Rankin, L. Matthies, "Fast and Reliable Obstacle Detection and Segmentation for Cross-country Navigation", *IEEE Intelligent Vehicle Symposium*, Vol.2, pp. 610-618, 2002.

[59] S. Thrun, W. Burgard, D. Fox, "*Probabilistic Robotics* ", MIT Press, 2005.

[60] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, P. Mahoney, "Stanley, the robot that won the DARPA Grand Challenge", *Journal of Field Robotics*, Vol. 23", Issue 9, pp. 661-692, 2006.

[61] A. Witkin, "Scale-space filtering", *Conference on Artificial Intelligence*, pp 1019–1022, 1983.

[62] A. Witkin, D. Terzopoulos, M. Kass, "Signal matching through scale space", *International Journal of Computer Vision*, Vol. 1, Issue 2, pp.133-144, 1987.

[63] Y. Yang, A. Yuille, "Multilevel enhancement and detection of stereo disparity surfaces", *Artificial Intelligence*, Issue 78, pp. 121-145, 1995.

[64] Y. Yang, A. Yuille, J. Lu, "Local, global, and multilevel stereo matching", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* pp. 274-279, 1993.

[65] R. Zabih and J. Woodfill,"Non-parametric local transforms for computing visual correspondence", *European Conference On Computer Vision*, vol. 2, pp 151–158, 1994.

[66] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations", *Proceedings of IEEE international conference on computer vision*, pp. 666-673, 1999.