Efficient stereo matching and obstacle detection using edges in images from a moving vehicle

by

Dexmont Alejandro Peña Carrillo B.Sc. M.Sc.

A thesis submitted in fullfilment of the requirements
for the degree of Doctor of Philosophy

to the

Dublin City University

Faculty of Engineering and Computing, School of Computing

Supervisor: Dr. Alistair Sutherland
Dr. Jennifer Foster

March 2017

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: _____ ID No.: 12212250 Date: March, 2017

Acknowledgements

I would like to thank all the people who supported me over the course of this venture and the previous ones which led me to enrol in Dublin City University.

Firstly I would like to thank my supervisor Dr. Alistair Sutherland who became a friend and provided me plenty of support and great comments to guide my research.

Second to my family, who from home were always supporting me and listening whenever I needed it.

I also would like to thanks to my examiners in the transfer process Kevin Casey and Ovidiu Ghita whose comments helped me to increase the quality of my research.

To my colleagues Fattah Alizadeh and Alireza Dehghani who share their experiences since the first day I arrived to Ireland.

To my collegues Marlon Oliveira, Fiona Dermody and all the guys in the bay who became friends and provided a nice environment.

This thesis was funded by the Irish Research Council under the EMBARK initiative, application No RS/2012/489.

dexmont

Dublin, Ireland

March, 2017

Contents

List of Figures

List of Tables

Abbreviation list

AD Absolute Difference. 26, 27, 32

BP Belief Propagation. 34

BT Birchfield and Tomasi dissimilarity. 26, 28, 37

CRT Complete Rank Transform. 4, 9, 10, 16, 19, 29, 50–52, 55–57, 70–77, 95–100, 102, 108, 129, 130

CSS Curvature Scale Space. 24

CT Census Transform. 4, 9, 10, 16, 18–21, 29, 50–55, 57, 58, 60–68, 70, 71, 74–77, 88, 93–98, 129

DEM Digital Elevation Maps. 43, 44

DP Dynamic Programming. 34–36, 43, 45, 46, 82, 107, 126

DSI Disparity Space Image. 41

ED Edge Drawing. 79, 80, 83–86, 88, 89, 92, 105–107, 109

EDPF Edge Drawing Parameter Free. 68, 70

FPGA Field-Programmable Gate Array. 38, 39, 45, 126

GPU Graphics Processing Unit. 38, 39, 46, 59, 106, 108, 110, 126, 127, 130

HOG Histograms of Oriented Gradients. 23

IPKR Inverse Peak Ratio. 58

LIDAR Light Detection and Ranging. 4, 6, 39, 40, 43, 57, 92

LoG Laplacian of Gaussian. 16, 17

LRC Left-Right Consistency Check. 31, 33, 38, 58, 59

NCC Normalized Cross Correlation. 26, 27

NSCT Non-Subsampled Contourlet Transform. 16, 20

PKR Peak Ratio. 33, 99

PKRN Peak Ratio Naive. 33

RT Rank Transform. 16, 18, 19, 29, 55, 56

SAD Sum of Absolute Differences. 26, 27, 29, 71, 74, 75, 77

SED Simultaneous Edge Drawing. 83, 84, 90–92, 97, 102, 105–111, 113, 116

SGM Semi-Global Matching. 36, 45, 46, 110

SIMD Single Instruction Multiple Data. 106, 124, 127, 130

SLAM Simultaneous Localization and Mapping. 131

SSD Sum of Squared Differences. 26, 27, 29

TCT Thresholded Census Transform. 52, 53, 55–57, 59–71, 74–77

WTA Winner Takes All. 30, 35, 58

ZNCC Zero-mean Normalized Cross Correlation. 26, 28

ZSAD Zero-mean Sum of Absolute Differences. 26, 27

Publication List

[1] Dexmont Pena and Alistair Sutherland. Non-parametric image transforms for sparse disparity maps. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pages 291–294. IEEE, may 2015.

[2] Dexmont Pena and Alistair Sutherland. Disparity Estimation by Simultaneous Edge Drawing. In Jiwen Chen, Chu-SongLu and Kai-Kuang Ma, editors, *Computer Vision – ACCV 2016 Workshops*, page inPress. Springer International Publishing, Taipei, 2016.

[3] Dexmont Pena and Alistair Sutherland. Fast Obstacle Detection Using Sparse Edge-Based Disparity Maps. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 66–72, Stanford, CA, oct 2016. IEEE.

Efficient stereo matching and obstacle detection using edges in images from a moving vehicle

by

Dexmont Alejandro Peña Carrillo

Abstract

Fast and robust obstacle detection is a crucial task for autonomous mobile robots. Current approaches for obstacle detection in autonomous cars are based on the use of LIDAR or computer vision. In this thesis computer vision is selected due to its low-power and passive nature.

This thesis proposes the use of edges in images to reduce the required storage and processing. Most current approaches are based on dense maps, where all the pixels in the image are used, but this places a heavy load on the storage and processing capacity of the system. This makes dense approaches unsuitable for embedded systems, for which only limited amounts of memory and processing power are available. This motivates us to use sparse maps based on the edges in an image. Typically edge pixels represent a small percentage of the input image yet they are able to represent most of the image semantics. In this thesis two approaches for the use of edges to obtain disparity maps are proposed and one approach for identifying obstacles given edge-based disparities.

The first approach proposes a modification to the Census Transform in order to incorporate a similarity measure. This similarity measure behaves as a threshold on the gradient, resulting in the identification of high gradient areas. The identification of these high gradient areas helps to reduce the search space in an area-based stereo-matching approach. Additionally, the Complete Rank Transform is evaluated for the first time in the context of stereo-matching. An area-based local stereo-matching approach is used to evaluate and compare the performance of these pixel descriptors.

The second approach proposes a new approach for the computation of edge-disparities. Instead of first detecting the edges and then reducing the search space, the proposed approach detects the edges and computes the disparities at the same time. The approach extends the fast and robust Edge Drawing edge detector to run simultaneously across the stereo pair. By doing this the number of matched pixels and the required operations are reduced as the descriptors and costs are only computed for a fraction of the edge pixels (anchor points). Then the image gradient is used to propagate the disparities from the matched anchor points along the gradients, resulting in one-voxel wide chains of 3D points with connectivity information.

The third proposed algorithm takes as input edge-based disparity maps which are compact and yet retain the semantic representation of the captured scene. This approach estimates the ground plane, clusters the edges into individual obstacles and then computes the image stixels which allow the identification of the free and occupied space in the captured stereo-views. Previous approaches for the computation of stixels use dense disparity maps or occupancy grids. Moreover they are unable to identify more than one stixel per column, whereas our approach can. This means that it can identify partially-occluded objects. The proposed approach is tested on a public-domain dataset. Results for accuracy and performance are presented.

The obtained results show that by using image edges it is possible to reduce the required processing and storage while obtaining accuracies comparable to those obtained by dense approaches.

1

Introduction

1.1 Introduction

Autonomous vehicles are leaving the realm of science fiction to become real life products. Over the last decade tasks like autonomous parking and plane flying have ceased to be restricted to human execution. In the context of ground vehicles the main advantages of this autonomy are the enhanced security and better usage of the road; relief of humans from simple tasks like delivering goods and increased mobility for people with disabilities.

Although some tasks can already be performed automatically by a car e.g. autonomous parking and driving on highways [1], complex tasks like fully autonomous driving in urban areas are still a challenge for the existing technology [2].

The main challenge faced by autonomous ground vehicles as for any other kind of autonomous robot is the perception of the surroundings, a task performed by humans easily just by looking at what is around them. By using the eyes, a human is able to perceive the distance of the objects in the surroundings and then classify these objects into obstacles or non-obstacles to navigation.

Based on this, the main goal of this thesis is to propose an approach to identify the possible obstacles for a mobile ground vehicle (and other kinds of ground robots) taking into account limitations in availability of resources present in low-cost systems.

[1] Carey, Nick (2015, May 5th) Nevada gives Daimler nod for open road tests of self-driving truck. *Reuters News*. Retrieved from http://www.reuters.com

[2] Shepardson, David; Woodall, Bernie (2016, July 1st) Tesla crash raises concerns about autonomous vehicle regulation. Retrieved from http://www.reuters.com

1.2 Motivation

Different sensors have been proposed to give a machine the ability to identify the obstacles to navigation. LIDAR and Radar are active sensors which respectively project light or radio waves into the environment. Then by measuring the time it takes the signal to return to the emitter it is possible to know the distance of the measured object and create 3D maps of the surroundings. Although LIDARs are highly accurate at long distances, they are expensive, consume high-power, present measuring errors on object boundaries and have problems detecting black objects and windows [1].

Radars may be classified by the range of distances they can sense. Short-range radars are cheap and have a large field of view but only work for close objects. Mid-range and long-range radars are able to reach long distances at the cost of having a narrow field and increased price [2]. In addition radars are unable to provide any information related to the road shape and are prone to interference as they are active sensors [3].

On the other hand, cameras are cheap, low power and are passive sensors, meaning they do not create interference with any other sensor on the car or in the surroundings. Additionally, it has been shown that cameras are able to detect robustly objects at up to 180m with baselines of 33cm and focal length of 1253 px [4] [5].

Due to the advantages in the use of cameras over LIDAR and radar, computer vision has gained a lot of attention from the research community and industry. Computer vision uses images captured by cameras as a way to extract information from the world. The correct interpretation of the image contents allows the detection and classification of possible obstacles to autonomous navigation.

Different camera set-ups have been explored to try to identify the possible obstacles including: mono-camera, catadioptric systems, stereo-cameras and multi-cameras. Mono-camera systems use only one camera with one sensor and one lens. They use changes in the focal length or sequences of images to measure the distance to the objects [6], [7], [8], [9]. These approaches lack a measure of scale [10], make assumptions on the contents of the scene or require other sensors like radar to recover scale [11], [2]. Catadioptric approaches use mirrors or lenses to capture different views of the objects using only one sensor at the expense of a reduced field of view or lower resolution [12], [13]. Stereo-cameras use two cameras aligned horizontally to emulate human vision, although other set-ups are

possible. Multi-camera approaches use more than two cameras to increase the number of views captured of the scene.

The use of stereo-cameras and multi-cameras is advantageous over mono-camera or catadioptric approaches. The simultaneous capture of different views allows the calculation of distances by triangulation instead of relying on assumptions like static objects or requiring extra sensors. Wide angle lenses allow the capture of images with a large field without a sacrifice in image resolution.

Although the stereo-camera approach presents advantages over mono-camera or catadioptric approaches, it shares the issues related to the use of cameras as sensors. Differences in the manufacturing process result in lens distortions and different responses to the illumination. As the image resolution is increased, the level of detail is also increased resulting in a larger number of pixels which require to be processed.

In this thesis, stereo cameras are chosen as input sensors due to their ability to recover scale by triangulation. Multi-camera set-ups are not used as they would require the processing of a larger number of images increasing the number of required computations.

Robust obstacle detection based on stereo-cameras for outdoor scenarios faces challenges such as occlusions; reflections; shadows and changes in illumination between the cameras. An additional challenge arises when the processing must be performed on-board a ground vehicle. Embedded systems that can used on offline vehicles have a limited amount of resources such as memory and processing power. Additionally, as the cost of the system becomes a limiting factor, these available resources are further decreased.

As the term 'obstacle' is abstract, for this thesis an obstacle is taken as any object lying on the ground that could collide with the vehicle. This definition requires the knowledge of the distance of the objects around the ground vehicle. Therefore this distance is computed using a stereo-camera. Due to the generic definition of obstacle used in this thesis, classification approaches are unsuitable as they would impose assumptions on the nature of the possible obstacles.

The data produced by the stereo-camera sensor has no concept of obstacles or objects, instead it produces stereo-images which are pixels with an associated location and intensity. By using triangulation it is possible to know the distance of the objects as long as the change in location across the stereo-images is known.

The change in location across the stereo-images is found by a process known as stereo-

matching. This process tries to create a map of the changes in location referenced on one of the stereo-images. This map is known as a disparity map.

Three main steps are involved in the stereo-matching process: pixel description, matching cost computation and best match selection. Although the stereo-matching process has been widely studied it still a challenging task. The proper matching of the pixels requires to overcome problems with changes in illumination and is usually a computationally intensive task.

One of the approaches to reduce the computational effort is to use only high level image features which are the important pixels in the stereo-images. As only a few of the pixels are used for stereo-matching, the produced disparity map is sparse. In contrast to a dense disparity map, which contains information about all of the pixels in the stereo-images.

Different kinds of image features have been proposed in the literature: points [14], [15], edges or ridges [16]–[19] and segments [20]. Point features produce disparity maps which are too sparse and it is not always possible to identify the image contents by only looking at the features [21], [22]. Additionally some complex processing has to be done on the image in order to identify these features [14], [15]. Edges and ridges are complementary. Edges are created by changes in colour or texture whereas ridges are the regions delimited by these changes. They are able to represent the image semantics while having compact representation [16]. Edges are easy to represent by chains of pixels, which also provide connectivity information, whereas ridges represent a region. In addition edges may be detected by looking only at changes in the image gradient.

Based on these properties of edges, this thesis proposes their usage for estimating the distance of the objects around the ground vehicle and then using this information to classify the image contents into obstacle or free space.

Thesis Statement: The edges in stereo-images provide enough information to estimate the distance of the objects in the scene and to classify their projection in the stereo-images as obstacle or free space.

1.3 Thesis Contributions

In this thesis four different contributions are made to the field of stereo-matching and obstacle detection. The first two contributions are related to the calculation of the disparity

for the images from a stereo-camera. The third contribution is related to the use of image edges to identify the obstacles in the images.

- **A new pixel descriptor able to identify the edges in one image pass.** This new pixel descriptor incorporates a similarity measure into the Census Transform. This similarity measure acts as a threshold on the image gradient and allows the identification of pixels located around edges in only one pass of the image. This results in a reduction in the search space of a local stereo-matching approach.

- **Evaluate the performance of the Complete Rank Transform in the stereo-matching context.** The Complete Rank Transform has been previously used to compute the optical flow. In this thesis its performance as pixel descriptor is evaluated in a local stereo-matching approach.

- **An approach for extracting 3D curves from stereo-images while simultaneously detecting the edges across the stereo-camera.** This new stereo-matcher extends an edge detector able to produce well-localized, one-pixel wide chains of pixels representing the image edges to run simultaneously across the stereo-images. This results in chains of points in 3D space which represent the edges projected on the stereo-images. Only a few anchor points require to be matched and then the disparity is propagated along the edges. This reduces the required number of computations while still providing accurate results.

- **An approach for labelling the pixels as obstacle or free space based on image edges.** This new approach computes the image stixels by using only the image edges as input data, instead of requiring a cost volume or occupancy grids as other approaches do. The approach is fast to compute and comparable to other stare-of-the-art approaches.

1.4 Thesis Plan

Following from the above, this thesis is organized as follows:

- **Chapter 2** Presents the state-of-the-art and related work in stereo-matching and obstacle detection.

- **Chapter 3** Presents the new pixel descriptor based on the Census Transform and shows its evaluation in a local stereo-matching approach along with the Complete Rank Transform.

- **Chapter 4** Presents the new approach for extracting 3D curves from stereo-images while simultaneously extracting the edges. This chapter also shows its evaluation and compares the results against other state-of-the-art stereo-matching approaches.

- **Chapter 5** Presents the approach for identifying the obstacles in the stereo-images based on edges only. It shows its evaluation and comparison against other state-of-the-art approaches.

- **Chapter 6** Presents the conclusions and future work.

2

Literature Review

2.1 Introduction

This chapter presents the state-of-the-art in depth extraction and obstacle detection. These are two of the most important and challenging tasks performed by any mobile robot as they are crucial for avoiding crashing while moving. Although they have been the focus of much research in recent years, the creation of obstacle maps in real-time using low computational resources is still an open challenge.

Humans are able to perform depth extraction and obstacle detection by using only their eyes. Therefore, one approach to replicate this behaviour in a machine is the use of cameras to perform this task. The analysis of image contents by a machine is known as computer vision and it has been the subject of much research in recent decades. Modelled on the human visual system, it is assumed in this thesis that the cameras used for capturing images are aligned horizontally, their capture sensors lie on the same plane and the cameras are calibrated and the captured images are rectified. It is also assumed that the capture by the stereo-camera is triggered simultaneously obtaining a synchronized pair of images also known as stereo-images. Due to this synchronization the movement in the scene is taken as negligible. These stereo-images are used to extract the distance of the objects from the stereo-camera and to create the obstacle map.

The following sections introduce the different approaches found in the literature for trying to solve this challenging task. First Section 2.2 presents the state-of-the-art in the calculation of the distance of the objects in a scene and then Section 2.3 presents the state-of-the-art in methods to classify the scene into obstacles (non-free space) and free

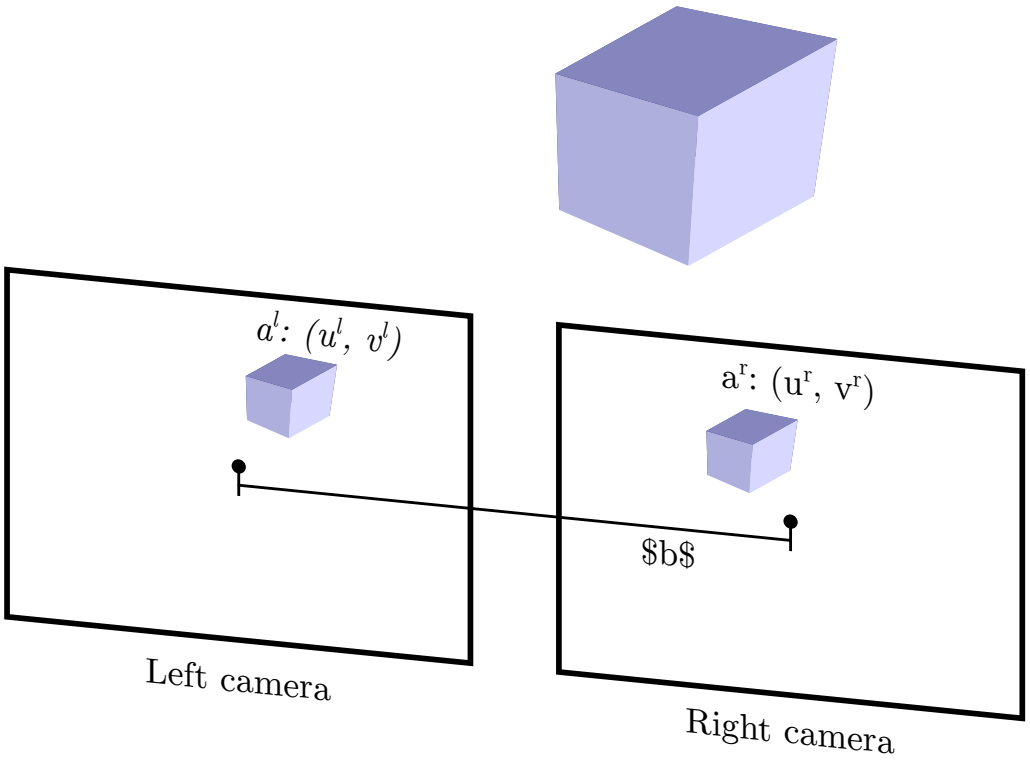


Figure 2-1: Stereo camera view. The image elements a^l and a^r represent the same object in the real world at different locations on the stereo-imagess.

space.

2.2 Depth Extraction

Depth is the difference in the z-coordinates between a 3D point in the space around the robot and the stereo-camera measured in real world coordinates. The cameras in the stereo set-up are separated by a baseline distance b. This separation results in a slightly different angle of view for each camera. Due to this difference in the views of the cameras, the objects captured in the images have different locations in the stereo-images (see Figure 2-1). The identification of the difference in the location of the objects is known as stereo-matching [10] and produces a pairing $m(a^l, a^r)$ where a^l and a^r are image elements in the left and right images respectively.

Once the pairing $m(a^l, a^r)$ is obtained by stereo-matching, triangulation is applied to calculate the distance between the captured objects and the stereo-camera. Triangulation uses the difference in the location of the image elements a^l and a^r along with the information from the stereo-camera geometry and their alignment to estimate the distance of a

point in the 3D world. The analysis of the geometry required to perform triangulation is well-known and a wide number of references are available in the literature. The reader is referred to [10] to review the concepts of camera calibration, rectification and triangulation and how they are used to calculate the distance of the objects by using epipolar geometry.

For rectified stereo-images, disparity d is the difference in the horizontal location between pixels p^l and p^r for the match $m(p^l, p^r)$. This matching uses pixels p^l and p^r as image elements a^l and a^r respectively. Pixels in the left and right image could be represented in image coordinates as $p(u^l, v^l)$ and $p(u^r, v^r)$ respectively, the disparity d may then be expressed as $d = u^l - u^r$. The match $m(p^l, p^r)$ can be expressed in image coordinates as $m(u^l, v^l, d)$ for the left image and $m(u^r, v^r, d)$ for the right image. An image $D(u^l, v^l)$ where the intensity values correspond to the disparity d of the match $m(u^l, v^l, d)$ on the left image is known as a left disparity map. This could be applied similarly for the matches referenced on right image coordinates. For simplicity, in the following sections the term disparity map refers to a disparity map referenced on left image coordinates only, i.e. $D(u, v) = D(u^l, v^l)$.

Disparity maps may be classified as either sparse or dense depending on the number of pixels for which the disparity is obtained. Sparse disparity maps are usually fast to compute but care should be taken when selecting pixels for which disparity is calculated, otherwise they cannot represent the image semantics. On the other hand dense disparity maps contain information for all or most of the pixels in the image but are slower to compute.

Approaches for solving the stereo-matching problem follow a common set of steps in order to perform the matching $m(a^l, a^r)$. These steps are: description, cost calculation and best match selection. Figure 2-2 presents a taxonomy of the methods found in the literature for stereo-matching. In the following sections, each of these steps is analysed and the state-of-the-art is presented. Additionally a set of approaches which take as input a sparse disparity map and produce a dense disparity map is presented due to their relationship to stereo-matching.

2.2.1 Description

In general terms a “description” identifies the attributes or characteristics of something using words. In computer vision a “description” identifies the attributes and characteristics

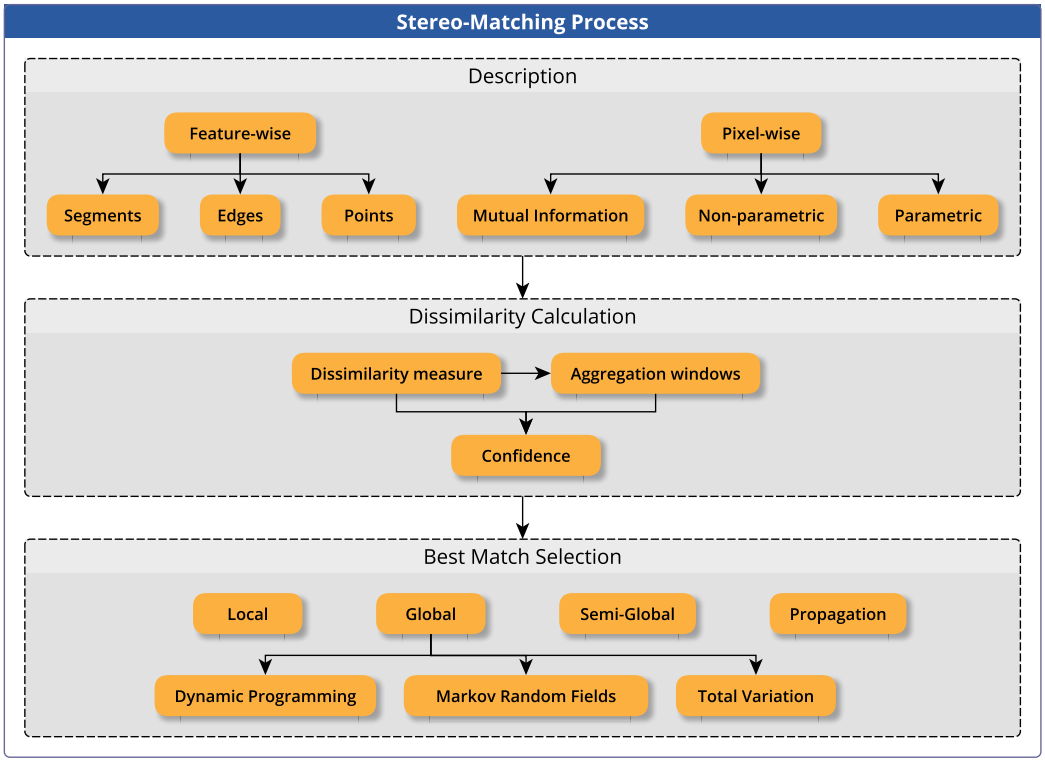


Figure 2-2: Taxonomy of stereo-matching approaches. The description takes as input rectified stereo-images and produces a descriptor for each image component. The matching cost calculation measure how well a pair of descriptors (one on the left and the other on the right image) match, i.e. how well they represent the same image component. The best match selection then takes the computed matching costs and designs matches between the descriptors on the left and right stereo-images. Please refer to the text of Section 2.2.1, Section 2.2.2 and Section 2.2.3 for a detailed description of each step.

of an image element a using a n-dimensional vector. This vector is known as a descriptor and is represented by $\xi(a)$. Characteristics or attributes which may be extracted from an image element a could be colour (intensity for grayscale images), texture and shape among others. For stereo-matching additionally it is desired that the chances that two different image elements a' and a'' produce the same descriptor are minimal. This is known as low repeatability. Although this area has been the subject of a large amount of research, the description of an image pixel with low repeatability is still a challenge.

An image $\mathcal{I}$ captured by a camera is composed of a set of ordered pixels $\mathcal{P}$ with an associated intensity $I(p), p \in \mathcal{P}$; by using abstract representations of $\mathcal{I}$ it is possible to identify elements which may be relevant for specific tasks. These elements are known as features and could be associated with a pixel $p \in \mathcal{P}$ or groupings of pixels. The intensity of a pixel $I(p)$ may be represented by a scalar value on a grayscale image or by a vector

in a colour image. In this thesis grayscale images are used as input due to the smaller resources required for storage and processing in comparison with colour images but the proposed algorithms may be extended to colour images with minimal changes.

Image noise could create problems when the descriptors are created. This noise could be produced by limitations in the sensor in terms of the quality of the sensor, the response to lighting conditions and limitations in the methods used for storing images. Therefore it is common to apply some kind of noise removal prior to the extraction of the descriptors. Although important, the analysis of approaches for noise removal is out of the scope of this thesis. The reader is pointed to [23], [24] for extensive surveys on noise removal techniques. A general but common assumption is that the image noise follows a Gaussian distribution. In this thesis this assumption is made and therefore Gaussian smoothing is applied as a noise reduction filter when specified.

Descriptors can be divided into two categories according to the image elements with which they are associated: pixel-wise or feature-wise. Pixel descriptors can be created for every image pixel by using the image intensities in a neighbourhood around the pixel or by using transformed versions of the image. Feature descriptors are only created for image features by using intensities of the pixels associated to the feature or by using abstract representations of the image. It is expected that an image contains only a few relevant features resulting in a smaller search space for the stereo-matching algorithm.

2.2.1.1 Pixel Descriptors

Pixel descriptors contain information about an image pixel and can be calculated for every image pixel p. This information may be obtained from the intensity $I(p)$ associated with the pixel, from the texture or from some transformed version of the image. Table 2.1 shows the state-of-the-art in pixel descriptors. For grayscale images, it is common to represent the intensity $I(p)$ using positive integers. This allows to take advantage of the ability of computers to perform integer operations. Additionally it is common to reduce the range of the intensities to be in the interval $[0, 255]$ to take advantage of the ability of computers to store data in 8-bit groupings.

Intensity-based descriptors try to use the intensity $I(p)$ of the image pixels to create a unique representation of pixel p. The simplest of intensity-based descriptors $\xi_I(p)$ is the

Information used	Advantages	Limitations
Intensity $\xi_I(p)$ [25]	No additional computation is required.	High repeatability.
Intensity neighbourhood $\xi_{\mathcal{W}}(p)$ [26]–[29]	Low computation required. Increased uniqueness compared to one intensity value only.	Aperture problem. Boundary smoothing.
Normalized Intensity $\xi_N(p)$ [25]	Robust to changes in offset.	Additional computations required to calculate the mean. Requires signed storage.
Laplacian of Gaussian $\xi_{LoG}(p)$ [25], [30]	Fast to compute.	Boundaries smoothing.
Texture $\xi_t(p)$ [31], [32]	Increased uniqueness compared to intensity based approaches.	Additional computation required for calculating the texture. Pixels over the same surface could produce the same texture information.
Rank Transform $\xi_{rt}(p)$ [33]	Robust to changes in illumination. Fast computation.	High repeatability. Aperture problem.
Census Transform $\xi_{ct}(p)$ [33]	Robust to changes in illumination. Low repeatability. Binary representation.	Aperture problem. Large windows tend to be slow to compute.
Complete Rank Transform $\xi_{crt}(p)$ [34]	Robust to changes in illumination. Low repeatability.	Slow to compute. Integer representation.
Modified Census Transform $\xi_{mct}(p)$ [35]	Robust to changes in illumination. Low repeatability.	Slower to compute than the Census Transform.
Ternary Census Transform $\xi_{ct3}(p)$ [36]	Robust to changes in illumination. Robust to noise.	Loss of binary representation.
Ordinal Measures $\xi_{ord}(p)$ [37]	Robust to changes in illumination.	Slow to compute.
Non-Subsampled Contourlet Transform $\xi_{nsct}(p)$ [38]	Shift invariant. Scale invariant.	High repeatability. Slow to compute.
Mutual Information $\xi_{mi}(p)$ [39], [40], [41]	Robust to changes in illumination. Low repeatability.	Slow to compute.

Table 2.1: State-of-the-art on pixel descriptors.

intensity value itself

$$\xi_I(p) = I(p) \tag{2.1}$$

Other approaches try to use intensities from a neighbourhood $N(p)$ around p. Different shapes of neighbourhoods have been proposed e.g. squares [26], [27], stripes [28], irregular shapes [29] and multiple window sizes [26]. Neighbourhood based approaches are susceptible to the aperture problem which occurs when the size of the neighbourhood does not allow the proper identification of the image region. This is illustrated on Figure 2-3.

The intensity neighbourhood descriptor $\xi_{\mathcal{W}}(p)$ is a vector created by concatenating intensity values $I(p)$ for every pixel in the neighbourhood $N(p)$, i.e. :

$$\xi_{\mathcal{W}}(p) = \bigotimes_{p' \in N(p)} I(p) \tag{2.2}$$

where $\bigotimes$ represents the concatenation operator. This descriptor assumes $\xi(p) = \xi(p')$ if p

(a) Left image. (b) Right image.

Figure 2-3: Illustration of the aperture problem. The green rectangle is the search neighbourhood on the left image. It is not possible to identify which of the two rectangles on the right image corresponds to the one in the left image by using only pixels contained in the neighbourhood defined by each rectangle.

located in the left image and p' located on the right image correspond to the same pixel. This assumption is widely used in stereo-matching [42].

The normalized intensity descriptor $\xi_N(p)$ extends the intensity neighbourhood descriptor $\xi_{\mathcal{W}}(p)$ by subtracting the mean value calculated over the neighbourhood $N(p)$ from each element in the vector [25], i.e. :

$$\xi_N(p) = \bigotimes_{p' \in N(p)} I(p) - \bar{I}(p) \tag{2.3}$$

where $\bar{I}(p)$ is the mean intensity calculated for the neighbourhood $N(p)$. This normalization adds robustness to changes in offset in brightness of the images, i.e. if one of the images in the stereo-images is brighter than the other, at the price of computing the mean for each neighbourhood $N(p)$.

Laplacian of Gaussian (LoG) descriptors $\xi_{LoG}(p)$ convolve the intensity image with a Laplacian of Gaussian bandpass filter in order to remove noise and offset in intensities and then create a vector by concatenating the resulting image over a neighbourhood $N(p)$. The drawback of this approach is that it tends to blur the object boundaries introducing errors [25], [30].

Texture-based descriptors try to describe a pixel-based on texture information $\xi_t(p)$ from a region around pixel p. Similar to intensity-based descriptors, texture-based descriptors define a neighbourhood around pixel p and then extract the corresponding texture information in a manner that is robust to affine transforms [31], [32]. The main drawback of these approaches is the additional computation required to calculate the texture for each pixel. Additionally, pixels which are located on the same surface may share the same texture leading to wrong matches.

Transform-based descriptors translate the intensity image captured by a camera into an alternate domain with the purpose of increasing the uniqueness of the representation $\xi(p)$ for each pixel p. Although different image transforms are available in the literature only a few of them are suitable for solving the stereo-matching problem. These are the Census and Rank Transforms [33] and their variants; the Controurlet Transform [43]; and the Non-Subsampled Contourlet Transform [38]. Other image transforms focus on the extraction of geometry or identifying patterns and therefore they are unsuitable for solving the stereo-matching problem.

The Rank Transform (RT) and Census Transform (CT) proposed by Zabih and Woodfill [33] are image transforms which assume that the local ordering of the intensity is kept constant across the stereo-images. This dependence on the ordering of intensities and not their magnitudes provides an inherent robustness to changes in illumination [25].

The Rank Transform $\xi_{rt}(p)$ defines the rank $r(p)$ of pixel p as the number of pixels with an intensity smaller than the intensity of p in a rectangular neighbourhood $N(p)$ of size $m \times n$, i.e. :

$$r(p) = \sum_{q \in N(p)} T(p, q) \tag{2.4}$$

where $T(p, q)$ is defined as:

$$T(p, q) = \begin{cases} 1 & \text{if } I(q) < I(p) \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

Then the Rank Transform is defined as the rank $r(p)$ of the centre pixel p_c in the neighbourhood $N(p)$, i.e. :

$$\xi_{rt} = r(p_c) \tag{2.6}$$

The main drawback of the Rank Transform is that the range of the rank $r(p)$ is limited to the number of pixels in the neighbourhood $N(p)$.

The Census Transform $\xi_{ct}(p)$ is calculated by concatenating the result of comparing each item in the neighbourhood against the centre pixel, i.e. :

$$\xi_{ct}(p) = \bigotimes_{q \in N(p)} T(p, q) \tag{2.7}$$

where $\bigotimes$ represents the concatenation operator and $T(p, q)$ is defined as in Equation (2.5). This image transform adds robustness to changes in illumination and by producing a binary string it takes advantage of the ability of computers to store and process binary data.

Due to the inherent robustness to changes in illumination of the Rank and Census Transforms different enhancements have been proposed in the literature. Notably the Complete Rank Transform and the Ternary Census Transform have been used to accurately calculate optical flow [34], [36] and face recognition [35].

The Complete Rank Transform $\xi_{crt}(p)$ extends the Rank Transform by concatenating the rank value $r(p)$ for every pixel in the neighbourhood $N(p)$ [34], i.e. :

$$\xi_{crt}(p) = \bigotimes_{p' \in N(p)} r(p') \tag{2.8}$$

where $\bigotimes$ is the concatenation operator. Although this descriptor carries more information than the Rank and Census Transforms and produces more unique values, the calculation of the rank for every pixel in the neighbourhood $N(p)$ adds a considerable number of comparisons and add operations resulting in a slow computation when compared with the Rank and Census Transforms.

The Modified Census Transform $\xi_{mct}(p)$ replaces the comparison against the value of the centre pixel in the Census Transform with a comparison against the mean intensity in the neighbourhood $N(p)$ [35]. By doing this a value is also obtained for the centre pixel increasing the uniqueness of the Census string at the cost of computing the mean for the neighbourhood $N(p)$.

The Ternary Census Transform $\xi_{ct3}(p)$ incorporates a similarity measure ε of the pixels in the neighbourhood $N(p)$ to the centre pixel [36], i.e. :

$$T_{ct3}(p, p') = \begin{cases} 0 & I(p) - I(p') > \varepsilon \\ 1 & |I(p) - I(p')| \leq \varepsilon \\ 2 & I(p') - I(p) > \varepsilon \end{cases} \tag{2.9}$$

although this incorporation of a similarity measure increases the robustness against noise and small changes in intensities, the advantage of a binary representation is lost. This similarity concept is used in this thesis to present a new variation of the Census Transform

in Section 3.2.1.

Other variants of the Census Transform which aim to be hardware friendly, to reduce the number of computations and to incorporate colour information are found in [44], [45], [46], [47], [48]. Although they have been shown to produce good results for stereo-matching, they are all based on the same principle the ordering of the intensities to represent each pixel. Therefore they are not described in detail in this thesis.

Descriptors based on the ordinal measure $\xi_{ord}(p)$ of the pixels in the neighbourhood $N(p)$ have been proposed by Bhat and Nayar obtaining good results for the stereo-matching problem [37]. Although robust to changes in gain and offset, these measures are slow to compute and cannot be implemented as a filter, limiting their application in real-time systems [25].

Descriptors $\xi_{nsct}(p)$ based on the Non-Subsampled Contourlet Transform (NSCT) use the coefficients of the NSCT at each scale of a non-subsampled pyramid [38]. This image transform uses non-subsampled pyramids and non-subsampled directional filter banks to provide a sparse representation of the input image. Based on the results presented in [38] the NSCT requires more experimentation in order to obtain confident and unique descriptors. This experimentation is out of the scope of this thesis. Therefore no further analysis is performed.

Mutual Information based descriptors $\xi_{mi}(p)$ have been used for incorporating robustness to illumination changes [39], [40], [41]. Although effective, the required computations limit their applicability in real-time systems [41].

In this thesis the Census Transform is chosen due to its inherent robustness to changes in illumination and the low resources required for its computation. Then an extension to the Census Transform is proposed in Section 3.2.1 with the purpose of allowing the identification of important features in the image without incurring extra computational efforts and keeping its binary representation.

2.2.1.2 Feature Descriptors

Feature descriptors contain information about an image feature which is defined as a pixel or group of pixels important for a certain task, in the context of stereo-matching. It is important that these features are easy to identify and robust under the view changes produced by separation of the cameras. It is also desired that the number of features

obtained in an image is low, reducing therefore the number of candidates which must be tested for a match in an image pair. The low number of image features in stereo-images results in disparity maps which contain information only for a few of the image pixels. Such disparity maps are known as sparse. Sparse disparity maps allow a reduction in the required resources to store and process them.

Features can be classified according to the information they represent as points, edges or ridges and segments. Some of them have been found to be robust to view-changes and have been successfully used for stereo-matching [16], [49], [50]. It is important to carefully select the features used to compute the disparity maps, as they would impact the amount of information kept about the scene.

Point features are obtained by identifying important pixels in the image. Then neighbourhoods are used to extract information around the important pixel. Although they have proven to be robust against view changes, their sparse nature makes them unsuitable for obstacle detection. As has been shown in [21], [22], in many scenarios points are not enough for representing the image content and for detecting obstacles. Based on this fact, point features are not included the scope of this thesis. For surveys on feature point descriptors please refer to [14], [15].

Edge features are obtained by using information from image edges. The main advantage of this kind of feature is its ability to represent the image semantics as edges are present on every object boundary [16]. Although they have been used for stereo-matching, their efficient use is still a challenging task, due to the wide variety of scenarios in outdoor environments. Approaches to describe edges can be divided according to the shape of the edge into straight lines and curves.

By assuming the edges are straight lines, a geometric model could be assumed [17]–[19] or a fixed shape neighbourhood may be created [51]. Although these approaches have proven to be successful in some scenarios, objects like pedestrians, trees and bicycles do not present many straight lines when captured in an image. This makes the use of straight lines unsuitable for generic obstacle detection.

Curves can be used to represent any object in an image at the price of increased complexity compared to straight lines. Table 2.2 presents a summary of the state-of-the-art in curve descriptors. In order to represent curves in vector form, the approaches for creating the descriptors can be divided according to the information they use into: edge

Information used	Advantages	Limitations
Edge points [52]	Fast to compute.	Sensitive to feature point detector. Sensitive to edge detector and occlusion. No connectivity information is used.
Intensity regions. [17], [53], [54]	Fast to compute. Similar to window-based matching but with reduced search space.	No connectivity information is used. Assume brightness constancy.
Orientation, difference in brightness at both sides of the curve, reprojection [22], [55]	Robust to view angle changes	No connectivity information is used. Reprojection could be slow to compute.
Orientation Histograms [49]	Robust to wide-baselines.	Slow to compute. High repeatability. High number of unmatched edges.
Curve shape [20], [56], [57]	Robust to affine transforms.	Slow to compute. Objects with the same shape would produce the same descriptor.

Table 2.2: State-of-the-art in curve descriptors.

points; underlying models; and the shape of the curve.

Edge-point-based approaches select an edge-point and use only this point for performing the matching of the edge assuming there is a direct relationship between the disparity of this edge point and the remaining points in the edge. Edge-point-based approaches use either feature points detected along the edge (e.g. corners or end-points) or each of the points independently.

Wei Wei and King Ngi Ngan [52] proposed an edge-point approach where corner points are described and matched as feature points. Using this matching the disparity is propagated over the edge pixels. Although fast, this approach assumes the disparity along the edges is constant. It may not be the case for lines, which are produced by objects on the ground plane e.g. lane markers or long vehicles.

Rao, Chung, and Hutchinson [17] proposed an edge matching approach in which the end points are described by using intensity windows and then a Bezier curve is fitted to the edge. Although effective, edge end-points are unstable under view changes and Bezier curves are slow to compute.

Baker and Binford [55] proposed to match edge-points independently by using information about angle, side, intensities, and contrast to create the descriptor. Then Dynamic Programming is applied to calculate the disparity values of the edge pixels. Although effective, the assumption of constant brightness may not be met in real world scenarios.

Tomono [53] proposed to describe edge-points using normalized intensity windows around the edge points and the gradient orientation. Then Dynamic Programming was used to perform the matching similar to [55]. Although Tomono obtained good results for

indoor environments, the matching had issues on low textured areas and an improvement was suggested as future work.

Witt and Weltin [28], [54] proposed an approach in which edge points are described by vertical or horizontal strips of intensity values created at each side of the edge. Although, this approach proved to be fast and robust on computer generated datasets, it is common to find changes in illumination in real world scenarios. In addition no edge connectivity information is used.

Meltzer and Soatto [49] proposed a bio-inspired edge descriptor based on Histograms of Oriented Gradients (HOG) for different scales. They proposed to identify anchor points along the edges by defining circular regions of radius r and selecting those points for which the integral of the Laplacian over the region produced an extremum. Then the descriptor is created by calculating the HOGs for all of the points in a circle of radius r centred on the anchor point. The orientations are weighted by edge strength and a Gaussian centred at the anchor points. Although it produced good results when compared to other bio-inspired methods, it presented problems near depth discontinuities and the calculation of histograms limited its application in real-time systems.

Fabbri and Kimia [22] used tangent attributes on curvelets to describe the image edges. During the matching stage, 3D reprojection information is used additionally to restrict the search space and ensure consistency on the reconstructed objects. Although this approach is effective, the authors do not provide information about running time and it would be expected that the use of 3D information limits its application in real-time and embedded systems.

Mokhtarian and Abbasi [56] used the shape of Jordan Curves to perform matching on a wide baseline stereo-camera. They first define the shape of a Jordan Curve and then define a distance measure for the shapes. Then the curves with the smallest distance are taken as a pair. Although this approach is robust to affine transforms, the presence of similar objects resulted in an erroneous matching as they may produce the same descriptor. Additionally, the approach is sensitive to the linking procedure used.

Mai and Hung [57] used Curvature Scale Space (CSS) to identify affine invariant points and segments in the curves. Then they are aligned by the Smith-Waterman algorithm producing an initial match. This match is used to calculate the affine transform of the curve pair, then the similarity of two curves is calculated by transforming the curve pair

using the estimated affine transformation. Although this approach is robust to partial occlusions, the computation of the CSS is computationally expensive and the estimation of the affine transformations limits its application in embedded systems.

Segment-based features require the identification of groups of pixels with common attributes (e.g. same colour, delimited by a gradient, blobs, etc.) known as segments. Then for each segment a constant disparity is assumed, and the pixel descriptors are created by using approaches similar to the pixel descriptors presented in Section 2.2.1.1 but taking as neighbourhood $N(s)$ all of the pixels in the segment s [20]. Although this kind of approach reduces the number of pixels matched, the segmentation process is computationally expensive [58]. Therefore they are not suitable for real-time processing.

2.2.2 Dissimilarity Calculation

Once the descriptors have been created, the dissimilarity $\delta(\xi(a_i^l), \xi(a_j^r))$ of a pair of descriptors $\xi(a_i^l)$ on the left image and descriptor $\xi(a_j^r)$ on the right image must be calculated. The selection of the dissimilarity measure is defined in a way that if $\delta(\xi(a_i^l), \xi(a_j^r)) = 0$ the descriptors $\xi(a_i^l)$ and $\xi(a_j^r)$ represent the same image element across the stereo-images. Hirschmuller and Scharstein [25] classify the dissimilarity measures into parametric and non-parametric. This taxonomy is used in this thesis.

The selection of the proper similarity measure faces several challenges, one of the most important is the presence of changes in illumination on the stereo-images. Some common changes are bias and vignetting.

A bias indicates that one of the the stereo-images has a higher brightness than the other. This effect occurs when the cameras present a different response to the illumination of the captured scene. Figure 2-4 shows an example of this effect.

Vignetting occurs due to limitations on the lens that result in images where the centre is brighter than the rest of the image. Figure 2-5 shows an example of the vignetting effect.

As seen in [25], different approaches have been proposed to eliminate these effects but some of them blur the images. The following text analyses different dissimilarity measures and shows how some of these measures take into account changes in illumination.

Figure 2-4: Example of the bias effect across the captured stereo-images. The left image is brighter than the image of the right even though they were taken at the same time. This effect happens due to differences in the response to the illumination of the sensors in the cameras. The shown images are part of the Middlebury v3 dataset [59].

Figure 2-5: Example of the vignetting effect. The image on the left side does not have vignetting effect. The image on the right side shows a light vignetting effect. It can be seen that the pixels in the middle are brighter than the rest for the right image. For visualization purposes the image with the vignetting effect has been manually edited to enhance the effect.

2.2.2.1 Parametric Dissimilarity Measures

Parametric dissimilarity measures assume the the brightness of the objects surface in the image is the same regardless of the location of the cameras. Objects whose surfaces present this behaviour are known as lambertian surfaces. These measures use the intensity $I(p(u^l, v^l)) \in \mathcal{I}$ and $I(p(u^r - d, v^r)) \in \mathcal{I}$ of pixels $p(u^l, v^l)$ and $p(u^r - d, v^r)$ in the left and right image respectively where d is the disparity of the pixel pair. In order to ease the reading, the following notation is used through the rest of this section $p^l = p(u^l, v^l)$ and $p^r - d = p(u^r - d, v^r)$.

Although different types parametric dissimilarity measures have been proposed in the literature, the use of the intensity magnitude results in sensitivity to changes in illumination. Therefore only the most representative similarity measures or those which incor-

Measure	Advantages	Limitations
AD [25]	No additional computation required on the input image.	Brightness constancy assumption. Aperture problem. High repeatability.
SSD [25]	Increased discrimination compared to AD.	Brightness constancy assumption. Aperture problem.
SAD [25]	Increased discrimination compared to AD. Fast to compute. Only additions are required.	Brightness constancy assumption. Aperture problem.
ZSAD [25]	Robustness to changes in offset.	Slower to compute than SAD. Aperture problem.
NCC [60], [61]	Robustness to changes in gain.	Blurs depth discontinuities. Slow to compute.
ZNCC [25]	Robust to changes in gain and offset.	Slow to compute.
BT [62]	Fast to compute.	Brightness constancy assumption.

Table 2.3: Parametric dissimilarity measures.

porate robustness to changes in illumination are reviewed in this thesis. For an extensive evaluation of dissimilarity measures please refer to [25], [63], [64], [65], [30].

Absolute Difference (AD) is commonly used as a dissimilarity indicator due to its simplicity. It uses the intensity descriptor $\xi_I(p)$ (see Equation (2.1)) for its calculation. This dissimilarity measure assumes brightness constancy across the image pair and could be used as a baseline for performance measurement [25]. AD dissimilarity $\delta_{AD}(p,d)$ is defined as:

$$\begin{aligned} \delta_{AD}(p,d) &= |\xi_I(p^l) - \xi_I(p^r - d)| \\ &= |I(p^l) - I(p^r - d)| \end{aligned} \tag{2.10}$$

where $I^l(p)$ is the intensity of pixel p in the left image and $I^r(p-d)$ is the intensity of the pixel corresponding to p at disparity d in the right image.

Another pair of commonly used similarity measures are Sum of Squared Differences (SSD) and Sum of Absolute Differences (SAD) . These dissimilarity measures use the intensity neighbourhood descriptor $\xi_{\mathcal{W}}(p)$ (see Equation (2.2)) and uses each of the descriptor components for its calculation. For clarity, the descriptor $\xi_{\mathcal{W}}(p)$ is replaced with its intensity components in the following expressions. Mathematically SSD dissimilarity $\delta_{SSD}(p,d)$ is defined as:

$$\delta_{SSD}(p,d) = \sum_{q \in N(p)} (I^l(q) - I^r(q-d))^2 \tag{2.11}$$

SAD dissimilarity $\delta_{SAD}(p,d)$ is commonly used in embedded systems as it does not

require multiplication and the required range of values is smaller compared to SSD. It is defined as:

$$\delta_{SAD}(p,d) = \sum_{q \in N(p)} |I^l(q) - I^r(q-d)| \tag{2.12}$$

AD, SSD and SAD are sensitive to changes in illumination as demonstrated in [25].

Only a few parametric dissimilarity measures take into account illumination changes. Zero-mean Sum of Absolute Differences (ZSAD) subtracts the mean intensity of the neighbourhood $N(p)$ in order to take into account changes in offset [25]. ZSAD dissimilarity $\delta_{ZSAD}(p,d)$ is defined as:

$$\delta_{ZSAD}(p,d) = \sum_{q \in N(p)} |I^l(q) - \bar{I}^l(p) - I^r(q-d) + \bar{I}^r(p-d)| \tag{2.13}$$

$$\bar{I}(p) = \frac{1}{|N(p)|} \sum_{q \in N(p)} I(q). \tag{2.14}$$

Normalized Cross Correlation (NCC) compensates for gain changes and is optimal for dealing with Gaussian noise [60], [61] but tends to blur depth discontinuities [25]. It is defined as:

$$\delta_{NCC}(p,d) = \frac{\sum_{q \in N(p)} I^l(q) I^r(q-d)}{\sqrt{\sum_{q \in N(p)} I^l(q)^2 \sum_{q \in N(p)} I^r(q-d)^2}} \tag{2.15}$$

Zero-mean Normalized Cross Correlation (ZNCC) is the only parametric dissimilarity measure which takes into account changes in offset and gain within the neighbourhood $N(p)$ [25]. It is defined as:

$$\delta_{ZNCC}(p,d) = \frac{\sum_{q \in N(p)} (I^l(q) - \bar{I}^l(p))(I^r(q-d) - \bar{I}^r(p-d))}{\sqrt{\sum_{q \in N(p)} (I^l(q) - \bar{I}^l(p))^2 \sum_{q \in N(p)} (I^r(q-d) - \bar{I}^r(p-d))^2}} \tag{2.16}$$

Birchfield and Tomasi [62] proposed a dissimilarity measure insensitive to image sampling BT . This approach measures how well a linear interpolated value obtained on the

right image fits with the corresponding value on the left image. It is defined as:

$$\delta_{BT}(p,d) = \min(A,B) \tag{2.17}$$

$$A = max\{0, I^l(p) - I^r_{max}(p-d), I^r_{min}(p-d) - I^l(p)\} \tag{2.18}$$

$$B = max\{0, I^r(p-d) - I^l_{max}(p), I^l_{min}(p)0I^r(p-d)\} \tag{2.19}$$

$$I_{min}(p) = min(I_-(p), I(p), I_+(p)) \tag{2.20}$$

$$I_{max}(p) = max(I_-(p), I(p), I_+(p)) \tag{2.21}$$

$$I_-(p) = (I(p) + I(p-1))/2 \tag{2.22}$$

$$I_+(p) = (I(p) + I(p+1))/2 \tag{2.23}$$

$$\tag{2.24}$$

Alternatively, any of the descriptors presented in 2.2.1.1 which produces non-binary data could be combined with the presented dissimilarity measures to increase robustness to changes in illumination or computing speed. Table 2.3 shows a summary of the available parametric dissimilarity measures.

2.2.2.2 Non-parametric Dissimilarity Measures

Non-parametric dissimilarity measures use only the local ordering of intensities in the neighbourhood $N(p)$. Therefore they are robust against illumination changes [25]. This category of dissimilarity measures uses the Rank Transform, Census Transform or Ordinal Measures presented in Section 2.2.1.1 as input data.

For the Rank Transform [33], Ordinal Measure from Bhat and Nayar [37] and the Complete Rank Transform [34] which are real valued, SAD (Equation (2.12))and SSD (Equation (2.11)) are commonly used for calculating the dissimilarity. The intensity values are replaced with the corresponding transformed data.

For the Census Transform and its variations, the Hamming distance is the standard selection. The Hamming distance $H(\xi^l(p), \xi^r(p-d))$ is defined as the number of different bits in the binary strings of $\xi^l(p)$ and $\xi^r(p-d)$ from the left and right image respectively, i.e. the number of elements which are zero in one string and one in the other. This dissimilarity measure can be efficiently implemented in embedded systems by the use of LUT tables or using bit-set counting functions when available [48], [25].

(a) Left image. (b) Right image.

Figure 2-6: Example of the search range for a local approach. The green square on the left image represents the search pattern, the green rectangle on the right image represents the search area.

2.2.3 Best Match Selection

The calculation of sparse and dense disparity maps requires the selection of the best match for the image elements across the stereo-images. In order to represent how good or bad a match is, a cost function $c(p, p-d)$ is defined where p is a pixel or feature on the left image and $p-d$ is the corresponding pixel on the right image at disparity d. This thesis assumes the stereo-images are rectified, this common assumption reduces the search space to an horizontal line decreasing the required number of computations. Approaches for best match selection can be classified, based on the the type of information they use, as: local, global or semi-global.

2.2.3.1 Local Approaches

Local approaches use only information from a neighbourhood around the matching image element and perform a cost calculation against the elements in the search area. Figure 2-6 shows an example of the search region in an image pair. Local approaches define a cost function $c(a, a-d)$ where a is an image element on the left image and $a-d$ is an image element on the right image with disparity d between d_{min} and d_{max}. For the remainder of this section the image element a is assumed to be a pixel p, then the cost function is defined as $c(p, p-d)$ but the search procedure and metrics could be applied to any kind of image element a.

The cost volume $V_c(x, y, z)$ is a common representation of the cost for every image pixel at disparities between d_{min} and d_{max}. The cost volume is referenced to one of the images in the stereo-images. For example, for a cost volume referenced to the left image,

the x and y coordinates correspond to the u and v coordinates in the left image and the z correspond to the disparity. The cost value $V(x, y, z)$ at coordinate (x, y, z) corresponds to the cost of matching pixel (x, y) on the left image with pixel $(x - d, y)$ on the right image.

The best match is selected using a Winner Takes All (WTA) strategy. This means the best match is taken as the element with disparity d which produces the minimum on the cost function $c(p, p - d)$.

$$\underset{d \in [d_{min}, d_{max}]}{\arg\min} \quad c(p, p - d) \tag{2.25}$$

Aggregation windows $A(p)$ also known as support regions, which allow an increase of the uniqueness in descriptors by assuming the disparity is constant over the window of size m by n [45], [66]. The aggregation cost is defined as:

$$c_A(p, p - d) = \sum_{q \in A(p, p-d)} c(q, q - d) \tag{2.26}$$

then the cost $c_A(p, p - d)$ is used as cost function in the optimization problem. For simplicity, in the remainder of the thesis the cost function $c(p, p - d)$ will be used irrespective of the use of aggregation windows.

The cost function $c(p, p - d)$ used in local approaches corresponds to one or more of the dissimilarity measures presented in Section 2.2.2:

$$c(p, p - d) = \delta(p, p - d), \tag{2.27}$$

additionally, it is common to use a confidence value $\beta(p, p - d)$ to remove spurious matches by thresholding. Techniques for the calculation of the confidence of the match in local approaches use information from the search range to asses the quality of the match. An extensive evaluation of confidence measures is presented in [64], [63], [65], [67], [30].

Left-Right Consistency Check (LRC) has proven to be one of the most effective methods for identifying spurious matches [63] and could be used along with other confidence measures [65]. LRC takes as input two disparity maps, the first referenced on the left image, the second referenced on the right image. Then the disparities obtained for a given pixel are tested in both disparity maps and if the difference in the disparity exceeds a threshold t_{LRC} the pixel is marked as non-consistent.

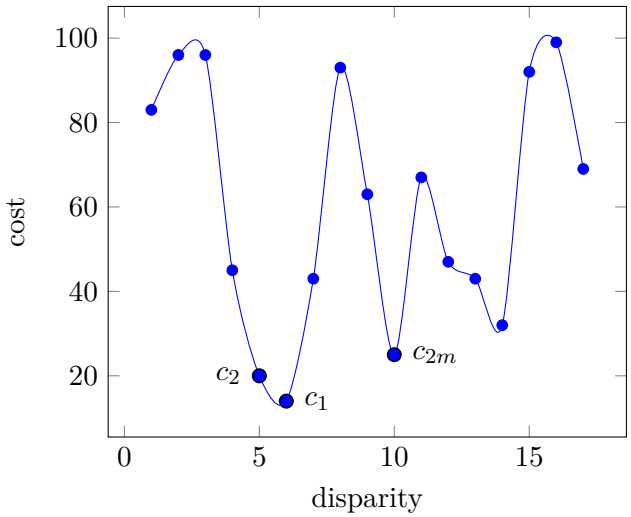


Figure 2-7: Costs used for the calculation of confidence in Table 2.4.

A summary of confidence measures is presented in Table 2.4. In this table c_1 is located at disparity d_1 and is the minimum cost found in the search space

$$c_1 = c(p, d_1) = \min_{d \in [d_{min}, d_{max}]} c(p, d) \tag{2.28}$$

c_2 corresponds to the second minimum at disparity d_2, i.e. the minimum value for which $c_2 > c_1$; and c_{2m} corresponds to the second local minimum at disparity d_{2m}, i.e. a minimum value that is not a direct neighbour of c_1. This is illustrated in Figure 2-7.

2.2.3.2 Global Approaches

Global approaches for disparity calculation use information from the whole image to define an energy function $E : S \rightarrow \mathbb{R}$ which maps a set of candidate solutions S to a measure of the quality of the solution. This energy function E represents the cost of assigning a disparity value to each image pixel. Most of the energy functions found in the literature have the form

$$E(\mathbf{d}) = E_{\text{data}}(\mathbf{d}) + E_{\text{prior}}(\mathbf{d}) \tag{2.29}$$

where $\mathbf{d}$ represents the disparity values assigned to the image pixels; $E_{\text{data}}(\mathbf{d})$ penalizes solutions which are inconsistent to the data and $E_{\text{prior}}(\mathbf{d})$ imposes a prior [74].

By using this kind of energy function, global approaches apply optimization techniques in order to identify a solution $\mathbf{d}^*$ which minimizes the energy $E(\mathbf{d})$, i.e. $\mathbf{d}^* = \arg\min_{\mathbf{d} \in S} E(\mathbf{d})$. In order to find the solution $\mathbf{d}^*$ two approaches have been published in the literature: the

Name	Formula
Matching Score Metric (MSM) [64]	$\beta_{MSM} = -c_1$
Curvature (CUR) [68]	$\beta_{CUR} = -2c(p,d_1) + c(p,d_1-1) + c(p,d_1+1)$
Peak Ratio (PKR) [64]	$\beta_{PKR} = c_{2m}/c_1$
Peak Ratio Naive (PKRN) [64]	$\beta_{PKRN} = \frac{c_2+\epsilon}{c_1+\epsilon}$
Nonlinear Margin (NLM) [64]	$\beta_{NLM} = e^{\frac{c_2-c_2}{2\sigma^2}}$
Probabilistic Metric (PRB) [64]	$\beta_{PRB} = \frac{NCC(d_1)}{\sum_d NCC(d)}$
Maximum Likelihood Metric (MLM) [64], [65]	$\beta_{MLM} = \frac{e^{-c_1/2\sigma^2}}{\sum_d e^{-c(p,d)/2\sigma}}$
Attainable Maximum Likelihood (AML) [64]	$\beta_{AML} = \frac{1}{\sum_d e^{\frac{c(p,d)-c_1}{2\sigma^2}}}$
Negative Entropy Metric (NEM) [69]	$p(d) = \frac{e^{-c(p,d)}}{\sum_{d'} e^{-c(p,d')}}$ $\beta_{NEM} = -\sum_d p(d)\log p(d)$
Winner Margin (WMN) [69]	$\beta_{WMN} = \frac{c_{2m}-c_1}{\sum_d c(p,d)}$
Winner Margin Naive (WMNN) [64]	$\beta_{WMNN} = \frac{c_2-c_1}{\sum_d c(p,d)}$
Left-Right Consistency Check (LRC) [68]	$\beta_{LRC} = \lvert d_1 - D_R(x-d_1,y)\rvert$ where $D_R(x,y)$ is the disparity assigned to the right image at the coordinate (x,y) when performing a right-left matching instead of a left-right.
Left-Right Difference (LRD) [64]	$\beta_{LRD} = \frac{c_2-c_1}{\lvert c_1 - \min\{c_R(q,d_R)\}\rvert}$ where $c_R(q,d)$ is the cost resulting from matching pixel q in the right image to a pixel with disparity d in the left image (right-left direction).
Perturbation (PER) [70]	$\beta_{PER} = \sum_{d!=d_1} e^{-\frac{(c(p,d_1)-c(p,d))^2}{\epsilon^2}}$
Modified Perturbation (MPER) [70]	$\beta_{MPER} = \sum_{d\in[d_{min},d_1-n]\cup[d_1+n,d_{max}]} e^{-\frac{c(p,d_1)-c(p,d)}{\sigma^2}}$ where n elements are excluded from the confidence computation. The authors suggest to set n to half of the window size used for obtaining the descriptor.
Self-Aware Matching Measure (SAMM) [71]	$\beta_{SAMM} = \frac{\sum_d (c(p,d-d_1)-\mu_{LR})(c_{LL}(p,d)-\mu_{LL})}{\sigma_{LR}\sigma_{LL}}$ where μ_{LR} and σ_{LR} are the mean and standard deviation of the cost function when matching on the left-right direction and $\mu_L L$ and $\sigma_L L$ are defined similarly but for matching the left image with itself.
Distinctive Similarity Measure (DSM) [72]	$\beta_{DSM}(x,y) = \frac{C_{L\_DTS}(x,y) x C_{R\_DTS}(x-d,y)}{c_1^2}$ where C_{L_DTS} and C_{R_DTS} are the distinctiveness maps of the left and right images.
Local Curve [65]	$\beta_{LC} = \frac{\max c(p,d-1), c(p,d+1) - c_1}{\epsilon}$
A-contrario Methodology [73]	This approach instead of define a confidence measure, proposes a method to identify the probability that a disparity value is assigned by chance and then uses Principal Component Analysis to build an a-contrario model.

Table 2.4: Confidence measures for local stereo-matching approaches. Please refer to Figure 2-7 and the text for information on the used cost minimum values.

first formulates the energy minimization in a discrete domain. This formulation corresponds to a labelling problem where the labels correspond to the disparity value of the matched pixels. The second approach formulates the energy minimization in a continuous domain. This kind of formulation is usually optimized by a Total Variation (TV) framework.

Approaches which model the minimization of the energy E as a labelling problem model use an energy function of the form:

$$E_{\text{labeling}}(d) = \sum_{p \in P} \delta(p, d_p) + \sum_{i,j \in N(p)} V_{i,j}(d_i, d_j) \tag{2.30}$$

where P is the set of pixels in the image; $\delta(p, d_p)$ is one of the dissimilarity measures presented in Section 2.2.2 and corresponds to the data term from Equation (2.29); and $V_{i,j}(d_i, d_j)$ is the cost of assigning the labels d_i and d_j to two adjacent pixels i and j in the neighbourhood $N(p)$ and corresponds to the prior term in Equation (2.29). It is common to find the Absolute Difference (Equation (2.10)) as the dissimilarity function due to its fast computation.

Markov Random Fields (MRF) models have been widely used for solving labelling problems [75], [76]. Although the energy minimization by MRF models is an NP-hard problem, approaches like Belief Propagation (BP), Graph Cuts (GC) and Dynamic Programming (DP) have proven to provide good approximations [74], [75].

Felzenszwalb and Huttenlocher [75] presented a multi-scale formulation of BP which requires a constant small number of iterations. Although this approach reduces the number of required computations, it processes the images in the Middlebury Stereo dataset [77] in around one second per image, which limits its application in real-time environments.

As an improvement to the work in [75], Sarkis and Diepold [78] proposed the use of sparse BP for increasing the performance. This reduced the running time to 75% of that obtained in [75] at the cost of an increase of 5% in the error. Additionally Trinh [20] presented a segmentation approach based on the work of [75]. This work [20] was able to reduce the computation time to 73% of the work in [75]. Although faster, the application of these approaches in real-time environments is limited.

Graph Cuts (GC) for energy minimization in stereo-matching were proposed first by Boykov, Veksler, and Zabih [79] and obtained accurate disparity maps for the Middlebury

Stereo dataset [77] in around 35 seconds. The running time for this GC based optimization was improved in [80] by incorporating an update to the label cost at each iteration of the algorithm proposed by [79]. Although it was not used for stereo, Delong, Osokin, Isack, *et al.* [80] used GC to improve the running time for segmentation, homography calculation and motion estimation. In [81], the method proposed in [80] was compared to SGM obtaining similar running times and accuracy by adding scene and temporary priors to the algorithm proposed in [80]. Although accurate, the available implementations still have computation times in the order of seconds which limits its application to real-time systems.

Dynamic Programming (DP) is an efficient optimization technique which has been successfully used for curve detection, contour completion, stereo-matching and deformable object matching [74]. In the context of stereo-matching the optimization was originally usually performed for each scan-line independently [55]. Then Ohta and Kanade [82] proposed a method for incorporating information from multiple scan-lines in the optimization. Ohta and Kanade performed Dynamic Programming in two stages: first an intra-scanline optimization is run for every edge pixel. Then an inter-scanline optimization is performed by using edge information to ensure consistency across the scanlines.

Witt and Weltin [28], [54] also matched only edge pixels but using a WTA strategy at the intra-scanline level while obtaining confidence measures. Then Dynamic Programming is used at the intra-scanline level to assign a disparity value for low confident or unmatched pixels. This work relies heavily on the edge information to reduce the number of match candidates and to ensure consistency. This suggests the importance of the image edges for solving the stereo-matching problem. This importance is addressed in Chapter 4. Felzenszwalb and Zabih [74] provide a review of the variants of dynamic programming which have shown good results in a trade-off between accuracy and processing time.

Total Variation (TV) approaches have been successfully implemented for image denoising, deblurring, segmentation and calculation of optical flow [83]. Variational methods formulate the energy E_{TV} in the form:

$$E_{TV}(u) = \int_{\Omega} \Psi_D(u)dp + \lambda \int_{\Omega} \Psi_S(u)dp \tag{2.31}$$

where Ω is the entire image domain; u corresponds to $u(p)$ which is in the disparity

domain for pixels p; $\Psi_D(p)$ corresponds to the image data and could be any of the dissimilarity function presented in Section 2.2.2; $\Psi_S(p)$ is a smoothness or regularization term which penalizes changes in disparity values in a neighbourhood and λ is the weight of the smoothness term relative to the data term.

Ranftl, Gehrig, Pock, *et al.* [83] firstly introduced a variational framework for solving the stereo-matching problem without using optical flow information. They defined a regularizer which uses gradient information in order to favour smoothness in homogeneous regions only and enforce a low smoothness otherwise. Although effective, this regularizer favours fronto-parallel surfaces. They obtain an average of 7.4% of error on the KITTI dataset [84] in a time of 7 seconds per image. Kuschk and Cremers [42] extended the regularizer of [83] by incorporating edge information in order to avoid favouring fronto-parallel surfaces. Then they minimized the energy function iteratively by using gradient ascending-descending in a primal-dual context. They achieved an error reduction compared to [83] and a running time of 20s.per image on the KITTI dataset [84] by implementing their approach in a GPU. Other variational approaches use information from optical flow [85], [86] in order to incorporate information from multiple frames into the disparity calculation. Although effective, the amount of required resources still limits its application in real-time low-cost systems.

Global approaches have proven to be successful in estimating accurate dense disparity maps but their application in low cost systems is still limited due to the high amount of computation required. Dynamic Programming has proven to be resource friendly but streaking artifacts are still an issue for its application. Streaking artifacts are misalignments of the objects boundaries across the scanlines of an image. This is shown in Figure 2-8. For applications like obstacle detection it is not required to know the disparity at every image pixel, as long as it is possible to identify the image objects. Based on this premise semi-dense disparity maps are used which are dense along the image edges but sparse in the remaining pixels of the image. By using this approach the required amount of resources is decreased without sacrificing the ability to represent the image semantics. More detail is presented in Chapter 4.

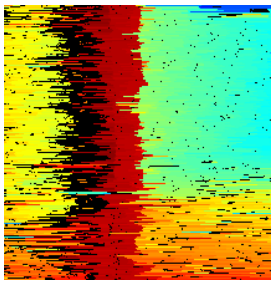


Figure 2-8: Streaking artifacts obtained by Dynamic Programming. The boundaries of a tree shown in the image present an offset across the scanlines.

2.2.3.3 Semi-Global Approaches

Semi-Global Matching (SGM) was proposed by Hirschmuller [87] in order to provide an approximation to a global cost calculation by using only local information. This resulted in an efficient approach able to work in running times similar to window-based approaches but reducing the effect of the aperture problem (see Figure 2-3).

In order to provide an approximation of the global cost, SGM defines an energy function by the formula

$$E_{SGM}(d) = \sum_{p \in P} \left(\delta(p, d_p) + \sum_{q \in N(p)} \xi(|d_p - d_q|) \right) \tag{2.32}$$

where P are the image pixels; $\delta(p, d_p)$ could be any dissimilarity measure presented in Section 2.2.2, Hirschmuller [87] proposed the use of Mutual Information and BT due to their robustness; $N(p)$ is a neighbourhood around p; and $\xi(x)$ is defined as:

$$\xi(x) = \begin{cases} \phi_1 & x = 1 \\ \phi_2 & x > 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.33}$$

where ϕ_1 is a penalization cost for pixels where disparity changes of 1 pixel, i.e. $|d_p - d_q| = 1$, ϕ_2 is a penalization cost for pixels with larger disparity changes.

In order to minimize this energy function Hirschmuller [87], proposed an approach in which the matching costs are aggregated in 1D from all directions equally. The cost L_r in

the direction r is then calculated as:

$$\begin{aligned} L_r(p,d) =& \delta(p,d_p) \\ &+ \min\{L_r(p-r,d), L_r(p-r,d-1)+\phi_1, \\ &L_r(p-r,d+1)+\phi_1, \min_i L_r(p-r,i)+\phi_2\} \\ &- \min_k L_r(p-r,k) \end{aligned} \tag{2.34}$$

By using these aggregated costs, it is possible to obtain the disparity map efficiently while penalizing discontinuities. Although reducing the computational complexity effectively, this approach still produces dense disparity maps, which might not be required for applications like obstacle detection, as will be shown in this thesis.

2.2.4 Disparity Propagation Approaches

Disparity propagation approaches take as input a sparse disparity map obtained by matching robust points, then global constraints are applied to fill the gaps in the disparity image to produce a dense disparity map.

Geiger, Roser, and Urtasun [50] proposed a method named Efficient Large Scale Stereo Matching (ELAS) where robust points are matched, then Delaunay triangulation is used to interpolate disparities using a picewise linear function. After this these interpolated disparities are used as prior and image likelihood and a Maximum A-Posteriori estimation is used to compute the disparities for every pixel. This approach proved to be fast to compute and have high accuracy and is one of the best performing in the literature. Although efficient and fast to compute, it would be reasonable to expect that a sparse disparity map, in which all of the surrounding objects could be identified, would be even faster and require less resources. This topic is explored in this thesis.

Sun, Mei, Jiao, *et al.* Sun, Mei, Jiao, *et al.* proposed an approach where the pixels are classified as stable or unstable based on Left-Right Consistency Check (LRC). Then a cost function which penalizes changes in disparity with respect to stable pixels and a geodesic filter is applied to the cost function in order to incorporate edge information and assign a disparity value to unstable pixels. This approach produces smooth and pleasant disparity images at low computational cost and high accuracy by using GPUs. Although this is one of the top performers on the Middlebury dataset, for many applications dense

disparity maps are not required, and therefore storage could be reduced by using sparse approaches. This work shows the importance of the use of edges in the matching process, which is explored further in this thesis.

2.2.5 Discussion on Depth Extraction

Although depth extraction is a well known problem and has received a lot of attention from the research community, it is still an open problem. Changes in illumination which are common in real life scenarios increase the difficulty of obtaining good quality solutions. Intensity-based descriptors are prone to suffer from changes in illuminations whereas descriptors based on morphological information have shown to be robust against gain and bias changes while being easy to compute.

Dense approaches contain high level of detail and are smooth but they require a large amount of resources for their computation. The approaches of Hirschmuller [87] and Geiger, Roser, and Urtasun [50] are the closest to real-time performance without requiring the use of specialized hardware like GPUs or FPGAs. Sparse approaches based on feature points do not represent the whole scene but might be used as starting points for iterative algorithms. Edge-based approaches have proven to be fast and able to represent the image semantics but they none of them exploits the connectivity information available on the edges.

Approaches which take into account the confidence of the matches have proven to reduce the number of outliers without incurring additional effort. This points to the importance of the use of confidence measures on the obtained disparity maps.

Local approaches although fast suffer from the aperture problem while global approaches require a large amount of computation. Although global approaches could be divided into small problems, the existing real-time solutions require the use of GPUs or FPGAs which limit their applicability in robotic scenarios.

In this thesis a new sparse algorithm for disparity calculations is proposed, this algorithm incorporates a pixel descriptor which is invariant to changes in illumination. The proposed approach uses confidence measures to obtain high confidence matches in a local manner for only a few anchor points. Then the disparity is propagated along the image edges achieving the capability of representing the image semantics without requiring the optimization of the disparity at each image pixel.

2.3 Obstacle Detection

Any moving entity requires the knowledge of its surroundings in order to avoid crashing. The determination of the areas where the entity could move without crashing is known as obstacle detection. Obstacle detection may be seen as the classification of the surroundings of the moving entity as either free or non-free space, where free space means areas into which the moving entity could move and non-free space is taken as being part of an obstacle. In this thesis an obstacle is taken as any object with a minimum height h_{obs} lying on the ground. No hanging objects are taken into account.

In the context of autonomous cars different sensors have been proposed in order to create a representation of their surroundings. LIDAR technology has proven to be successful in different outdoors scenarios but is costly and can be hacked without requiring physical contact [1]. This thesis is focused on the use of computer vision as sensor due to its passive nature. Further reasons are the low cost and easy availability of cameras for the automotive market; and the similarity of cameras to the human eyes.

Computer vision-based obstacle detection requires a methodology to calculate the distance for each of the objects contained in the images obtained by the camera. Approaches which use only one camera (monocular) require a way to relate the scale of the obtained images to the real world. For achieving this, some approaches use information from an active sensor such as sonar [89]. Another approach for obtaining scale information is to assume a static camera [90], a static world [91] or to try to identify changes in known objects such as angles between corners [92].

In the context of ground vehicles the world and camera are dynamic and there is no guarantee that all of the found objects in the surroundings are known beforehand. This limits the application of monocular approaches in this context. As shown in Section 2.2, stereo cameras may be used for extracting a disparity map, which along with triangulation may be used to estimate the distance of the objects in the image. This approach is used in this thesis.

Triangulation of dense disparity maps results in dense points clouds. These point clouds are similar to the ones obtained by LIDAR or laser-based approaches. This sim-

[1] Harrys, Mark (2015, Sept. 4th) Researcher Hacks Self-driving Car Sensors. *IEEE Spectrum*. Retrieved from http://spectrum.ieee.org

ilarity of the point clouds allows the use of a wide set of algorithms which have been developed for obstacle detection using LIDARs. Although it may provide a good representation of the surroundings of the car, a dense point cloud contains a large amount of information which results in slow processing and high resource usage. In order to reduce this resource requirement and speed up the processing different alternatives for representing the surroundings and detecting obstacles have been proposed, including: occupancy grids, elevation maps and stixels.

As the triangulation is performed before the obstacle detection, any error at this stage is carried forward to the location of the obstacles in the 3D world. In order to avoid this error propagation, some approaches which work directly on the disparity space have been proposed. These approaches perform triangulation only after the pixels have been classified as free or non-free space. More details on this are presented on Section 2.3.1.

2.3.1 Disparity Space Images

Disparity Space Images (DSI) contain enough information to identify characteristics of the 3D world without performing triangulation. The projection of disparity images to 3D world coordinates is prone to errors due to uncertainties in the rectification step and produces a sparseness effect for objects which are distant from the camera. Processing based on the disparity space avoids the requirement to represent the objects in the 3D world as object models, voxels or grids. Additionally, by avoiding triangulation the number of required computations is also reduced.

The u-disparity and v-disparity images are histograms obtained from the disparity image by accumulating disparity values along columns and rows respectively. The u-disparity was proposed by Labayrade, Aubert, and Tarel [93]. This image is obtained by accumulating the pixels of the same disparity d along the v axis of the image (horizontally) [93]. The u-disparity image was proposed by Hu and Uchimura [94] as a complement to the v-disparity image. This image is created by accumulating the pixels of the same disparity d along the u axis of the image (vertically).

Labayrade, Aubert, and Tarel [93] used the v-disparity image for estimating the ground profile by robust line fitting and for obstacle detection. By assuming the yaw and roll of the camera are minimum, the authors fitted a line to the v-disparity image by using the Hough transform. The slope of the line was then used to calculate the road profile and

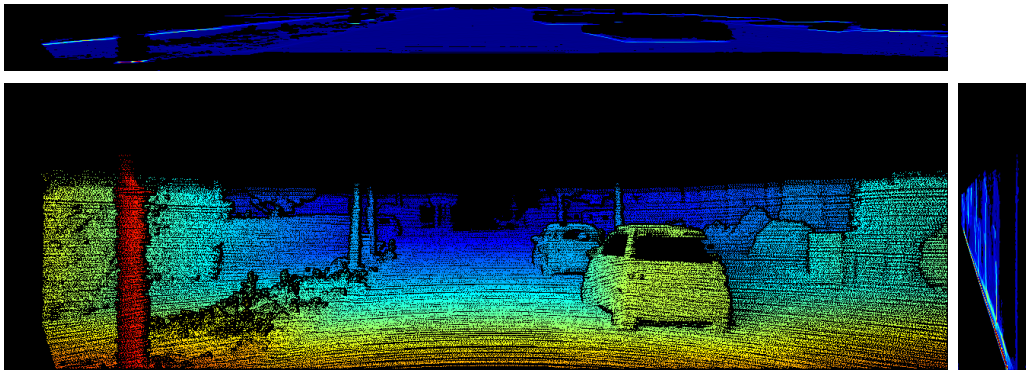


Figure 2-9: Examples of uv-disparity images obtained from a dense disparity image. At the top is located the u-disparity image, at the centre is located the source disparity image and at the right the v-disparity image is located.

height and pitch of the camera. Then by identifying vertical lines on the v-disparity image the boundaries between the road and the obstacles were identified. Although efficient this method for detecting obstacles tended to merge different obstacles close to each other into one. It is important to note that the method proposed by Labayrade, Aubert, and Tarel for ground profile extraction has become a popular method for extracting the ground profile in the literature on obstacle detection and is used in conjunction with other approaches.

Hu and Uchimura [94] used the v-disparity image for estimating the ground profile and proposed the u-disparity to identify the obstacles. Hu and Uchimura realized that by using the u-disparity image it was possible to identify the obstacles as they produce a close to horizontal line due to the fact that they tend to have a constant disparity. This approach proved to be efficient and has been widely used for obstacle detection. The main drawback of this approach is the requirement of a threshold for determining when an horizontal line should be taken as an obstacle or part of the road.

In [95] Broggi, Caraffi, Fedriga, *et al.* used image edges in the uv-disparity images for obstacle detection. They created the v-disparity image from a disparity map obtained only from high gradient pixels. They used the gradient angle to additionally remove false candidates and calculated the road profile using this information. After the road profile was estimated, they created a dense disparity image by constraining the search space to pixels with a disparity close to the disparity specified by the ground line on the v-disparity image. By doing this, the number of required computations was reduced. This points to the capability of the edges to identify obstacles.

In [96] Soquet, Perrollaz, Aubert, *et al.* used a propagation algorithm to improve

the road profile estimation on the v-disparity image and to improve the detection of the boundaries between the road and the obstacles. Then [97], [98] used uv-disparity images to create occupancy grids (see next section). [99] used uv-disparity images for ego-motion estimation. Fakhfakh, Gruyer, and Aubert [100] created a weighted version of the uv-disparity images by using using vertical and tilted windows to differentiate the road and the obstacles. Finally, Iloie, Giosan, and Nedevschi [101] used the uv-disparity images and segmentation to detect and cluster pixels associated with pedestrians.

The u-disparity and v-disparity images provide an efficient approach to road profile estimation and obstacle detection but, until now, they have been created only from dense disparity images. In this thesis a new variant is proposed created from sparse disparity maps obtained from edge images. For more information on this refer to Section 5.2.1.

2.3.2 Occupancy Grids

Occupancy grids represent the space by grids which reflect the occupancy of the surroundings using a probabilistic approach. They were first introduced by Elfes [102] for the representation of sonar data and for navigation. Since then, they have been extended to be computed for different kinds of sensors such as LIDAR [103] and stereo cameras [104]. Additionally they have been used for fusing the data from different sensors [104].

Occupancy grids require a formulation of the uncertainty in the data from the sensor. For the case of stereo-matching, this uncertainty could be obtained by modeling the triangulation error [104], by integrating the cost volume [105] or by using the matching confidence [106].

When disparity images are triangulated to obtain point clouds, objects located far from the camera produce only few points. This sparseness is not desirable in obstacle detection. In order to solve this problem, polar occupancy grids were introduced to represent the space by a grid with elements (u, d) where u is the horizontal location of the point in camera coordinates and d is the disparity of the pixel [107]. An additional advantage of polar occupancy grids over Cartesian occupancy grids is that they do not need triangulation, reducing therefore the computational cost. Badino, Franke, and Mester [107] used polar occupancy grid to detect the object boundaries against the ground plane by using Dynamic Programming obtaining fast and accurate results.

Occupancy grids allow a seamless sensor-merging approach and provide a good rep-

resentation of the 3D world. Although this information is useful for boundary detection, information regarding the height and geometry of objects is lost in its calculation as the occupancy grid only contains information from the probability than an object is present on its cells. In this thesis a new approach is presented which has a 1-1 equivalence to occupancy grids, but which allows to keep the geometric information about the objects in the scene. This is presented in Section 2.3.4.

2.3.3 Digital Elevation Maps

Digital Elevation Maps (DEM) provide a grid representation (top-view) of the 3D world where the value associated with each cell corresponds to the height of an object or ground. They were first proposed by Zhang [108] for the creation of 3D maps for planetary rovers. Elevation maps have the advantage of providing an estimate for the height of the objects in the surroundings, whereas occupancy grids only indicate the free or non-free space. As with occupancy grids, elevation maps can be calculated from point clouds independently of the used sensor.

In [108] Zhang used planar and quadratic surface models to fit point cloud data obtained by stereo-matching. Then an iterative approach was used to reduce erroneous points by thresholding the point-to-surface distances and recalculating the surfaces. The obstacles were identified by thresholding the DEM.

Oniga and Nedevschi [109] classified the point cloud data as road or obstacle according to the density of the DEM cells. They assumed that obstacle cells would have a larger density than the road cells at a given depth due to the fact that fronto-parallel surfaces create more points than the road. Then they fitted a quadratic surface to the road by using a region growing technique with a robust seed obtained by RANSAC and used this road model to identify the points corresponding to one object. Vatavu, Danescu, and Nedevschi [110] used this approach to classify the road and obstacles and used a free-form object model to identify the object boundaries and perform tracking.

In order to avoid triangulation, Vergauwen, Pollefeys, and Van Gool [111] proposed to obtain elevation maps from disparity maps by selecting a point with zero height in both cameras and simulating a vertical line located at that point. Then a light source is placed in the left view and the projected shadow in the right image is intersected with the vertical line in the right image. By doing this they were able to identify the part of the left image

which is visible and the part which is occluded. The same applies if the right image is used as reference. This approach allowed them to create elevation maps at any resolution by applying interpolation and incorporating information from multiple views taken over a period of time from the stereo-camera.

Although Digital Elevation Maps allow obstacle detection in dense stereo data, they require the processing of every point in the cloud. For highly dense point clouds such as the ones obtained by high resolution images, the required resources would limit the application in embedded and real-time systems. The creation of DEM from sparse or semi-dense disparity maps is still an open problem and could be the focus of further research.

2.3.4 Stixels

Stixels represent the 3D world via vertical rectangles with an associated disparity drawn over the images captured by the stereo-camera. It is implicit that any image pixel, which does not belong to a stixel, belongs to the road. Stixels allow a compact representation by requiring only two points and a depth value to represent one stixel. As it is expected that one stixel contains a large number of pixels the resources required to store the image are significantly low.

Stixels were first proposed by Badino, Franke, and Pfeiffer [112] to perform efficiently the analysis of dense disparity maps from high-resolution images. Although the stixel-world was proposed by Badino, Franke, and Pfeiffer, a similar representation of the free space was proposed previously by Rebut, Toulminet, and Bensrhair [113], but instead of using vertical rectangles, they used vertical lines to represent the non-free space.

Similarly Kubota, Nakano, and Okamoto [114] used vertical columns to represent the obstacles in the disparity space, although their research was focused on the detection of the boundaries between the obstacles and the road by Dynamic Programming while ignoring the height of the obstacles.

Badino, Franke, and Pfeiffer [112] proposed to create the stixel-world using dense disparity images, triangulation to get real-world coordinates and polar occupancy grids to identify the objects. They first computed a dense disparity image by SGM on an FPGA. Then they obtained a polar occupancy grid as described in Section 2.3.2 to identify the free space. After this, they applied background subtraction to process only foreground

objects and applied Dynamic Programming for identifying the best segmentation between the background and the upper boundary of the objects. Finally they used the obtained information to create the stixels while applying outlier rejection and noise suppression. This representation turned out to be robust and compact while providing enough information to identify the traffic environment (obstacles and the road).

Benenson, Timofte, and Van Gool [115] used stixels to identify pedestrians, but instead of dense disparity maps, they used the cost volume for the processing. This reduced the possible errors on triangulation and sped up the processing. They first calculated the cost volume by a local approach. Then the v-disparity image was obtained but the pixels represented the summed cost for each (v, d) pair. Then robust line fitting was used to estimate the ground plane. Following the approach from [114] Dynamic Programming was used to get the stixels depth from the u-disparity image. After this the height of the stixels was estimated by identifying local minima in the cost volume which correspond to the disparity assigned to each stixel. They evaluated the accuracy of the stixels in identifying pedestrians using an annotated database obtaining good results.

Following the work on [115], Benenson, Mathias, Timofte, *et al.* [116] proposed an alternative formulation of the stixel world in order to speed up the object detection. The ground plane was estimated using the v-disparity image obtained from the cost volume as in [115] but they computed only one-out-of-N rows below the horizon, taking the horizon as a input data. The estimation of the stixel distance was performed by vertically splitting the image into multiple row bands and for each image column the maximum horizontal gradient was selected. Also as they assumed the objects are wider than one column the image was scanned only at one-out-of-M columns. They reformulated the cost function used in [115] for estimating the stixel distance to reflect these changes. The authors state that they were able to reach a performance of 260 fps for the stixel computation on a high-end laptop using a combination of GPUs and multi-core processing.

Pfeiffer, Gehrig, and Schneider [65] focused on the incorporation of confidence measures to improve the accuracy of the computation of the stixels. They used SGM [87] for dense stereo-matching and analysed different confidence metrics. Then the confidence metric were used as outlier probability for the matches. They proved that the MLM confidence metric (see Table 2.4) provided the best separation between outliers and inliers and used it to determine the outlier probability and solve a MAP problem where the outlier probability

is defined as the probability that a disparity d_v on row v belongs to a stixel s_n. In order to take into account a global outlier model, a set of bounds were applied to the outlier probability allowing an accurate way to remove false positives when outliers are not identified correctly. Additionally thresholds on the confidence are used to remove outliers.

Although the approaches based on stixels are fast and accurate, they rely on the use of dense cost volumes or dense disparity images. It would be reasonable to expect that an approach which uses sparse techniques would benefit from the accuracy, speed and compactness of the stixels representation. Therefore an approach which is based on sparse techniques is presented in Chapter 5.

2.3.5 Other Approaches

Other approaches for obstacle detection analyse every point in the cloud independently, fit surfaces to point clouds, or use flow fields.

Approaches, which analyse the entire point cloud and label every point as obstacle or free space, use one of the following techniques: they assume the obstacles are any point not belonging to the ground [117]; they use neighbouring relationships on the points to fit surfaces to the cloud [118]; they use 2D segmentation to cluster the points in the cloud and perform surface fitting [119]; they use only sparse point clouds created from robust feature matching and clustering techniques [120]; or they cluster the points into cuboids.

Cuboid models are able to fit the geometry of most cars [121]. They have an associated width, height and length $C = (w, l, h)^T$ and, when used for tracking, a position and speed are also associated $C = (w, l, h, p, s)^T$. Cuboid models can be obtained from point clouds [27], [122], [123] but they could be also computed from other representations such as occupancy grids [124] and stixels [121]. Computation of cuboids from dense point clouds could be costly but mid-level representations like stixels present an efficient way to estimate them. Therefore further analysis of this kind of approach is a promising area for further research.

2D and 3D flow fields provide an alternate way to detect obstacles by identifying the displacement of the scene contents. 2D flow fields are know as optical flow. This flow represents the apparent displacement of pixels in a sequence of 2D images [125], [126]. Based on this 2D flow Sebesta and Baillieul [91] proposed a method for obstacle detection

by taking into account the fact that close objects produce a larger displacement than far objects.

3D flow fields are known as scene flow, this flow represents the three-dimensional displacement of points in a sequence of point clouds [127]. Several approaches have been proposed for calculating the scene flow and they have been proven to be successful for obstacle detection by approximating the scene by planar elements [128], [129]. Although flow fields allow the estimation and analysis of possible trajectories for obstacles, tracking of obstacles across a sequence of images is out of the scope of this thesis. It is left as future research, to evaluate the proposed representations in this thesis for estimating the motion of the obstacles.

2.3.6 Discussion of Obstacle Detection

Several approaches have been used to determine the free space around a mobile robot and provide an efficient representation. Unfortunately the calculation of the obstacle map in real-time using low computational resources is still a challenging task. Most of the algorithms found in the literature rely on the use of dense depth maps which require a considerable amount of computational resources to be calculated. Only a few have been designed with sparseness in mind but the identification of the proper features in the images is not a straightforward task.

In order to cope with this problem, in this thesis an obstacle detector is proposed based on sparse depth maps obtained from image edges. Then it is shown that this information is enough to obtain mid-level representations like stixels without sacrificing robustness and the capability to identify all the possible obstacles. The proposed algorithm for estimating stixels from sparse edge maps in 3D is presented in Chapter 5.

2.4 Discussion of the Literature Review

After reviewing the state-of-the-art in obstacle detection for autonomous navigation it could be said that obstacle detection using stereo vision is promising but has many challenges. Accurate state-of-the-art approaches for extracting depth using stereo-cameras require a large number of computations resulting in a long processing time. Therefore they are not suitable for autonomous vehicles. By reducing the accuracy some approaches

provide a faster performance but are still unable to provide results in a sub-second time-frame without the use of specialized hardware like FPGAs and GPUs which might be expensive, require high thermal dissipation, be hard to maintain or consume high power.

A summary of the weaknesses of current approaches to obstacle detection using computer vision are:

- Sensitivity to changes in illumination.
- Sensitivity to noise in the sensor.
- Computationally intensive.
- Accurate systems require high amount of computational resources and are slow to compute.
- Most of the systems require previous knowledge of the obstacle geometry.

Although some approaches aim to solve some of these weaknesses, no approach has been found to overcome all of them simultaneously. Therefore, this thesis presents three approaches that try to fill this gap by focusing on the use of image edges for obtaining sparse representations.

3

Pixel Description Robust to Changes in Illumination

3.1 Introduction

This chapter focuses on the introduction of two new pixel descriptors into the stereo-matching context. The first is an extension of the Census Transform with the goal of incorporating edge information and reducing the search space at the best-match selection stage. The second is the Complete Rank Transform, which has so far only been used for the calculation of optical flow but it is reasonable to expect that its high discrimination power would benefit the stereo-matching problem. Figure 3-1 shows the areas of contribution of the proposed pixel descriptors.

In the previous chapter, the literature review described some of the weaknesses in the current methods for obtaining real-time 3D maps. In order to deal with these weaknesses, this thesis proposes a new pixel descriptor, which is an extension of the existing Census Transform. This descriptor incorporates gradient information in order to identify both the image edges and highly textured areas allowing a reduction in the search space for local-based stereo-matching approaches. This results in a reduction in the overall number of computations required to calculate a disparity map.

Image edges allow the representation of the image contents using only a few pixels whilst keeping the semantic and geometric meaning around the object boundaries [130], [16], [131]. This reduction in the number of pixels is desirable for local stereo-matching approaches where each pixel must be compared with d_{max} pixels at the best-match selection stage.

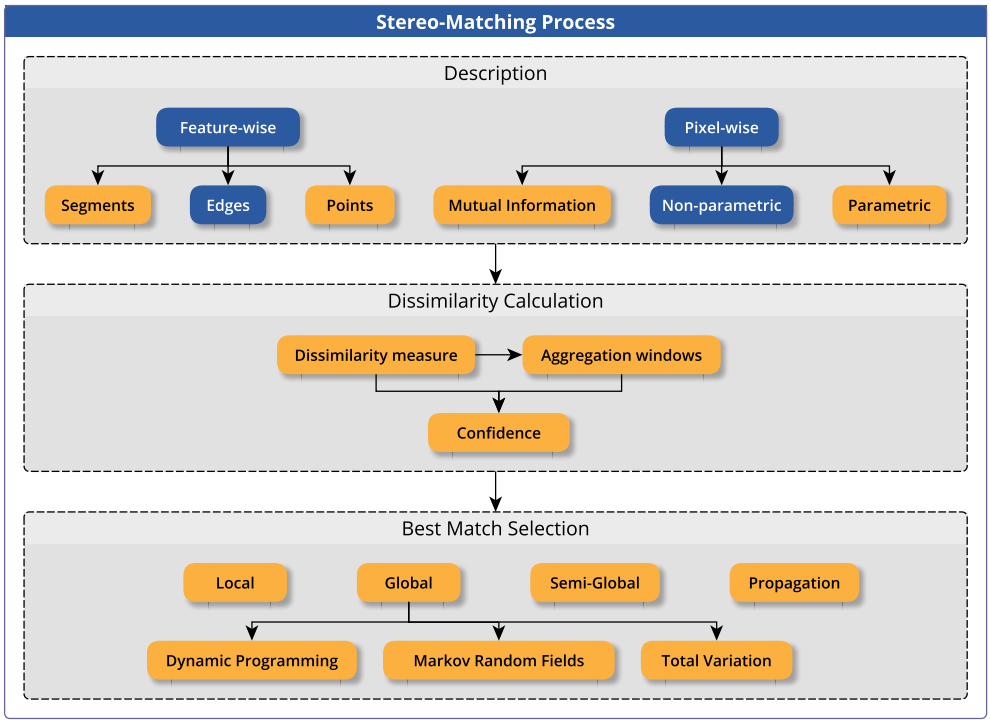


Figure 3-1: Areas of the stereo-matching process contributed (blue) by the proposed pixel descriptors. The TCT bridges feature-wise and pixel-wise description by allowing the identification of the pixels around the edges during the computation of a non-parametric pixel descriptor.

The chapter is organized as follows: first, a new pixel descriptor based on the Census Transform is presented along with the Complete Rank Transform pixel descriptor. Second, both pixel descriptors are evaluated in a local stereo-matching environment and compared against both the original Census Transform and an intensity window descriptor which is widely used in stereo-matching. Third, a discussion of the obtained results is presented.

3.2 Proposed Descriptors

This section introduces two new pixel descriptors. The first is based on the Census Transform and the second is a new application of the Complete Rank Transform, which was previously used for calculating optical flow. Both image transforms are able to produce pixel descriptors which are robust to changes in illumination as shown in [34]. Additionally, the first new pixel descriptor has the advantages of a binary representation and the capability of detecting image edges without a separate edge detection stage. An example of the transforms for a window size of 3×3 is shown in Figure 3-2.

4	14	83
4	25	88
3	15	65

(a) Intensities

1	1	0
1		0
1	1	0

(b) CT

1	0	0
1		0
1	0	0

(c) TCT

1	3	7
1	5	8
0	4	6

(d) CRT

Figure 3-2: Result of the functions $T(p,q)$, $T_{tct}(p,q)$ with $\varepsilon = 15$ and $r(p)$ for the CT, TCT and CRT respectively. Figure a shows intensities corresponding to a 3×3 window. Figures b, c, d show the contribution to the descriptor from each of the pixels in the input intensities.

3.2.1 The Thresholded Census Transform

The Thresholded Census Transform (TCT) extends the Census Transform (CT) (see Section 2.2.1.1) by including a similarity measure ε while keeping its binary representation. The addition of this similarity measure has two effects: first, only those pixels which have an intensity significantly smaller than the centre pixel can affect the value of the descriptor. Second, the similarity measure ε behaves like a threshold on the gradient. This allows the detection of edges and highly textured areas without the need for an additional stage. More detail on this is presented in Section 3.2.1.1.

The Thresholded Census Transform is similar to the Ternary Census Transform

$$T_{ct3}(p,p') = \begin{cases} 0 & I(p) - I(p') > \varepsilon \\ 1 & |I(p) - I(p')| \leq \varepsilon \\ 2 & I(p') - I(p) > \varepsilon \end{cases} \tag{2.9}$$

which also adds a similarity measure but the latter does not have the advantages of a binary string representation. The Ternary Census Transform adds a similarity measure ε to the comparison function $T_{ct}(p,q)$ with the purpose of encoding pixels which are similar to the centre pixel (see page 19). This concept of a similarity measure is incorporated into the TCT but instead of setting the pixels with an intensity similar to the centre pixel to 2, they are set to zero. It means when all of the pixels in the neighbourhood $N(p)$ are in the interval $(I(p)-\varepsilon, I(p)+\varepsilon)$ the resulting string would be formed by zeroes only, meanwhile on the Ternary Census Transform it would be formed by $0, 1$ or 2.

The Thresholded Census Transform $\xi_{tct}(p)$ modifies the function $T(p,q)$ used in the

4	14	83
4	25	88
3	15	65

(a) Intensities

6	16	85
6	25	90
5	17	67

(b) Vigneting

14	24	93
14	35	88
13	25	75

(c) Bias

1	0	0
1		0
1	0	0

(d) TCT

Figure 3-3: Effect of changes in illumination on the Thresholded Census Transform using a similarity $\varepsilon = 15$. The centre pixel is highlighted in grey. Figure b) shows vigneting effect (the intensity is increased by 2 units as we move from the centre pixel). Figure c) shows bias effect (the intensity is increased by 10 units for all of the pixels in the window). There is no change in the result of the comparison function.

Census Transform

$$T(p,q) = \begin{cases} 1 & \text{if } I(q) < I(p) \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

to incorporate the similarity threshold ε. Pixels which are similar to the centre pixel produce a zero-bits resulting in strings formed only by zeroes on non-edge areas. The comparison function for the TCT is expressed as:

$$T_{tct}(p,q) = \begin{cases} 1 & \text{if } I(q) < I(p) - \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where ε is the incorporated similarity measure. An example of the effect of this modified comparison functions is shown in Figure 3-2.

By using the ordering of the pixels to produce the descriptor, the TCT keeps the robustness to changes in illumination native to the Census Transform. As shown in Figure 3-3, the vignetting effect and bias effects (see Section 2.2.2) produce no changes in the result of the comparison function.

As only those pixels which are significantly different from the centre pixel are used in the TCT, pixels in uniform or untextured areas are easy to identify during the creation of the TCT as they produce a zero-string. This identification is useful for the best match selection process, as pixels which are known to be textureless can be ignored in the dissimilarity calculation thus reducing the total number of required computations. When this identification is not known before the best match selection, texture maps must be applied after the disparity calculation to avoid spurious matches as in [45]. The effect of the reduction in the number of candidates in the best match selection is highly important

as the displacement of the objects across the stereo-camera grows as the baseline grows.

3.2.1.1 Edge Detection by the TCT

The gradient at a pixel in an image measures how quickly the intensity function changes in a specific direction at that pixel. Peaks in the gradient correspond to image edges[132], [130], [133]. Image edges are important as they generally represent object boundaries [55], [134]. In order to identify the peaks in the gradient it is common to apply a threshold t_g and keep only those pixels whose gradient is larger thant the threshold t_g.

As shown by Hafner, Demetz, and Weickert, the Census Transform incorporates the gradient information indirectly [135]. Hafner, Demetz, and Weickert showed that each Census bit encodes the derivative of the intensity image in the direction of q. This occurs as part of the comparison function $T_{ct}(p, q)$ (Equation (3.1)) against the centre pixel p.

An alternative representation of the comparison function $T_{ct}(p, q)$ used in the Census Transform is:

$$T_{ct}(p,q) = \begin{cases} 1 & 0 < I(p) - I(q) \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

The differential $I(p)-I(q)$ in Equation (3.2) corresponds to a calculation of the gradient at pixel p in the direction of q. The conditional $0 < I(p)-I(q)$ indicates that only positive gradients would produce 1-bits on the CT string. By adding the similarity measure ε to the conditional $\varepsilon < I(p) - I(q)$ the differential $I(p) - I(q)$ is restricted to be positive and larger than ε. This behaviour corresponds to a thresholding operation on the gradient and corresponds to:

$$T_{tct}(p,q) = \begin{cases} 1 & \varepsilon < I(p) - I(q) \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

which is equivalent to Equation (3.1). Taking only the positive gradients would result in descriptors which are strings containing only zeros at darker side of the edge, meanwhile the other side may contain 0 or 1.

3.2.1.2 Computational Complexity Analysis for the Pixel Descriptor

This section aims to show the impact of adding the similarity measure to the TCT as pixel descriptor.

The steps required for computing the Census Transform are as follows:

1. Get value of the centre pixel. $O(1)$.
2. Compare one pixel against the centre pixel. $O(1)$.
3. Update the bit value according to the result of the comparison. $O(1)$.
4. Repeat this for for $mn-1$ pixels in a window of size $m \times n$. $O(mn-1) \times (O(1) + O(1) + O(1)) = O(3(mn-1)) = O(mn)$

The addition of the similarity value adds one extra operation which is the subtraction of the similarity threshold ε. The required steps for this are as follows:

1. Get value of the centre pixel. $O(1)$.
2. Subtract the similarity threshold ε. $O(1)$
3. Compare one pixel against the result of the subtraction of the similarity threshold ε. $O(1)$.
4. Update the bit value according to the result of the comparison. $O(1)$.
5. Repeat this for for $mn-1$ pixels in a window of size mn. $O(mn-1)*(O(1)+O(1)+O(1)+O(1)) = O(4(mn-1)) = O(mn)$

Although a subtraction operation is added to the TCT with respect to the CT, this operation has no impact on the final number of operations. Therefore no overhead would be expected by using the TCT instead of the CT for calculating disparity maps. In contrast, it is expected that by using the similarity ε the search space would be constrained and a reduction in the required number of operations would be expected.

3.2.2 The Complete Rank Transform

The Complete Rank Transform (CRT) was proposed by Demetz, Hafner, and Weickert for the calculation of optical flow [34]. This image transform extends the Rank Transform proposed by [33] (see Equation (2.6)) by calculating the rank of every pixel in the neighbourhood $N(p)$ instead of just for the centre pixel as in the Rank Transform. By doing this the range of values obtained by the descriptor is increased from $mn-1$ for the Rank

4	14	83
4	25	88
3	15	65

(a) Intensities

1	3	7
1	5	8
0	4	6

(b) Rank $r(p)$ for each pixel

Figure 3-4: Example of the Complete Rank Transform, $\xi_{crt}(p) = 1, 3, 7, 1, 5, 8, 0, 4, 6$.

Transform to $(mn)^2 - 1$ for the CRT. This increased range of values decreases the chances that two different pixels produce the same descriptor.

The Complete Rank Transform uses the rank $r(p)$ (Equation (2.4)) defined for the Rank Transform, but it concatenates the rank of every pixel p_i in the neighbourhood of $N(p)$, i.e.

$$\xi_{crt}(p) = \bigotimes_{p_i \in N(p)} r(p_i) \tag{3.4}$$

this results in strings formed by integer number from 0 to $mn-1$ where mn is the number of elements in the neighbourhood $N(p)$ of size $m \times n$. An example of the CRT is shown in Figure 3-4.

It is expected that as the CRT produces a more "unique" string, the accuracy of the best-match selection would be increased at the price of increasing the number of required operations. This is analysed in Section 3.3.3.

3.3 Evaluation

In order to evaluate the performance of the Thresholded Census Transform and the Complete Rank Transform, a local stereo-matching approach is used in this thesis. The implemented approach is based on that proposed by Humenberger, Zinner, Weber, *et al.* [45] as it was proven to provide real-time disparity maps in an embedded system by using hardware specific optimizations. In order to provide a baseline for the performance of the approach, none of the hardware-specific optimizations are implemented in this thesis at the price of losing the real-time performance. In order to keep a benchmark of the performance that could be achieved by using hardware-specific optimizations, the algorithm proposed in [45] has been implemented and evaluated on the same images producing a comparison factor.

By avoiding hardware-specific optimization, only the processor speed, system bus and

memory access would affect the performance of the implementations in this thesis. The machine used for testing is a standard laptop with an Intel Core i7-2675QM running at 2.2GHz with 8GB of RAM.

The testing is performed on the Middlebury v3 [59] and KITTI stereo 2015 [129] datasets. The Middlebury dataset provides 15 rectified stereo-images aligned horizontally with dense ground truth for indoor scenarios obtained by using structured lighting techniques. The Middlebury dataset includes images with changes in illumination and large occlusions. The evaluation is performed on the quarter image sizes provided in the evaluation website [1]. The KITTI dataset provides 200 rectified stereo-images aligned horizontally with semi-dense ground truth on road scenarios obtained by a LIDAR. The KITTI dataset provides one megapixel resolution imagery with shadows, reflections and illumination changes. The evaluation is performed on the training images at full resolution provided on the evaluation website [2].

As widely used in stereo-matching benchmarks [59], [129], accuracy *acc* is defined as the ratio of the number of pixels with an error smaller than a specific threshold to the number of obtained disparities i.e. :

$$acc = \frac{\#\text{ pixels with error } < e_t}{\#\text{ obtained disparities}} \tag{3.5}$$

Recall is defined as the ratio of pixels with an error smaller than a specific threshold to total number of available disparities on the ground truth i.e. :

$$rec = \frac{\#\text{ pixels with error } < e_t}{\#\text{ total disparities on ground truth}} \tag{3.6}$$

The evaluation first compares the Thresholded Census Transform against the Census Transform and analyses the effect of the window size and the similarity in the produced disparity maps. Also the edge-detection capability is tested for the Thresholded Census Transform. Then the Complete Rank Transform is compared against an intensity neighbourhood descriptor, which does not require any additional processing for its calculation.

[1] `http://vision.middlebury.edu/stereo/eval3/`
[2] `http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo`

3.3.1 Stereo-Matching Approach

The implemented approach for stereo-matching is based on that proposed by Humenberger, Zinner, Weber, *et al.* [45] with a few changes to avoid any hardware-specific optimizations (intrinsics, vector operations and parallel processing).

First the pixel descriptors are calculated for the left and right images. In order to calculate these descriptors the images are loaded into memory and converted to gray-scale by using the OpenCV library [136].

After the descriptors are obtained, the cost volume (see Section 2.2.3.1) is calculated and stored in memory using the approach in [45], [48] in order to reduce the memory usage. The approach from Zinner, Humenberger, Ambrosch, *et al.* [48] was shown to be memory efficient for obtaining the cost of the left to right and right to left directions in one pass. As shown in [25], the Census Transform strongly benefits from the use of aggregation windows, therefore different aggregation windows sizes (see Equation (2.26)) are tested in order to identify the best performing window size.

For the matching a Winner Takes All (WTA) approach is used as in [45], [48]. Only matches that pass the LRC (see Section 2.2.3.1) with a threshold of $t_{lrc} = 1$ are kept. Additionally matches with confidence lower than the threshold t_c are discarded. The confidence metric used is based on the β_{PKR} confidence metric (see Table 2.4) but with small changes to avoid a zero denominator and to bound the confidence within the interval $(0, 1]$. The modified version is named Inverse Peak Ratio (IPKR):

$$\beta_{PKR^{-1}} = 1 - \frac{c_1 + 1}{c_{2m} + 1} \tag{3.7}$$

where c_1 is the smallest cost and c_{2m} is the second local minumum as defined in β_{PKR}. A confidence value close to zero would mean that the match is likely to be wrong, whereas a confidence value close to one means that the match is likely to be good. An analysis of this is presented in [45], where it is shown that a small difference between the minimum cost and other local minima makes it hard to say if the difference is caused by noise or some repetitive structures, whereas a large difference is a sign that the cost curve has a prominent minimum which is desirable. This difference is expressed by the ratio $\frac{c_1+1}{c_{2m}+1}$. The confidence threshold t_c is analysed further to determine the best value in the following

	Time (ms.)	
	[45]hw. optimized	no hw. optimized
Description	8.7	98
Cost volume	22.61	146
Matching	19.37	81
LRC	1.4	3
Total	52.08	328

Table 3.1: Comparison of the running time between the implementation in [45] and an implementation of the same approach without hardware specific optimizations.

section.

In order to validate the software implementation and to have a benchmark for the running times, the approach used in [45] was tested on the same 31 images of the Middlebury datasets from 2001 [77], 2003 [137], 2005 [138], [139] and 2006 [138], [139].

No hardware-specific optimization are used in the implementation used in this thesis, therefore it is required to have a reference for the speed-up which could be attained by using hardware-specific code. In order to compare the running times, Table 3.1 shows the average time required to produce the disparity maps for the Middlebury images used in [45]. Although the real-time performance cannot be reached by the implementation used in this thesis, the results provide a comparison of the expected performance by using hardware optimizations. Table 3.1 shows that by using a hardware optimized code (SSE instructions for a PC) a speed up of up to 6X could be achieved. Even larger speed-ups could be achieved by using GPU or DSP platforms.

3.3.2 Thresholded Census Transform

This section details the experiments performed on the TCT in order to measure the effect of the incorporation of the similarity measure on accuracy and computation speed for the obtained disparity maps. First the effect of the window size on the TCT is analyzed. Then the effect of the aggregation window size is analysed. After this, its edge detection capability is assessed. Finally, different values for the confidence parameter are tested.

3.3.2.1 Window Size Effect on the TCT

First the effect of the transform window size on the TCT is analysed. In order to do this, the TCT is evaluated using the stereo-matching approach detailed in Section 3.3.1. No aggregation window or confidence information is used and all the attention is focused on

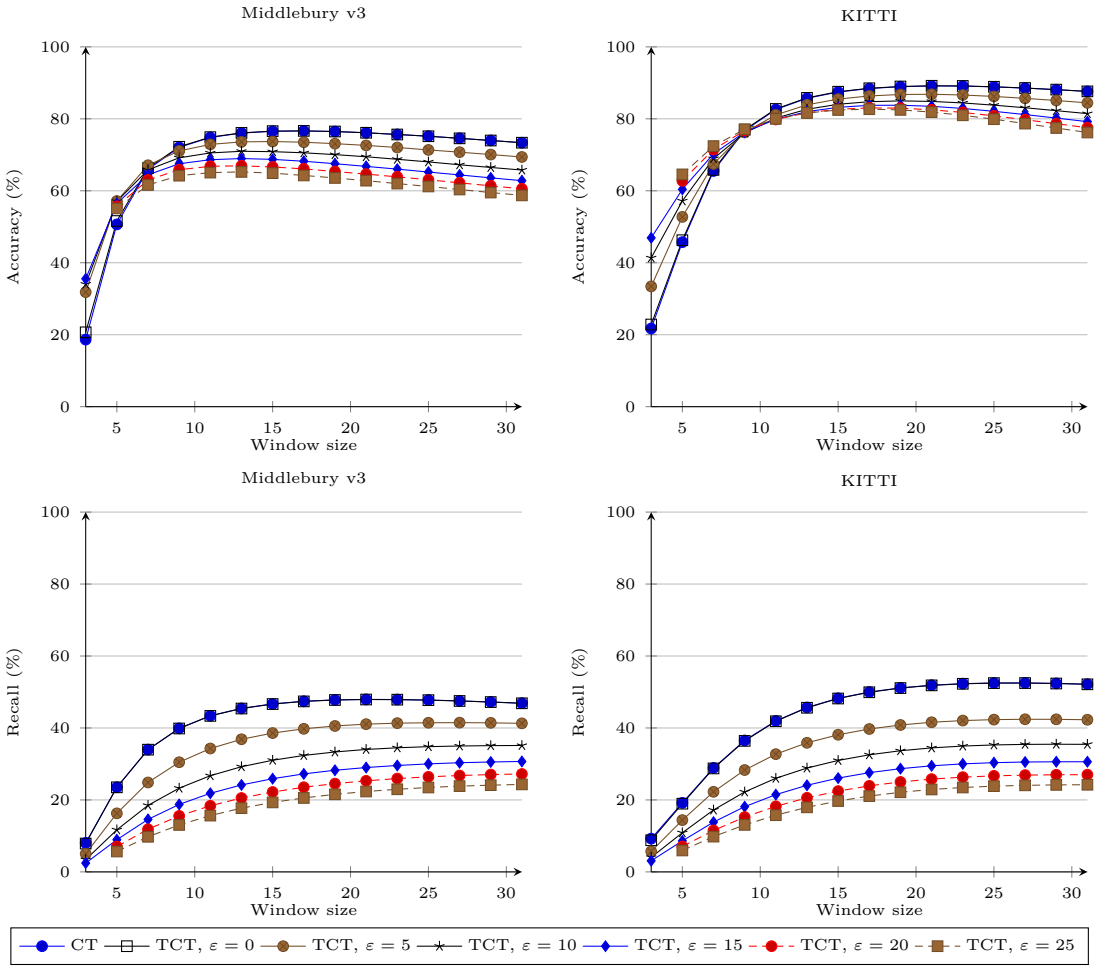


Figure 3-5: Effect of the window size on CT and TCT.

the TCT. The testing is performed by using a square window of size $n \times n$. The window size n is set to lie within the interval $(3, 29)$ using only odd integers so the centre pixel has the same number of pixels on each side of the window. Window sizes larger than 29×29 take a long time to be computed which is not desired for a real-time system. As it is expected that the similarity ε affects the performance of the TCT as descriptor, different similarity values are used in the interval $[0, 25]$ as values larger than 25 showed a marked decrease on the recall. A deeper focus on the effect of the similarity on the metric is shown in the following sections.

Figure 3-5 shows the effect of the window size on the accuracy and recall of the TCT. The Census Transform is shown as reference. The shown plots indicate the average accuracy and recall for the Middlebury V3 [59] and KITTI 2015 datasets [129]. As no sub-pixel values are calculated, the error threshold is set to 1. It can be seen that for the Middlebury dataset the maximum accuracy is obtained by using a transform window size of 17×17

whereas the maximum recall is obtained by a transform window size of 21×21. This is explained by the fact that large window sizes produce longer strings. Therefore the number of values which may be represented is increased. Note that the maximum Hamming distance between two strings of the same length is equal to the number of bits in each string.

For the KITTI 2015 dataset, the maximum accuracy is obtained by a transform window size of 21×21 whereas the maximum recall is obtained by a window of 27×27. As suggested by the KITTI 2015 dataset, a disparity is taken as erroneous if the error is larger than 3 units. The increased window sizes on this dataset are explained by the nature of images. Although the images in the KITTI dataset are high-resolution, they have a higher level of noise than the ones in the Middlebury v3 dataset; the number of objects out of focus is high and reflections are present in the images.

From Figure 3-5 it can also be seen that the similarity value of 0 produces disparity maps with almost the same accuracy and recall as the CT. It also can be seen that for every transform window size the use of the similarity measure increases the accuracy in respect to the Census Transform. This is explained by the fact that larger similarity values would translate into fewer candidates in the search range. As there are fewer candidates the chances of finding the right match is higher but the number of obtained values is smaller, therefore the recall is smaller than the Census Transform. For larger windows on the other side, large similarity values translate into more values set to zero in the binary string. For large windows which already produce many "unique" values, the number of values is decreased. A deeper analysis of the effect of the similarity ε on the TCT is presented in Section 3.3.2.3.

3.3.2.2 Aggregation Window Effect on the TCT

As showed by Hirschmuller and Scharstein [25] the use of aggregation windows greatly improves the performance of the Census Transform. This is due to the fact that a window size of $n \times n$ would produce only $n^2 - 1$ different values, and the aggregation increases the possible number of values by "aggregating" the values in a neighbourhood. It is expected that a similar behaviour would be presented by the TCT. In order to test this, different aggregation windows sizes are used on the TCT. The squared aggregation window sizes used are in the range of $[3, 17]$ as larger values produce a drop in the performance and

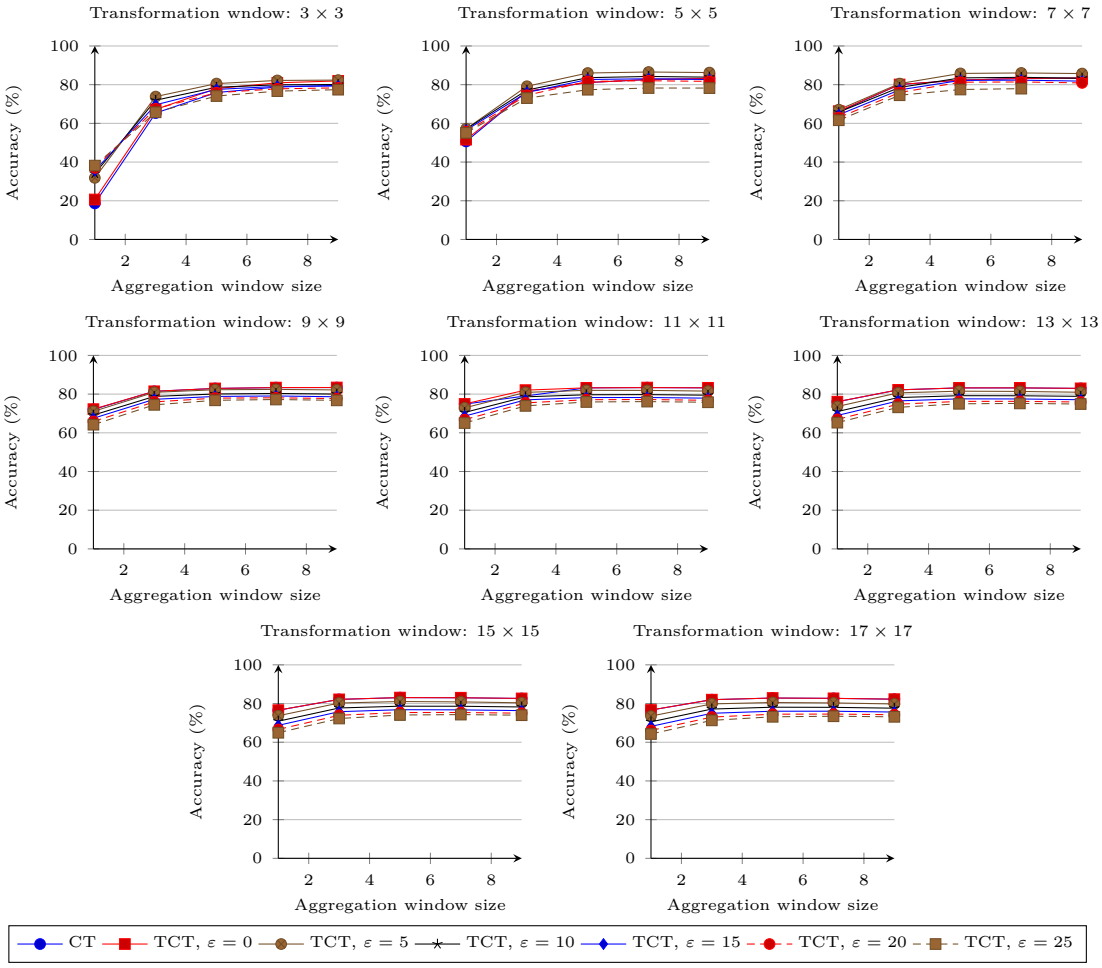


Figure 3-6: Effect of the aggregation window on the accuracy for the Middlebury v3 dataset using different transform window sizes.

increase the required number of computations.

Figures 3-6 and 3-7 show the average accuracy and recall for different aggregation window sizes on the Middlebury v3 dataset. It can be seen that for small transform windows large aggregation windows produce an increase of the accuracy but for large transform windows there is no significant advantage of the use of aggregation windows larger than 5×5. It is also shown that no significant increase in accuracy is obtained for transform windows larger than 7×7. For a similarity $\varepsilon = 0$ a window size of 7×7 with an aggregation window of 9×9 produces a maximum in accuracy of 77% but the difference with an aggregation window of 5×5 is close to only 1% whereas small aggregation windows require less computation than large.

For the recall, larger aggregation windows lead to higher recall as more disparities are obtained, but due to the fact that the obtained disparities are located around the

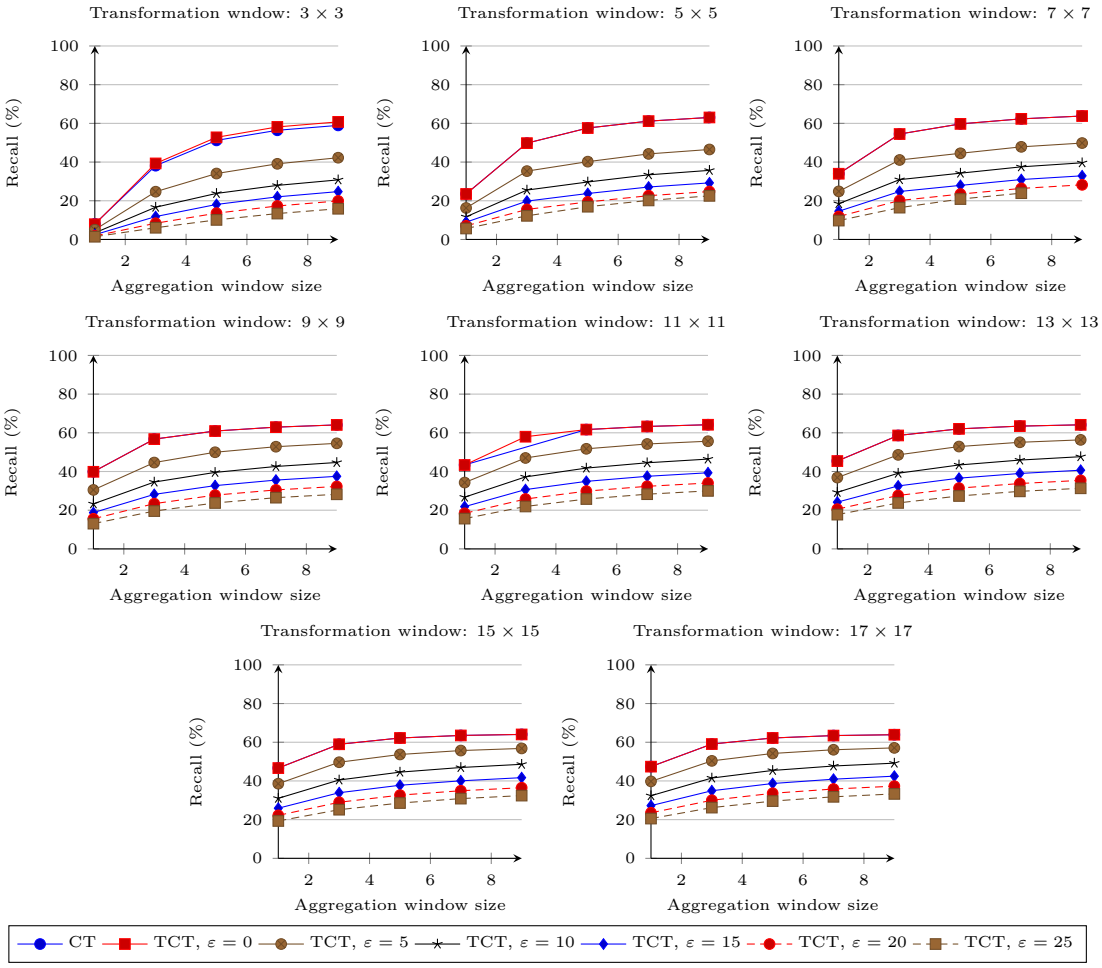


Figure 3-7: Effect of the aggregation window on the recall for the Middlebury v3 dataset using different window sizes for the image transform.

edges a human viewer is able to identify the image contents even on the low recall images. Therefore the increase in the recall is not a determinant factor in the selection of the aggregation window size. For the Middlebury v3 dataset, the best window combination is 7×7 for the transform window and 5×5 for the aggregation window. This combination would produce a good trade-off between accuracy and computational load. It is important to note that the effect of the similarity ε has not yet been taken into account. As shown in Figure 3-6 the use of the similarity ε could increase the accuracy even for smaller transform and aggregation windows. This effect will be explored in Section 3.3.2.3.

Figure 3-8 and Figure 3-9 show the average accuracy and recall for different aggregation window sizes on the KITTI 2015 dataset. Although the maximum accuracy (94.6%) is obtained by using a transform window of 17×17 and an aggregation window of 9×9 the difference between a transform window of 7×7 and an aggregation window of 9×9

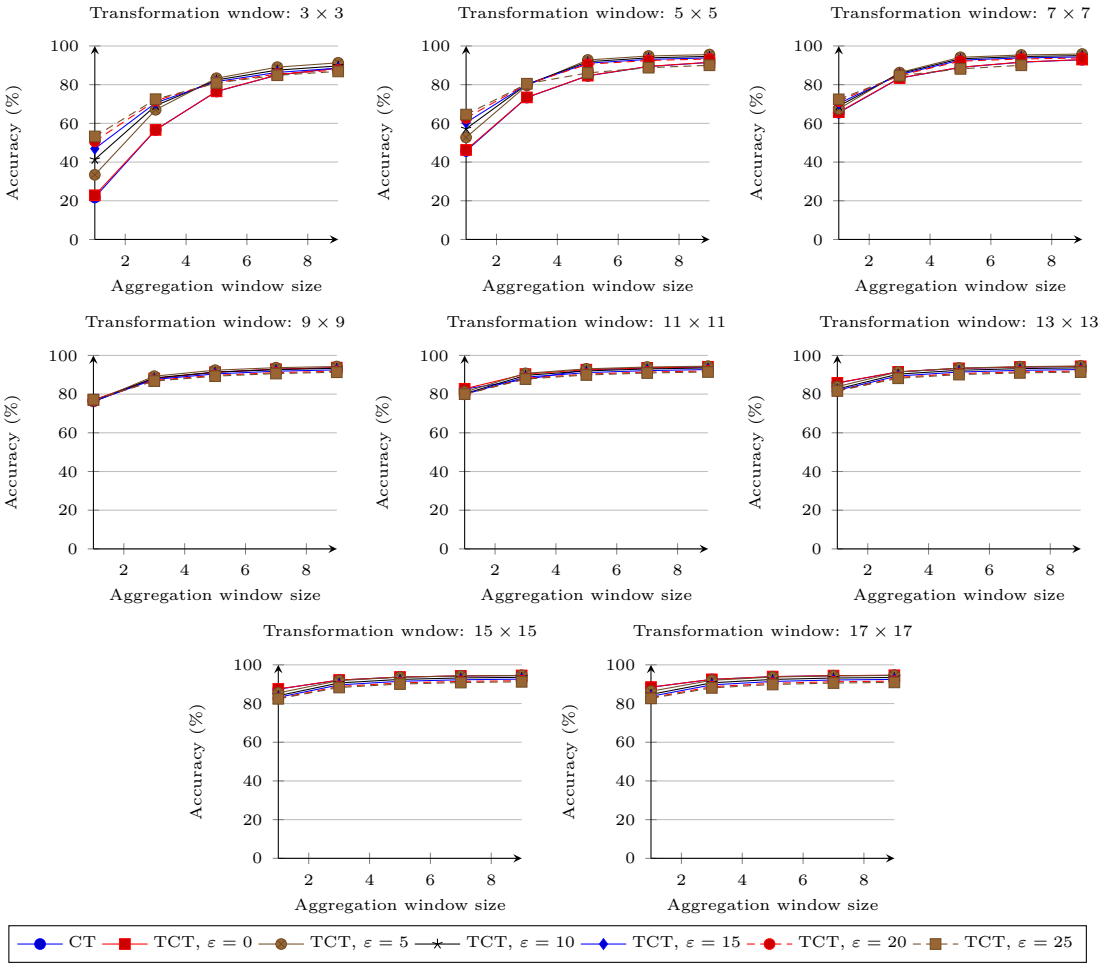


Figure 3-8: Effect of the aggregation window on the accuracy for the KITTI dataset using different window sizes for the image transform.

(accuracy of 92.9 %) is less than 2%, the latter being around 4 times faster. This could be explained by the fact that larger window sizes produce bit strings large enough for differentiating the pixels in the search range. All of this with a similarity $\varepsilon = 0$. The figure also shows that a similarity value could yield an increase in accuracy even for smaller window sizes which would require less computation. This effect will be explored in Section 3.3.2.3.

As for the Middlebury v3 dataset, large aggregation window sizes increase the recall, but low recall images still allow the identification of the contents by a human viewer. Therefore this is not a determinant in selecting the window size.

As one of the main goals of this thesis is to produce a system which is computationally efficient in an embedded system, it is highly important to use the smallest transform window and aggregation window possible without compromising the overall accuracy of

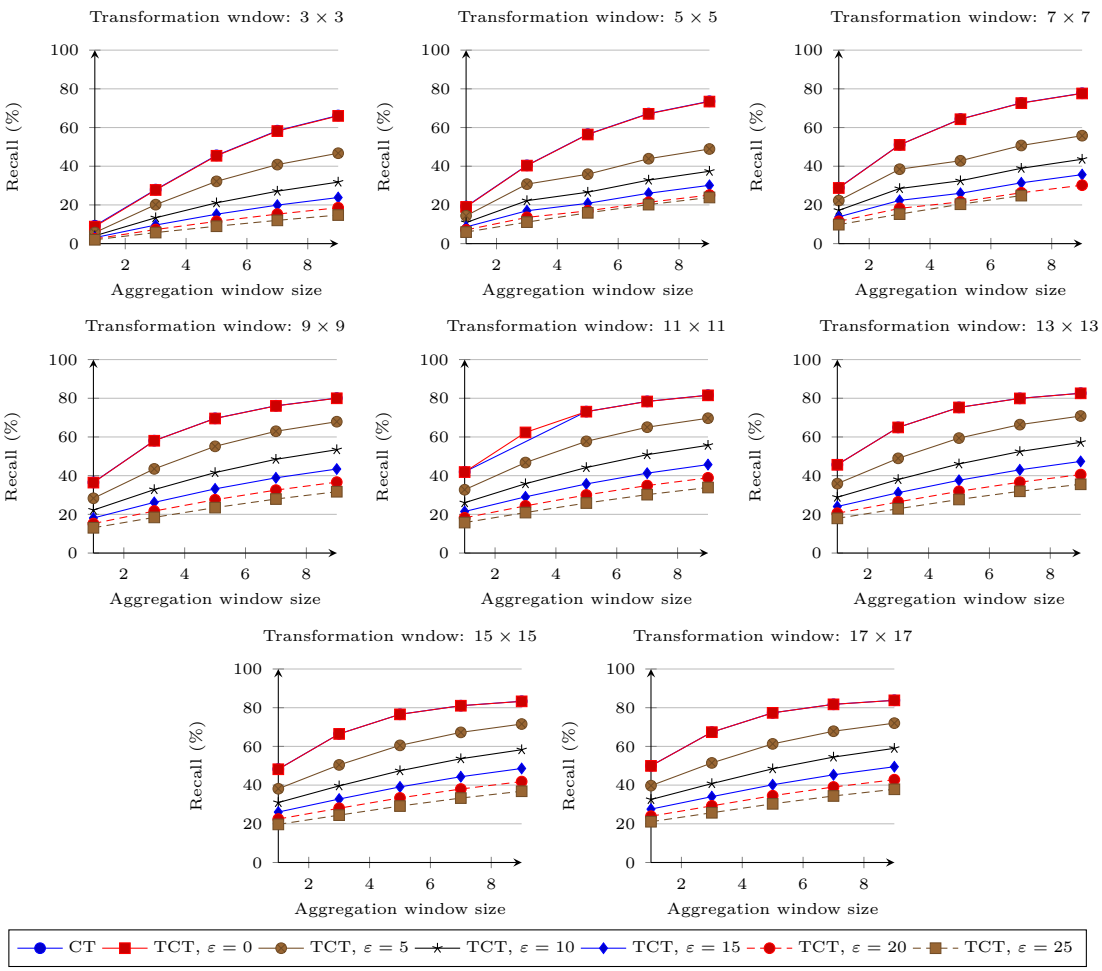


Figure 3-9: Effect of the aggregation window on the recall for the KITTI dataset using different window sizes for the image transform.

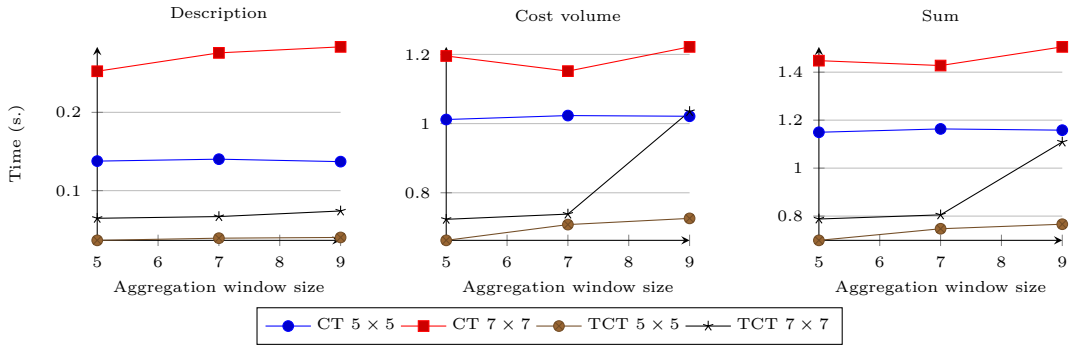


Figure 3-10: Running times for different transformation and aggregation windows of the TCT on the KITTI dataset.

the system.

Figure 3-10 shows the resulting running times for the calculation of descriptors and cost volume on the KITTI dataset. The time for performing the search of the best match is not included as this stage happens after the cost computation has been computed. After analyzing the effect of the window sizes, a transform window of 5×5 and aggregation window of 7×7 were selected due to the decrease in the computing time while losing only 5% accuracy in comparison to a transform window of 17×17 and aggregation window of 9×9. This combination of windows will be used for the remainder of this chapter.

Figure 3-10 shows that the calculation of the aggregation window does not impact greatly on the running time when small transformation sizes are used. Large windows increase the running time but they are not included in this plot as they do not produce a significant increase on the accuracy of the resulting disparity maps as detailed in Section 3.3.2.2.

3.3.2.3 Effect of the Similarity Threshold on the TCT

The similarity measure ε of the Thresholded Census Transform allows the detection of pixels with a high gradient. It is to be expected that the value of this threshold would affect the performance of the TCT and its edge detection capability.

In order to test the effect of the similarity threshold ε of the TCT, it was evaluated on the Middlebury v3 and KITTI 2015 datasets. The similarities ε shown are located in the interval $[0, 20]$. Similarity values larger than this ($\varepsilon > 20$) produced a lower accuracy and a drop on the recall. These images contained only few matched pixels making it impossible for a human to determine its contents. No confidence information is used in

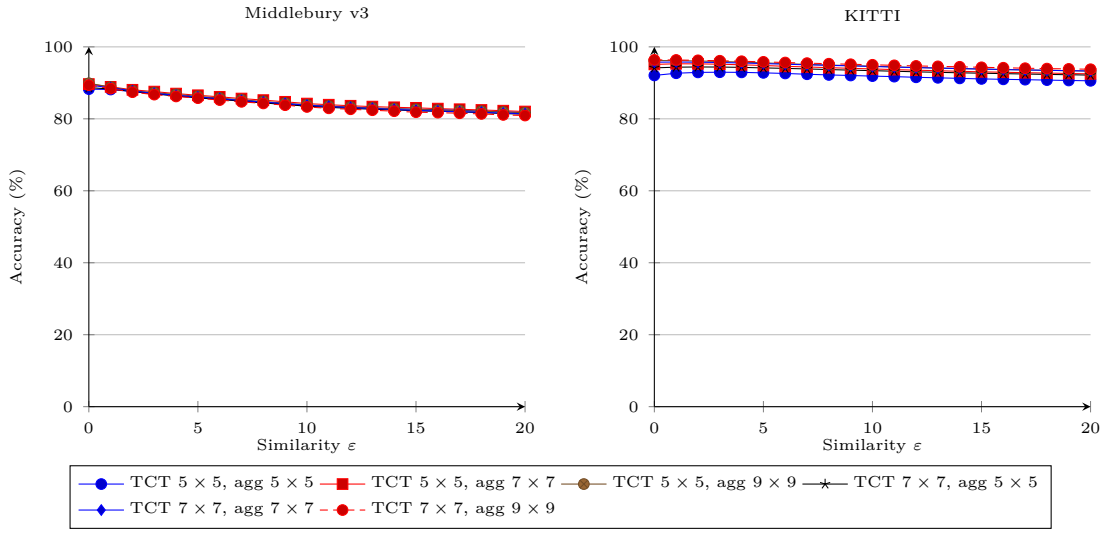


Figure 3-11: Effect of the similarity ε on the TCT.

this evaluation.

Figure 3-11 shows the effect of the similarity ε on the accuracy of the TCT for different transform and aggregation window sizes. The figure shows that use of the similarity ε can increase the accuracy by up to only 5% for the KITTI 2015 dataset. After analyzing the images it was found that erroneous pixels obtained with the Census Transform located on high texture areas are also erroneous for the TCT irrespective of the similarity value ε. In consequence, although the TCT is able to reduce the search space and produce accurate disparities for pixels located along the edges, the overall number of obtained disparities is decreased. As the number of erroneous pixels is kept almost constant for different similarity values ε the overall accuracy is decreased. Figure 3-12 shows this behaviour for one of the images in the KITTI 2015 dataset. The erroneous pixels (red) are the same for both images.

The maximum accuracy on the KITTI 2015 dataset is obtained by using a similarity $\varepsilon = 4$ for the tested transform and aggregation window sizes. For the Middlebury v3 this maximum accuracy is obtained by using a similarity of 1 which means the similarity value does not really produce an increase in accuracy with respect to the Census Transform. It is important to note that when $\varepsilon = 0$ the TCT produces the same bit-strings as the Census Transform. If the resulting bit-strings of the TCT are used as masks for $\varepsilon = 0$ the accuracy is only increased by less than 1% for $\varepsilon = 0$ with respect to the Census Transform.

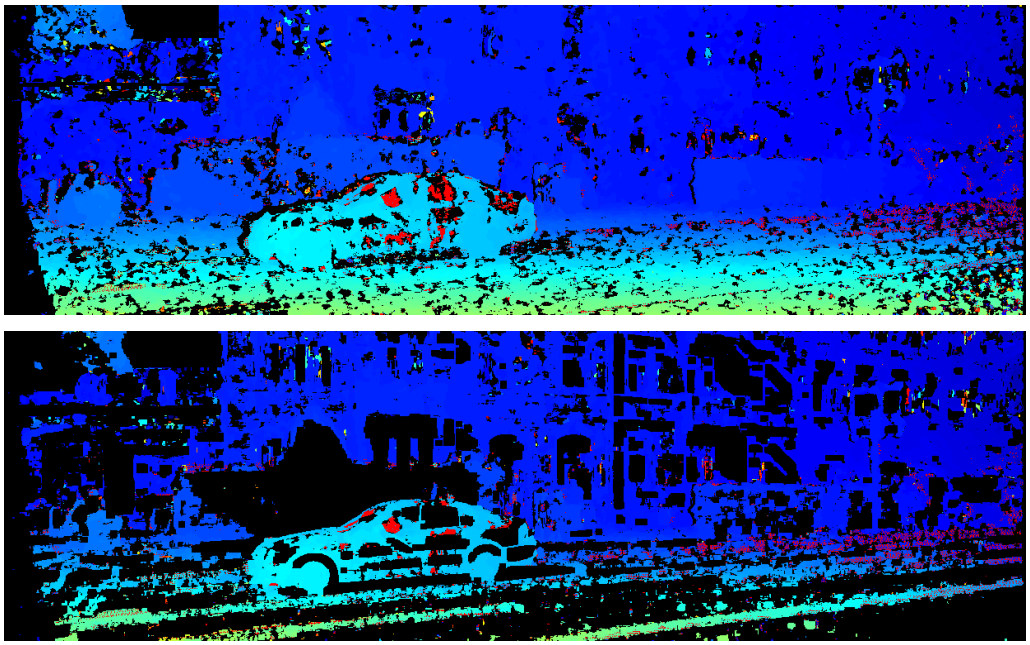


Figure 3-12: Sample disparity maps obtained by using the Census Transform (top) and TCT with $\varepsilon = 15$ (bottom). Although some regions with low texture were removed by the TCT, most of the erroneous pixels (red) in highly textured areas remain the same for both images.

3.3.2.4 Edge Detection Capability of the TCT

The similarity value of the TCT behaves as a threshold on the gradient of the pixels in the window as described in Section 3.2.1. This behaviour enables the detection of image edges without requiring a separate step for edge pixels identification. This is important because image edges allow representation of the image contents in a compact way. It is to be expected that an image has only a few edge pixels compared to the total number of pixels in the image. By using this edge information it is possible to reduce the computation load as only a few pixels would need to be matched.

In order to test the edge detection capability, the edges identified by the TCT are compared against two state-of-the-art edge detectors, the Canny edge detector [140] and Edge Drawing Parameter Free (EDPF) [141], [142]. The results are shown in Figures 3-13 and 3-14. The accuracy is computed as a ratio of the number of edge pixels detected by the TCT which are marked as edges by Canny or EDPF. Although it can be seen that recall is decreased with the similarity value ε, by analyzing the images it was found that the edges detected by the TCT are located in a distance of 2 pixels in average from the edges detected by Canny and EDPF. Figure 3-15 shows that nearly all the edges detected

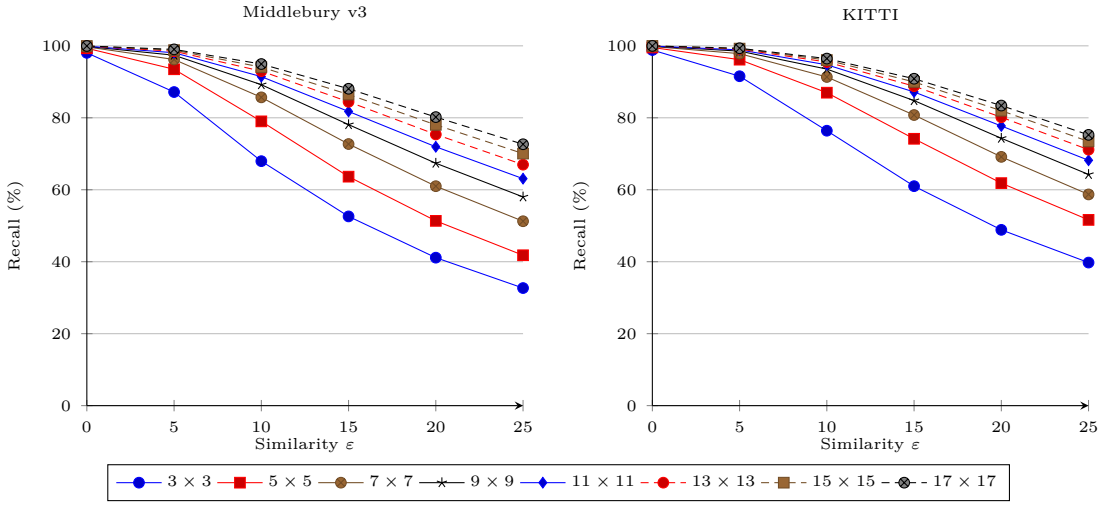


Figure 3-13: Canny edges detected by the TCT for different window sizes.

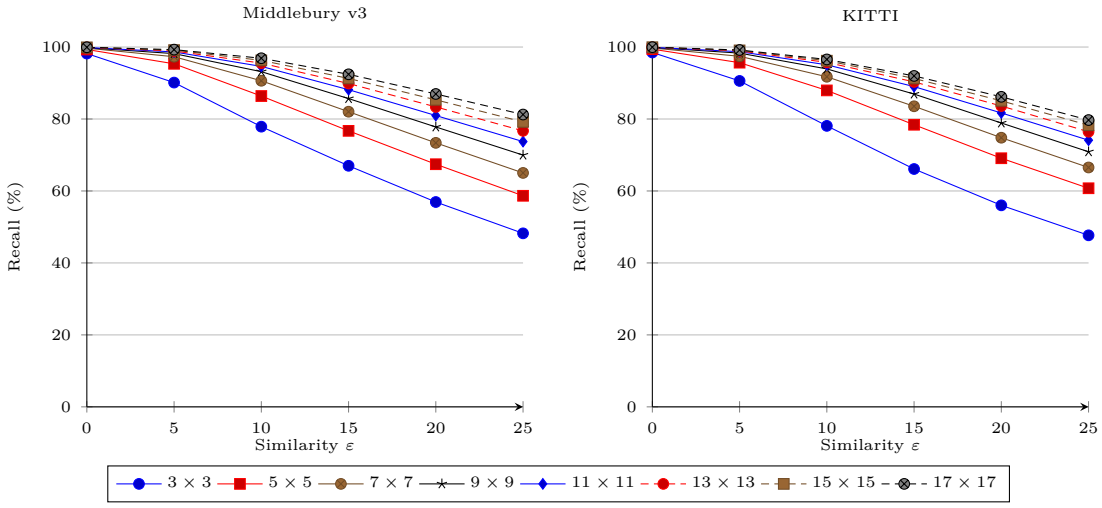


Figure 3-14: EDPF edges detected by the TCT for different window sizes.

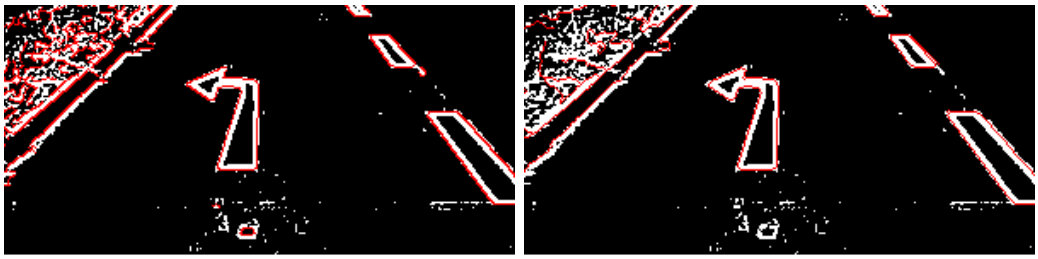


Figure 3-15: Pixels masked by the TCT using a similarity $\varepsilon = 15$. The pixels in red are the edges detected by Canny (left) and EDPF (right). Pixels in white are the pixels which obtain a TCT string with at least one value different from zero. Black pixels correspond to zero-strings on the TCT.

by the TCT are located within a close distance to the edges. As this thesis does not aim to create an edge detector but to use edges to reduce the search space on a stereo-matching approach no further processing is done to find the right location of the edge.

3.3.2.5 Analysis of the Effect of Confidence Measures

The last parameter to test in the stereo-matching approach is the threshold to apply to the match confidence. In order to test this, different values were explored in conjunction with the parameters of the CT and TCT identified above as producing the maximum accuracy. A transform window 5×5 and aggregation window of 7×7 are used for both the CT and the TCT. A similarity value $\varepsilon = 4$ is used for the TCT.

Figure 3-16 shows the effect of the confidence threshold applied to the Census Transform and the Thresholded Census Transform. The confidence values are in the interval $(0, 1]$. It can be seen that large confidence values produce a drop on the recall as only a few matches produce a confidence close to 1. For the remainder of this chapter, a confidence threshold of 0.1 is selected as it produced an increase in the accuracy close to 5% without producing a large drop in the recall.

3.3.3 The Complete Rank Transform

As the Complete Rank Transform was originally proposed for the calculation of optical flow no evaluation of its descriptive power is available in the stereo-matching context. Therefore as part of this thesis its behaviour is analysed using the same local stereo-matching approach described in Section 3.3.1. As the CRT produce descriptors with integer values in the range $[0 - mn]$ for a window size $m \times n$ the SAD dissimilarity metric

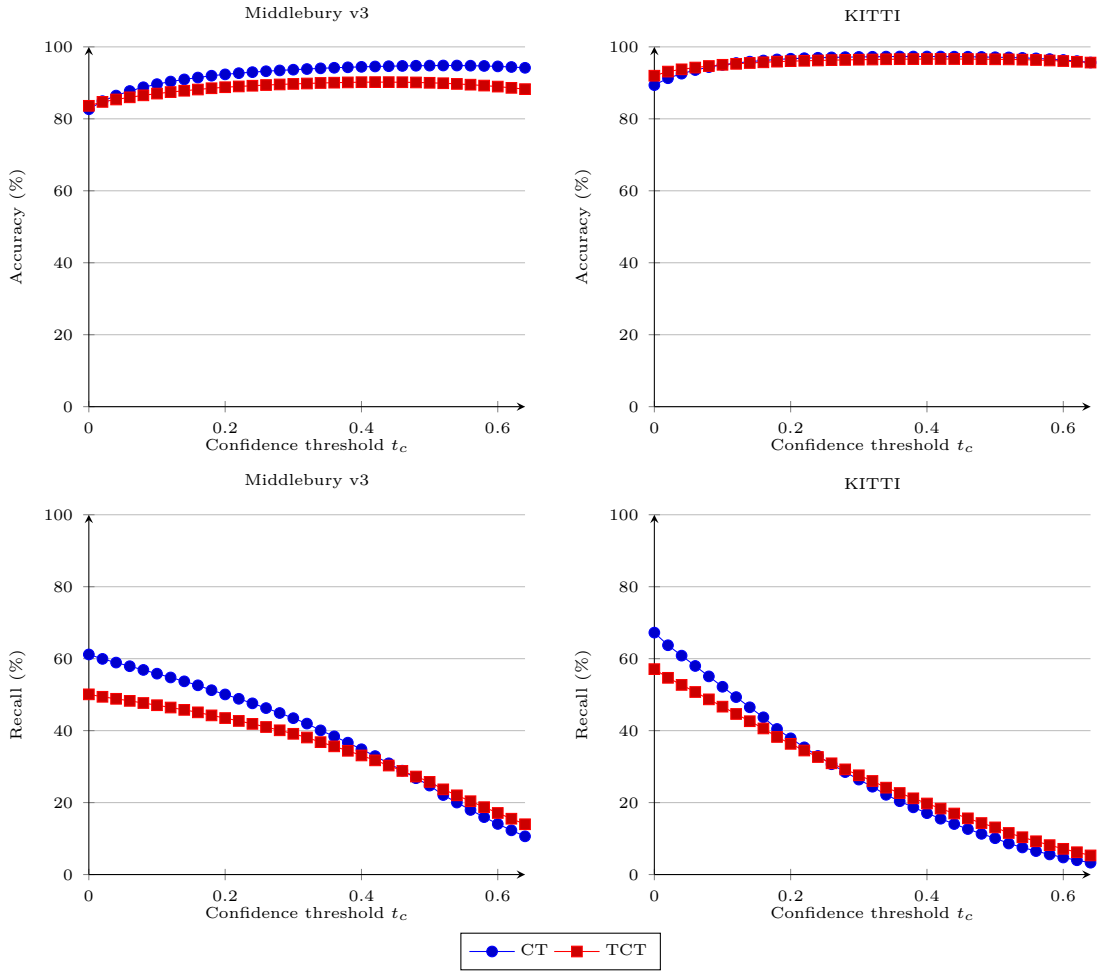


Figure 3-16: Effect of the confidence measure on the TCT and CT.

is used due to its computation speed.

First the effect of the window size is analysed. For this, the used window sizes $m \times n$ are set to be in the interval $[3, 30]$. Figure 3-17 shows the results of the evaluation of the window size on the Middlebury v3 and KITTI 2015 datasets. It can be seen that for a window size of 15×15 the accuracy and recall present a maximum for both datasets. Similar to the CT and TCT, larger windows lose accuracy due to the borders smoothing effect created by a large window.

The effect of the use of aggregation windows on the CRT is analysed in the same manner as for the TCT. Figure 3-18 shows the result of applying different aggregation window sizes along with different transform windows. No confidence information is used. This figure shows that an aggregation window of 3×3 along with a transform window of 11×11 produced results with an accuracy similar to larger transform windows on the Middlebury and KITTI datasets. This behaviour is also presented in the recall of obtained

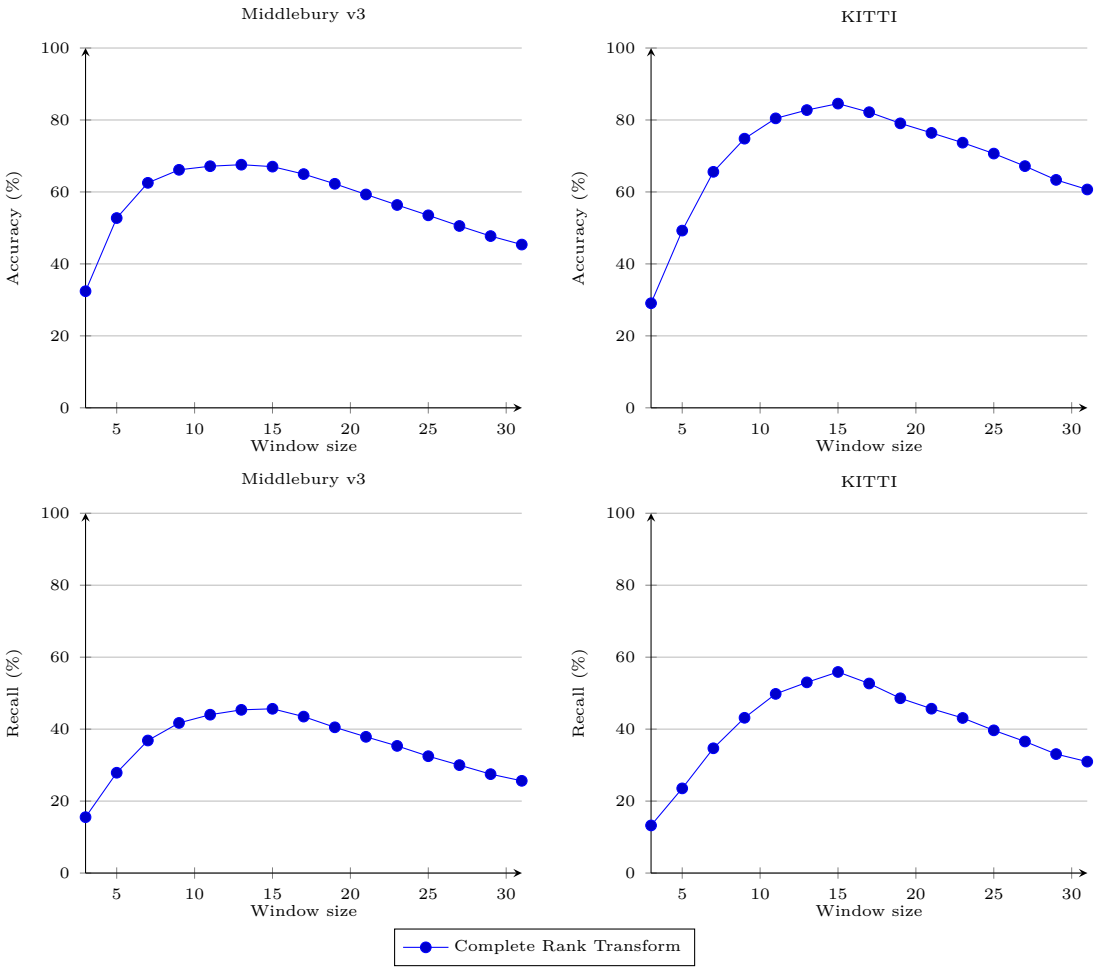


Figure 3-17: Effect of the transform window size on the accuracy of the CRT.

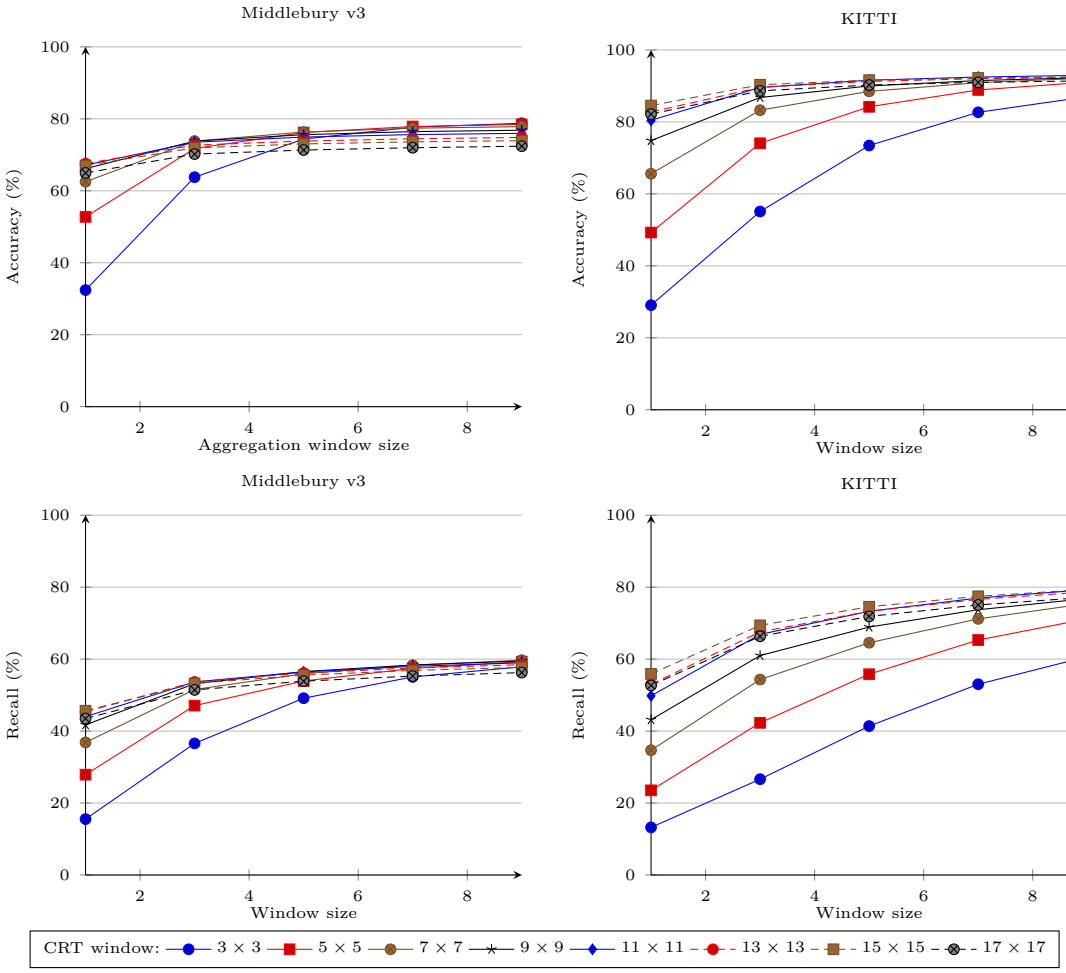


Figure 3-18: Effect of the aggregation window size on the accuracy of the CRT.

disparity maps. A small transform window would translate into fewer operations required for the transform computation. Therefore for the remainder of this thesis the combination of a transform window of 11×11 along with an aggregation window of 3×3 is used for disparity calculation.

As can be seen in Figure 3-19 the window size has a large effect on the computation time of the CRT. This is caused by the calculation of the rank of every pixel in the transform window. As detailed in Section 3.2.2 the estimation of the rank for one pixel requires to know the number of pixels with smaller intensity. Although this procedure could be speeded up by using counting sort [143], the resulting computation time limits its applications in real-time scenarios when a dense approach is used.

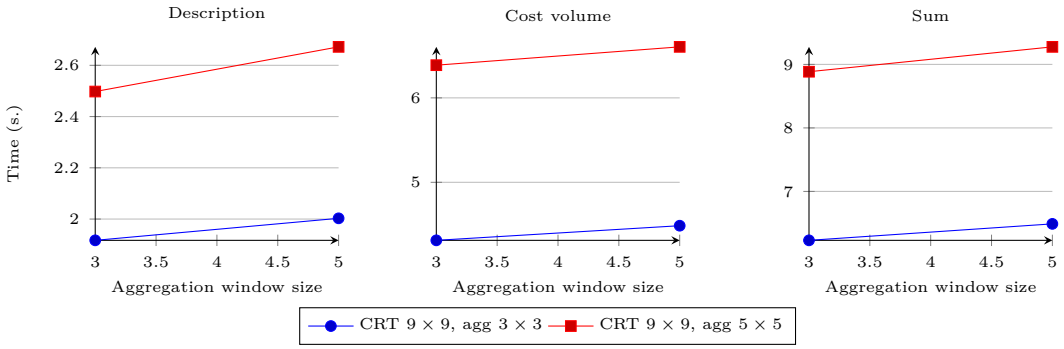


Figure 3-19: Timing on the calculation of the Complete Rank Transform descriptor.

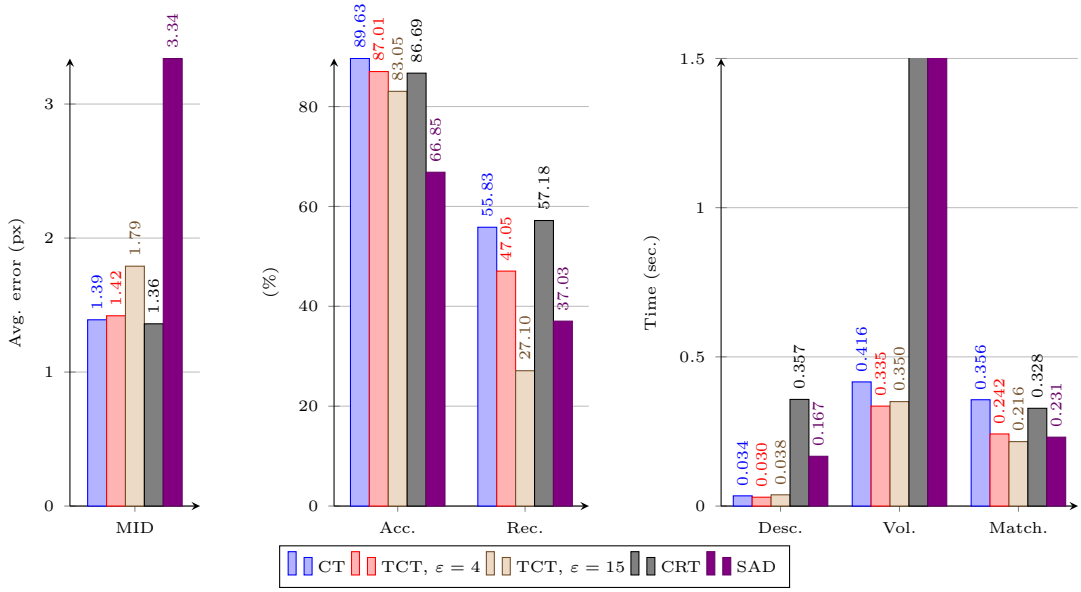


Figure 3-20: Comparison of the CT, TCT, CRT and intensity window on the Middlebury v3 dataset. The accuracy is computed for pixels with a disparity error ≤ 1.

3.3.4 Descriptors Comparison

After identifying the best parameter combination for each of the descriptors, the obtained disparity maps are compared. The average results on the Middlebury v3 and KITTI 2015 dataset are shown in Figure 3-20. The table shows the average end-point error which is the difference between the obtained disparities and the ones provided by the ground truth. The running time is the time required for calculating the descriptors, cost volume and matching. It can be seen that by using the TCT it is possible to obtain the same accuracy as in the CT whilst reducing the running time. This reduction in the running time is due to the masking of zero bit-strings which correspond to uniform areas.

Figure 3-22 shows sample disparity images obtained for the KITTI 2015 dataset using each of the compared pixel descriptors. These images show that by using the TCT the

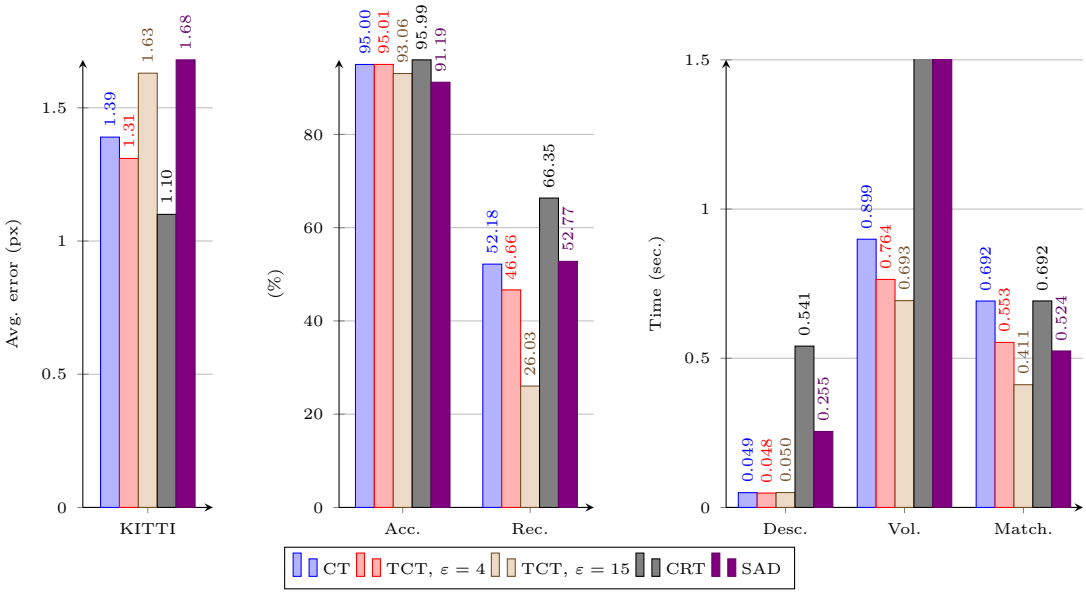


Figure 3-21: Comparison of the CT, TCT, CRT and intensity window on the KITTI 2015 dataset. The accuracy is computed for pixels with a disparity error ≤ 3.

obtained disparity maps only have values for pixels located around image edges or in highly textured areas. This characteristic is desired for systems based on edge information. Although the disparity maps obtained by the TCT have lower density when compared to the ones obtained by the CT they contain only information around image edges and highly textured images, whereas it is impossible to discern the location of the obtained disparities on the CT without using further information. Although the accuracy is decreased for large similarity values, the TCT disparity maps contain mostly pixels around the image edges ignoring texture in the image. The sample image in Figure 3-22 shows that a human would be able to distinguish the image contents of the disparity image obtained by the TCT even though it is less dense when compared to the CT. As this thesis is focused on the use of edge-based disparity maps this is a desirable result.

The CRT can produce highly dense regions in the inner areas of the objects. This is due to the low number of changes in the ordering of the pixels on the object surfaces. Although recall is high, the time required for calculating the disparity maps is also high, as a large number of computation windows are required. Large transform windows require sorting all of the pixels in the window to calculate the rank of each of them. This operation takes a long time to compute. Additionally, large windows produce long strings increasing the cost computation time. This long computation time limits its application in real-time systems for dense approaches.

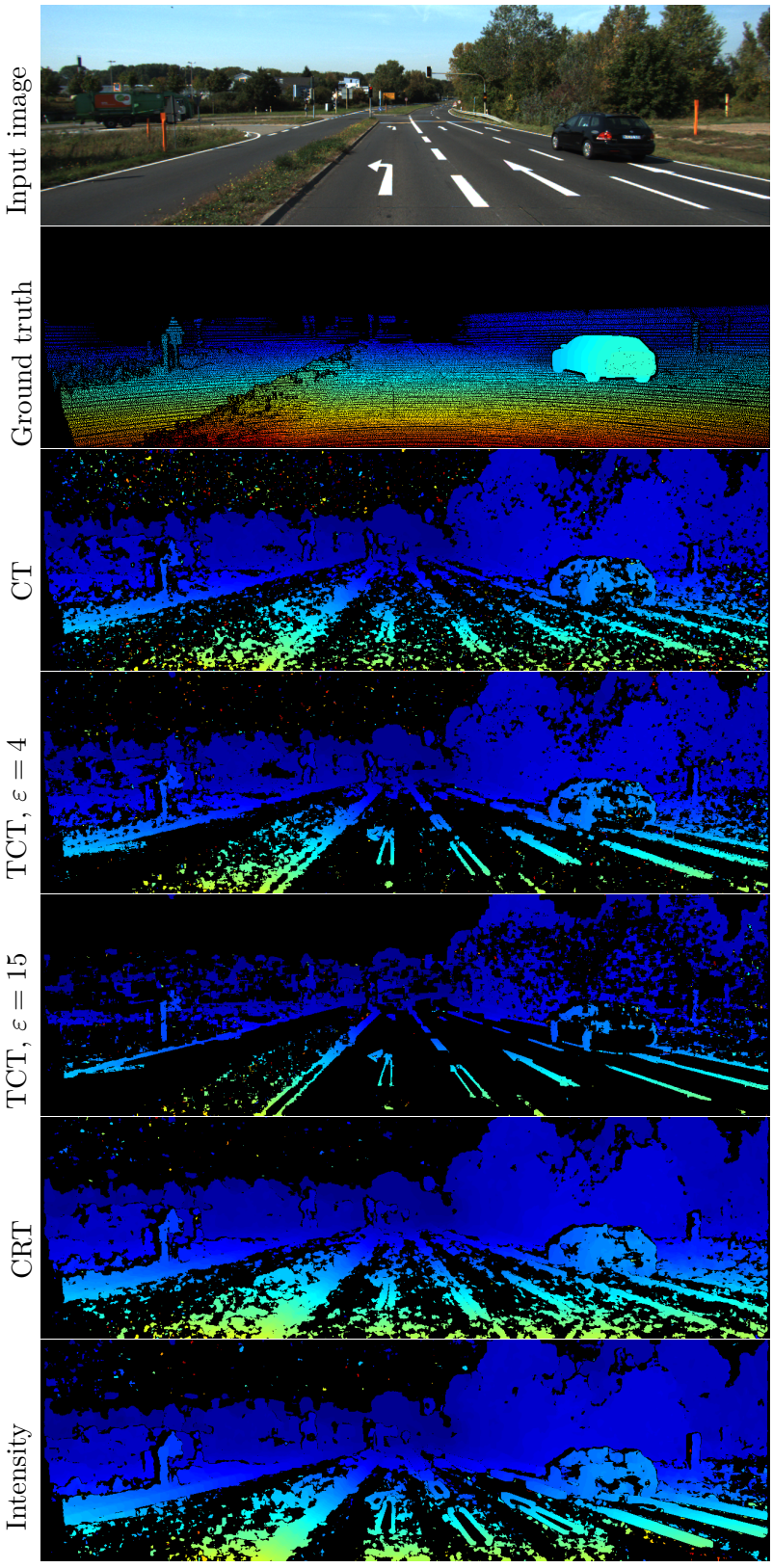


Figure 3-22: Sample images from the KITTI 2015 dataset.

An intensity window of size 11×11 with aggregation window of size 3×3 and SAD cost is included for reference, showing the lowest accuracy of the comparison with a large running time. This points to the efficiency of binary-based representations, namely the CT and TCT.

3.4 Discussion

This chapter introduced a new pixel descriptor and evaluated its performance on the Middlebury and KITTI datasets. Additionally the Complete Rank Transform was evaluated for the first time in the context of the stereo-matching problem using a dense area-based approach. The proposed descriptor, the Thresholded Census Transform inherits the robustness to lighting conditions from the Census Transform and additionally it is able to identify areas of low texture without additional processing.

As was shown in the previous sections, the TCT succeeded in reducing the overall matching time for a local stereo-matching approach. This reduction in the matching time was obtained by avoiding the computation of cost values for low texture regions by reducing the search space at the matching stage without a significant loss in accuracy. Additionally, the obtained disparity maps contain information only from pixels which are located around image edges.

The Complete Rank Transform showed a higher accuracy and recall than the CT and TCT at the cost of an increased computation time. As the first step in a local stereo-matching approach is to compute the descriptors for all of the pixels in the image, the increased computation from the CRT limits its application in real-time systems for local stereo-matching approaches.

Further research on the TCT would include the use of an adaptive similarity threshold ε according to the image contents. Also, it would be interesting to identify the effect of sparse and non-square windows like the ones proposed in [44]. Although they may increase the resulting accuracy, its study is out of the scope of this thesis. Additional research could focus on converting the TCT into an edge detector.

Although the TCT showed a reduced computation time compared to the Census Transform, the speed up of the local stereo-matching was not as impressive as expected. Therefore a different approach for stereo-matching is presented in Chapter 4.

4

Disparity by Simultaneous Edge Drawing

4.1 Introduction

Image edges are good candidates for extracting information from a stereo-camera for two main reasons: (a) typical images usually contain only a small number of edge pixels, e.g. image pixels in an outdoors scene typically represent less than 20% of the image pixels; (b) edges contain sufficient information to represent the image semantics [130], [16], [131]. These two properties are desired in a low-resources stereo-matching approach, as a small number of pixels translates into a small number of candidates at the matching stage.

As presented in Chapter 3 using edge pixels in a dense approach reduced the required number of computations. It is expected that by using an approach which relies highly on edge pixels the number of computations could be decreased. Therefore this chapter introduces a new method for low-level disparity estimation. This method produces sparse disparity maps based on image edges. The method works by extending the principles of Edge Drawing (ED), a fast and accurate edge detector. The extended version produces disparities for edge pixels at almost the same cost as running the edge detector on both images independently. This results in fast and accurate disparity values which additionally include connectivity information, a characteristic which could be used in higher level scene recognition algorithms. Figure 4-1 shows the areas of the stereo-matching process where the proposed approach makes a contribution.

It is important to note that the edges used in this thesis are not restricted to any particular shape. This allows the representation of a large number of objects, in contrast to approaches based on straight lines which can represent only a few of the structures

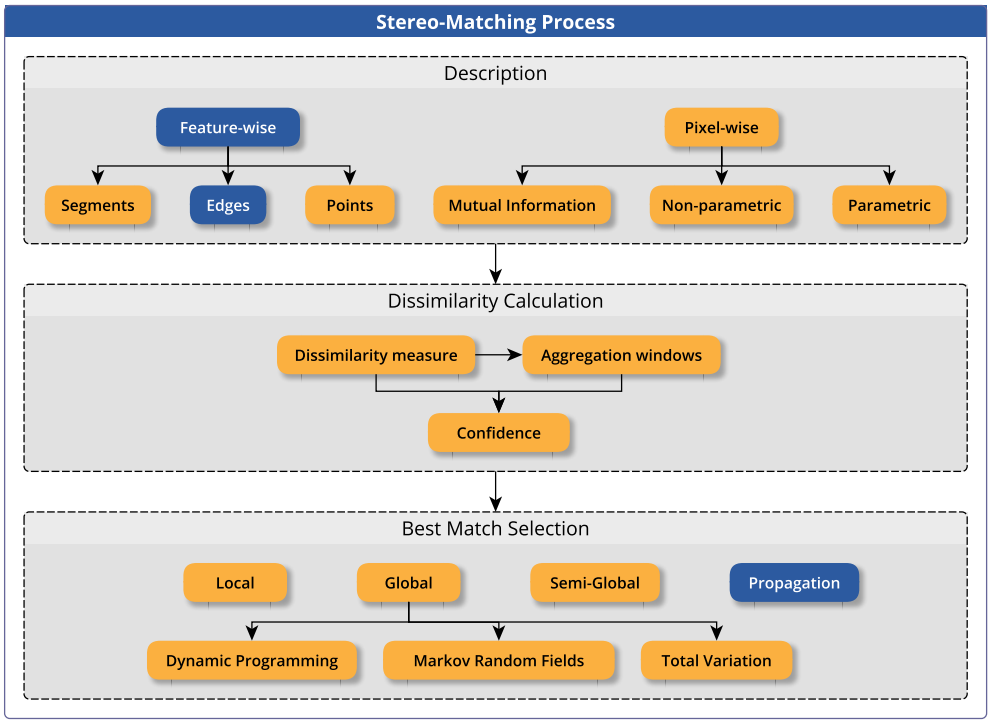


Figure 4-1: Areas of the stereo-matching process contributed (blue) by the proposed approach. SED changes the way the edge-features are used for stereo-matching by propagating the disparities while the edges are detected.

found in a typical outdoor scene like [16], [21], [18], [144], [51].

This chapter is structured in the following way: first the edge detector ED is described as it is used as a base for the proposed algorithm. Then the proposed disparity extractor is presented. After this the disparity extractor is evaluated on the Middlebury v3 and KITTI 2015 datasets and the results of the evaluation are presented and compared with other state-of-the-art methods for disparity extraction based on edges.

4.2 Edge Detection by Edge Drawing

After reviewing the literature on approaches for using edges for stereo-matching and edge detection, only one was found to "navigate" along the image gradient to create ordered chains of consecutive edge-pixels. The principle of this approach seemed suitable for the propagation of disparities along the image edges. This edge detector is Edge Drawing (ED) proposed by Topal, Akinlar, and Genc [142], [145]. Although other edge detectors like Canny [140] are able to detect the image edges accurately and quickly, they tend to produce non-connected edge pixels [142] and therefore they are not suitable for disparity

propagation.

The principle behind the Edge Drawing (ED) edge detector is to imitate the behaviour of a child joining the dots in an image [142], [145]. This edge detector has four main stages: noise-removal, computation of image gradient magnitude and direction, identification of anchor points and a smart routing procedure for performing the edge localization and linking.

The noise removal step is common to most image processing approaches. It convolves the image with a kernel in order to blur the noisy pixels. Topal, Akinlar, and Genc [145] used a Gaussian kernel of size 5×5 with $\sigma = 1$ for performing the noise removal. Although other noise removal approaches exist in the literature, this is commonly used due to its high speed and low computational complexity.

The image gradient provides crucial information for the detection of the image edges. Topal, Akinlar, and Genc showed that ED was able to perform well using any gradient operator, i.e. Sobel, Prewitt, Scharr, etc. and selected Prewitt due to its low complexity [145]. The direction of the edges is set according to the gradient orientation. If the gradient $G(x, y)$ is horizontal, it means that a vertical edge passes through pixel (x, y), and vice versa.

The anchor points are defined as pixels with a high probability of being edge pixels. As pointed out by Topal and Akinlar, approaches for feature point finding could be used for extracting the anchors but they are computationally complex. Instead, non-maxima suppression produced good anchor points in an efficient way in [142]. Topal, Akinlar, and Genc used non-maxima suppression in the direction of the gradient to identify the peaks, i.e. if a pixel has a gradient higher than both of its neighbours it is marked as an anchor. The anchor points are seeds for starting the smart routing procedure, in order to decrease the number of seeds, the image is scanned at intervals s i.e. every one out of s lines is processed; as it is expected that multiple anchor points would be found on each edge.

After the anchor pixels have been identified, the Edge Drawing algorithm applies a smart routing procedure in order to connect the anchor points and obtain the chains of edge pixels. This smart routing procedure moves from an anchor point using the gradient information until the next anchor point in the edge is reached. This is performed until there are no more candidate pixels or a previously linked pixel is found. The smart routing procedure is illustrated in Algorithm 4.1 taken from [142]. Starting at the location (x, y)

the neighbours on the left $(x-1, y-1)$, $(x-1, y)$ and $(x-1, y+1)$ are tested and the one with the maximum gradient is picked. A similar approach is applied for the right, up and down directions. Only the tested neighbours are updated, i.e. if the linking is on the right the tested neighbours are $(x+1, y+1)$, $(x+1, y)$ and $(x+1, y-1)$.

Algorithm 4.1 Linking on the left of an anchor at (x, y). [142]

Symbols used:
(x, y): Processed pixel
G: Gradient map
D: Edge direction map
E: Edge map

```
function GO_LEFT(x,y,G,D, E)
    while (x, y) is in the gradient map, not linked and is horizontal do
        Mark E(x, y) as linked
        if G(x − 1, y − 1) > G(x − 1, y) and
            G(x − 1, y − 1) > G(x − 1, y + 1) then
            x ← x − 1
            y ← y − 1                                   ▷ Up-left
        else if G(x − 1, y + 1) > G(x − 1, y) and
            G(x − 1, y + 1) > G(x − 1, y − 1) then
            x ← x − 1
            y ← y + 1                                   ▷ Down-left
        else
            x ← x − 1                                   ▷ Straight left
        end if
    end while
end function
```

The smart routing procedure produces one-pixel wide continuous chains of pixels which correspond to the image edges. This chain of pixels is obtained without requiring any further processing, resulting in a very efficient and accurate approach for edge detection. This characteristic is exploited in the current thesis to propagate the disparity values across the stereo pair.

4.3 Related Work

The efficiency of the use of image edges for estimating disparity maps has been attractive to different researchers as shown in Chapter 2. The approaches for edge-based stereo-matching can be classified according to the geometry of the edge into: straight line based and curve based.

Straight line based approaches assume the edges in the image could be described by using only straight lines. When a curve is found by these approaches, the curve is

divided into segments fitted by straight lines. Although edges of this kind have been used successfully to calculate disparity maps for indoor scenes [21], [19], outdoor scenarios contain only a small number of straight lines resulting in disparity maps unable to represent the image semantics as can be seen in the results from [144], [51], [16] and [18].

Curves on the other hand are able to fully represent the image semantics at the cost of an increased complexity in the matching. Approaches used for matching curves can be divided into edge-segment matching, edge-pixel matching and 3D fitting. Edge segment matching approaches obtain a descriptor for the whole edge or sub-segments, then they are compared across the stereo-images. Edge-pixel matching approaches calculate a descriptor for every edge-pixel and then the edge pixels are matched individually. 3D fitting approaches use information from the cameras to obtain a 3D relationship between the curves, then the re-projection error is minimized across the stereo-pair.

Edge segment matching approaches are sensitive to the results of the edge detector used, as seen in [49], one edge in the left image could result in multiple edges on the right image of the stereo-images. This could be a problem for the calculation of the disparity of each edge pixel and result in ambiguous matches.

Edge-pixel matching approaches have proven to be successful in calculating disparity maps. These approaches are similar to pixel-based stereo-matching approaches. A descriptor is computed for every edge pixel and then it is matched across the stereo pair using a local approach or Dynamic Programming [61], [28], [54], [146], [146], [55]. These approaches then use the edge connectivity information to identify wrong matches and identify edges created by object boundaries. As the matching is performed across the scan-line, these processes are usually fast and require low resources.

3D fitting approaches have proven to be successful at accurately estimating the geometry of the 3D objects, which produce the edges, but they are sensitive to the location of the end-points [147], [17] resulting in only few matched edges. Approaches which are able to add robustness to the location of the end-points are computationally complex [22], [57] and therefore slow to compute.

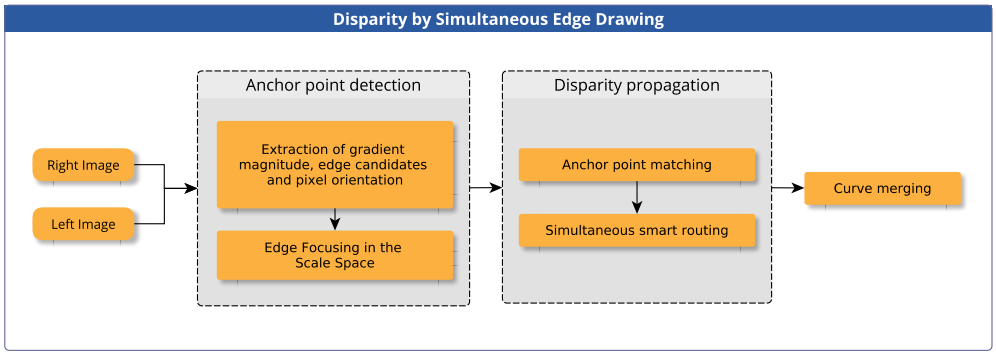


Figure 4-2: Pipeline of the Disparity by Simultaneous Edge Drawing algorithm.

4.4 Proposed Disparity Estimator

As the goal of this thesis is to produce disparity maps which can be implemented on an embedded platform the approach used must be low in resource consumption. Image edges are able to fully represent the image semantics, therefore the state-of-the-art was reviewed to identify the most suitable approaches in an embedded environment. Although different approaches for matching edges are available, they tend to be computationally expensive or able to match only a subset of the image edges resulting in edges which do not fully represent the image semantics. This suggests the need for a new disparity extractor which is able to produce fast and accurate edge-based disparity maps using low resources while still being able to represent the image semantics.

The proposed disparity estimator, which is named Simultaneous Edge Drawing (SED), extends the Edge Drawing algorithm to run simultaneously across the stereo pair. By doing this, there is no limitation imposed on the shape of the obtained edges. Therefore it is suitable for outdoor scenarios that may present any kind of geometry e.g. trees and people. The resulting algorithm allows the estimation of the disparity for edge-pixels at the cost of running the edge detector over both images in the stereo-camera and matching a small number of pixels. Figure 4-2 shows a schematic diagram of the proposed algorithm.

The Edge Drawing edge detector works by linking pixels starting on an anchor point. The proposed extension to the Edge Drawing algorithm is as follows:

1. Perform the matching of the anchor points across the stereo-images using an area based approach.

2. Modify the smart routing from ED to link the pixels simultaneously across the

stereo-images.

Although simple, this approach will produce accurate disparities under the following assumptions:

(i) Only one edge passes through each anchor point.

(ii) If two anchor points match, the edges passing through the matched anchor points match.

(iii) The linking procedure follows the same edge at all the times.

(iv) The edges in the left and right images have the same number of pixels.

Assumption (i) is used in ED and is enforced by stopping the smart routing when a pixel already linked is found. In this way, each edge-pixel could appear in at most one edge. As anchor points are part of the edge, they are marked as linked, therefore they cannot appear in more than one edge.

Assumption (ii) is enforced by assumption (i), as only one edge is allowed per anchor point, the only way an edge could pass through an anchor point pair is if they are a match too.

Assumption (iii) is not enforced explicitly and its effect on the resulting performance is analysed in the evaluation section.

Assumption (iv) might not be true in most of the cases, especially on occluded edges and edges with an angle close to zero. In order to avoid breaking it, the stopping criteria breaks the linking when a pair of linked edges start to differ.

The proposed algorithm for disparity estimation by Simultaneous Edge Drawing has the following stages: computation of the image gradients, edge orientation and edge candidates map; identification of anchor pixels; anchor matching; simultaneous smart routing; curve merging and length validation.

4.4.1 Computation of Image Gradients, Edge Orientation and Edge Candidate Map

The image gradient is used for guiding the linking, it expresses the amount of change in intensity horizontally or vertically. Pixels whose horizontal gradient is larger than the

vertical gradient correspond to vertical edges, and vice versa. The edge candidate map are pixels which are candidate to become an edge.

As in ED, blurring is applied to the input images in order to remove the noise. It is applied by convolving the stereo-images with a Gaussian kernel with $\sigma = 1$ as it produces good results in [145].

The computation of the image gradients G_x and G_y is performed by convolving the images with Prewitt kernels P_x and P_y as in ED, i.e. $G_x = P_x * I$ and $G_y = P_y * I$ where I is the input image. This is calculated on the left and right image of the stereo-images. The Prewitt kernels are defined as:

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } \quad P_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{4.1}$$

The gradient magnitude G is computed by using:

$$G = \sqrt{G_x^2 + G_y^2} \tag{4.2}$$

The edge orientation H at pixel p is computed by comparing the vertical and horizontal gradients G_y and G_x:

$$H = \begin{cases} 1 & G_x < G_y \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

This produces a value of 1 for horizontal edges and 0 for vertical edges.

The edge candidate map E indicates the pixels which are candidate to become an edge. It is computed by applying a threshold to the gradient magnitude for each pixel and for each image in the stereo-images. The threshold is taken from [141], where only gradient values obtained by quantization errors are removed. For the Prewitt kernel this occurs at $t_g = 8.48$.

The Prewitt gradient could produce non-integer values, therefore in order to avoid floating point operations, the gradient value is scaled to the range $[0, 255]$. This is performed after the edge orientations and edge-map have been computed. Their computation is not affected by the scaling.

4.4.2 Identification of Anchor Points

Anchor points are pixels highly likely to be part of an edge, they are used for matching and to start the simultaneous smart routing. After eliminating edge candidates which might be produced by quantization error, the number of remaining candidates is large. In order to identify robust anchor points, a Gaussian Scale Space is used for identifying edge candidates present over different scales. The Edge Focusing algorithm presented by Bergholm [148] is able to track edge pixels through the Gaussian Scale Space, therefore it is used in this thesis for identifying the anchor points. The Edge Focusing algorithm illustrated in Algorithm 4.2. A value of $\sigma_{MAX} = 4$ is taken as suggested in [148]. Larger values would introduce an additional computational load due to the Gaussian blurring and a loss in the number of recovered candidates.

Algorithm 4.2 The Edge Focusing algorithm.

Symbols used:
I: Input image
σ_{MAX}

function EDGE_FOCUSING(I, σ_{MAX})
  $\sigma = \sigma_{MAX}$
  while $\sigma >= 1$ **do**
    Apply Gaussian blurring to I using σ.
    Compute the pixels gradient and edge orientation using the blurred image.
    Apply Non-Maxima suppression in the direction of the gradient.
    $\sigma \leftarrow \sigma - 0.5$
  end while
end function

As in ED a scan interval s is used to reduce the number of processed anchor points resulting in a reduction in the number of required computations. The scale interval means that the image is scanned at every s rows, i.e. for $s = 1$ every row is processed, for $s = 2$ only one of every two rows is processed and so on.

Once the anchor points have been identified, they are sorted in descending order according to the gradient magnitude as in ED [141] using counting sort [149].

4.4.3 Anchor Point Matching

In order to match the anchor points across the stereo-images, a local stereo-matching approach is used. The probability that an image edge corresponds to an object boundary

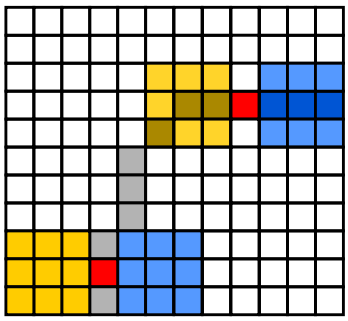


Figure 4-3: Location of the windows used for calculating the descriptor for an anchor a (red). The yellow area represents the location of the window used to compute $\xi_1(a)$ and the blue area represents the location of the pixels used to compute $\xi_2(a)$.

is high as weak edges created by texture are removed by the Scale Space [148].

It is important to note that the anchor point matching and the simultaneous smart routing process are performed interactively i.e. one non-linked anchor point is taken and matched and then the simultaneous smart routing is started. Only after the simultaneous smart routing has been completed the next anchor point is matched. As it is highly likely that one edge contains multiple anchor points, these anchor points are linked during the simultaneous smart routing and get a disparity value. By doing this, the number of anchor points required to be matched is decreased.

By assuming all the edges are located on image boundaries two descriptors $\xi_1(a)$ and $\xi_2(a)$ are computed for every anchor point a. Figure 4-3 shows the location of descriptors $\xi_1(a)$ and $\xi_2(a)$. Descriptor $\xi_1(a)$ is located at the left of the anchor pixel a (irrespective of the orientation of the anchor) at a distance d_a which guarantees that no pixel on the other side of the edge is taken into account by the descriptor. Descriptor $\xi_2(a)$ is located on the right of a at distance d_a using the same principle. As different pixel descriptors are available in the literature, a careful selection is presented in Section 4.5.1. Pixel descriptors are computed on the input stereo-images without any processing.

The dissimilarity between descriptors is computed by using one of the measures presented in Section 2.2.2. As two descriptors are available per anchor point, a local approach (see Section 2.2.3.1) is used for matching descriptors $\xi_1(a)$ and $\xi_2(a)$ independently. Then the the minimum value of the dissimilarity of both descriptors is taken as the match cost. Additionally, the orientation of the anchor points is used to reduce the number of candidates, i.e. for a vertical anchor point, only vertical anchors are taken as candidates. A similar approach is applied for horizontal anchors.

Two confidence metrics are applied to identify spurious matches. The inverse β_{PKR} confidence metric, Equation (3.7) is used for removing matches which might be caused by repetitive structures. Additionally the maximum attainable cost ratio β_{MAX} is used for eliminating matches which present a high cost value.

The maximum attainable cost ratio β_{MAX} is defined as a ratio between the minimum cost (dissimilarity) found among the anchor candidates and the maximum cost (dissimilarity) for a given descriptor. This is expressed as:

$$\beta_{MAX} = 1 - \frac{c_1}{c_{MAX}} \tag{4.4}$$

where MAX correspond to the maximum dissimilarity which could be obtained by the descriptor. For example, if a 3×3 Census Transform is used as descriptor the maximum dissimilarity occurs when the descriptors 00000000 and 11111111 are found. This would result in a Hamming distance of 8, therefore $c_{MAX} = 8$;

An LR-consistency check is applied as seen in Section 2.2.2 for identifying occluded anchor points. By assumption ii, occluded anchor points would correspond to occluded or partially occluded edges and therefore no simultaneous smart routing is performed.

4.4.4 Simultaneous Smart Routing

The anchor point matching provides a seed for the disparity of two matched edges across the stereo-images while the smart routing used in ED is extended to propagate this disparity along the edge pixels moving one pixel-pair at a time.

The smart routing from ED is an iterative procedure which takes as input an image pixel and then it navigates over the image gradient to the next pixel on the same edge. By doing this, it produces as output a chain of connected pixels belonging to the same edge. Resulting edges are one pixel wide and well localized, as shown in [145], [142], [141].

In this thesis, the smart routing in ED is modified so as to link pixels across the stereo-images. It takes as input a pixel pair and then by using the same principle from ED it moves one pixel in each image of the stereo-images. The modified version of the smart routing has the following stopping criteria:

(i) The next pixel to be linked in either of the images in the stereo pair is not in the edge candidate maps E^l and E^r for the left and right image respectively, i.e. the

thresholded gradient of the next pixel is zero.

(ii) A previously linked pixel is found.

(iii) The difference in disparity between a pair of consecutive linked pairs is larger than a threshold t_d.

(iv) The difference on the y location of the pixels is larger than an epipolar threshold t_e.

Stopping criteria (i) and (ii) are inherited from ED and verify against the edge candidate maps E^l and E^r and test against previously linked pixels. They avoid the addition of pixels which are not edge candidates and the inclusion of one pixel into multiple edges (assumption (ii)).

The simultaneous smart routing procedure follows edges one pixel at a time in a stereo pair without any knowledge about the relationship of the followed edges. This would result in the possibility of following wrong edges at edge crossings or when the wrong anchors are matched. In order to avoid this the stopping criteria (iii) checks for the following condition to be met:

$$|d_c - d_a| < t_d \tag{4.5}$$

where d_c is the disparity at the pair of pixels being linked and d_a is the disparity of the matched anchors used as seed for the smart routing. This ensures that the edge is located in the same disparity region as the matched anchors. In the case that an object spans several disparity regions the missing edge pixels would be recovered by using another pair of matched anchors as seed on a new run of the simultaneous smart routing.

In order to account for errors in image rectification and in the localization of the edges, stopping criteria (iv) allows an epipolar misalignment up to threshold t_e:

$$|y_c^l - y_c^r| < t_e \tag{4.6}$$

where y_c^l corresponds to the vertical coordinate of the pixel p_c^l being linked on the left image and y_c^r corresponds to the vertical coordinate of the pixel p_c^r being linked on the right image.

As in [145], [142], [141] it is not clear how to handle the changes in the orientation of the edges in the smart routing procedure, Algorithm 4.3 describes the criteria applied in

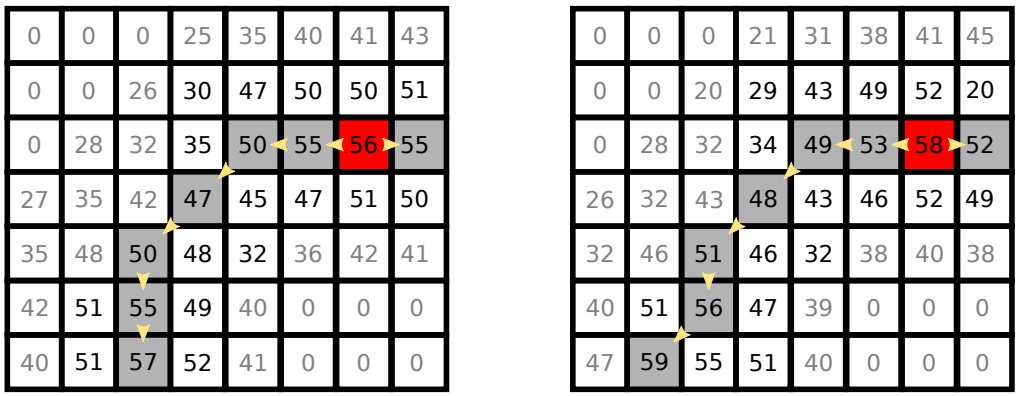
Algorithm 4.3 Procedure used to get the direction of the next pixel to be linked in SED.

Symbols used:
(x_c, y_c): Current location.
(x_l, y_l): Last pixel linked.

```
function GET_DIRECTION((x_c, y_c), (x_l, y_l), h(x_c, y_c))
    if Pixel is horizontal then
        if x_c > x_l then
            DIRECTION ← RIGHT
        else
            DIRECTION ← LEFT
        end if
    else                                              ▷ Pixel is vertical
        if y_c > y_l then
            DIRECTION ← DOWN
        else
            DIRECTION ← UP
        end if
    end if
end function
```

this thesis. For an horizontal edge, if the last displacement was to the right, then the next tested pixels are on the right, otherwise the next pixels taken into account are the ones located at the left of the current pixel. A similar approach is applied for vertical edges, if the last displacement was up, then the next tested pixels are above the current pixel, otherwise the ones below the current pixel are tested.

Figure 4-4 illustrates the linking procedure started in the left direction for an horizontal anchor. Taking the anchor point (red) as reference the pixel with the maximum gradient among the three immediate neighbours in the left direction is selected i.e. 55 among $\{50, 55, 47\}$ for the left image and 53 among $\{49, 53, 46\}$ for the right image. This process is repeated by taking as reference the new selected pixel. For the next iteration the selected pixels are 50 among $\{47, 50, 45\}$ and 49 among $\{43, 49, 43\}$ for the left and right image respectively. The next iteration selects 47 among $\{30, 35, 47\}$ and 48 among $\{29, 34, 48\}$ for the left and right image respectively. In the next iteration there is a change in orientation (from horizontal to vertical) so the neighbours to be tested are determined by Algorithm 4.3. By applying Algorithm 4.3 the next direction for each image is $DOWN$, therefore the next selected pixels are 50 among $\{50, 48, 32\}$ and 51 among $\{51, 46, 32\}$ for the left and right images respectively. This process is repeated until any of the termination conditions is reached.



(a) Left image (b) Right image

Figure 4-4: Example of the simultaneous smart routing in SED. The linking starts on the anchor points on each image (red). Then it moves to the left one pixel at the time. For a detailed analysis please refer to the text.

Similarly to ED, the linking procedure is started in the left and right directions for a horizontal anchor and in the up and down directions for a vertical anchor and repeated until any of the termination conditions is reached. Although not implemented in this thesis, if sub-pixel accuracy is required, it could be easily computed using the gradient information for fitting a curve and locating the edge at the sub-pixel level as in [150].

4.4.5 Curve Merging and Length Validation

The final stage of the disparity extractor is to merge the linked curves into long edge segments. The smart routing could produce more than one chain of pixels for the same edge. This occurs when the smart routing is stopped and the remainder of the edge is recovered by starting the linking at another anchor point on the same edge. In order to avoid these segmented edges, a simple merging strategy is applied. If two endpoints are located within a radius of size r_m and the difference in their disparities is smaller than the threshold t_m, they are assumed to be the same edge and are merged. Gaps between the merged segments could be filled by interpolation, although this is not implemented in this thesis as the gaps are small.

The proposed algorithm outputs a set of chains of connected, well-localized points in the form (x, y, d) where x and y are image coordinates on the left image and d is the disparity at pixel (x, y). These connected chains of points are able to represent the image semantics. If required, by using the camera matrices, the points could be projected into real world coordinates. This information could then be used for obstacle detection as

shown in Chapter 5.

A final optional step is to keep only 3D curves with a minimum number of pixels. This would get rid of small edges created by texture. For doing this a threshold t_l is applied which indicates the minimum length of the kept curves.

4.5 Evaluation

This section evaluates each of the parameters of the proposed algorithm. Simultaneous Edge Drawing is evaluated using the Middlebury v3 [59] and KITTI 2015 [129] datasets. The Middlebury v3 provides images with dense ground truth for indoor scenarios obtained by structured lighting techniques. The KITTI 2015 dataset provides images with semi-dense ground truth for outdoor scenarios obtained by a LIDAR system.

In order to obtain edge-based ground truth, the edges are located by Edge Drawing on the left image from the stereo pair. Then the ground truth is dilated with a structuring element of size 3×3 and only the values at the edge pixels are kept as in [28].

The section is structured as follows: first the best pixel descriptor is identified for matching the anchor points. Then the parameters of SED are evaluated to identify their impact into the accuracy of the resulting algorithm. After the best parameters are selected, SED is evaluated on the KITTI 2015 and Middlebury v3 datasets. As there are only a few curve based approaches available, SED is compared against those proposed in [54], [151].

4.5.1 Anchor matching

The anchor points are identified by the Edge Focusing algorithm across the Scale Space [148]. The correct matching of the anchor points is a crucial step in SED as it provides the disparities which are propagated along the edges. The descriptor is chosen from the ones robust to changes in illumination presented in Chapter 2.

The anchor points are matched by using a local stereo-matching approach. First the anchor point descriptors $\xi_1(a)$ and $\xi_2(a)$ are computed at a distance $d_a = m_d/2 + m_a/2 + 1$ from the anchor point using a descriptor window of size $dw_m \times dw_n$ and an aggregation window of size $aw_m \times dw_n$. Figure 4-5 shows the location of the descriptors with respect to the anchor point. Anchor descriptors are computed in the input images without noise removal.

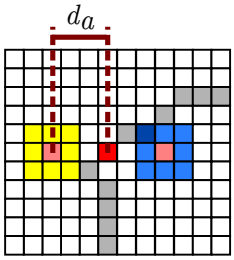


Figure 4-5: Distance d_a between the centre pixel used to compute the descriptors $\xi_1(a)$ and $\xi_2(a)$ and the anchor point a. The anchor point is shown in red, the window used for descriptor $\xi_1(a)$ and $\xi_2(a)$ are shown in yellow and blue respectively.

As stated in Section 2.2.1.1 the Census Transform and the Complete Rank transform are insensitive to changes in illumination. Therefore these descriptors are tested in this thesis. Additionally, in order to compare the performance of these descriptors with others used in the literature, an intensity strip is used as in [54] although it is not robust to changes in illumination. The Thresholded Census Transform presented in Chapter 3 showed good performance on identifying the edge pixels. Unfortunately it would produce zero strings when computed on either side of the edge in low texture regions as is the case for a location at an offset from the anchor point.

As only anchor points are matched (in contrast to all of the pixels in a dense approach) the set of candidates differs from what is presented in Chapter 3. This could change the performance of the pixel descriptors and therefore they need to be tested in this new scenario. First the effect of the transformation window and aggregation window is analysed. Then the effect of confidence metrics on the matches is tested. It is important to note that at this stage all of the anchor points are matched, in contrast as when using the simultaneous smart routing only a subset of the anchors is matched.

4.5.1.1 Window Size Analysis

The size of the transformation window and the aggregation window proved to have a high impact on the accuracy of the matches in Section 3.3.2.1 and Section 3.3.2.2. In order to test the effect of these windows on the matching of anchor points, different window sizes are tested. Figures 4-6, 4-8 and 4-10 show the effect of these window sizes. For the intensity strip descriptor no aggregation window is used as in [54]. No confidence information is used at this stage. Only the LR-confidence test is applied to identify and discard occluded anchors.

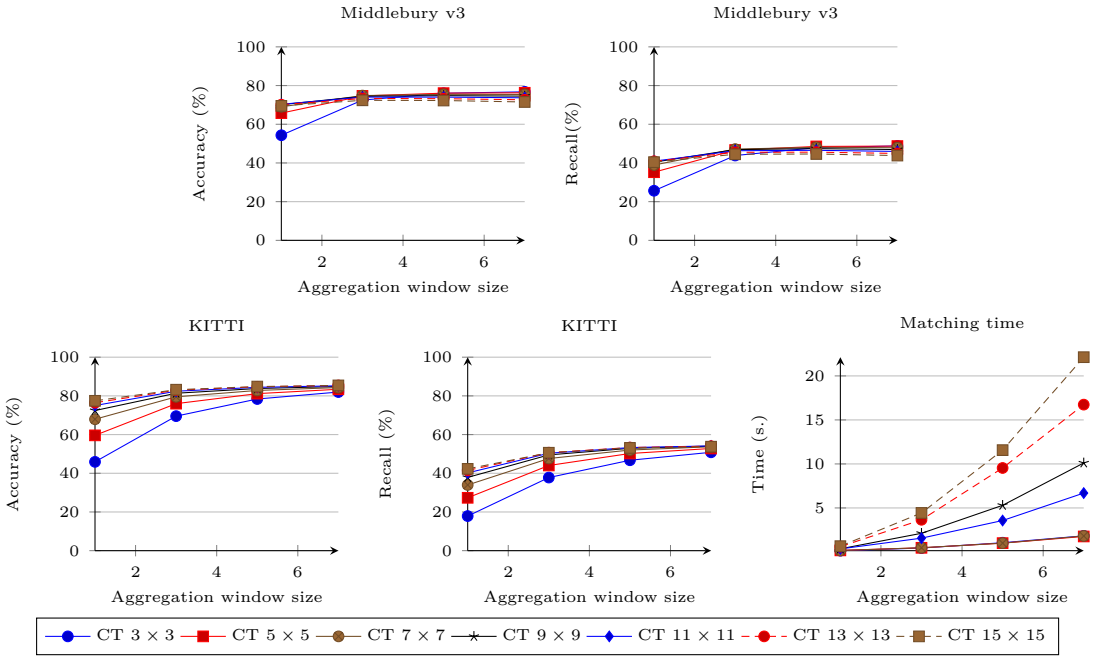


Figure 4-6: Average accuracy, recall and running time for the Census Transform at different window sizes for transformation and aggregation on the Middlebury v3 and KITTI 2015 datasets. The running time shown is for the KITTI dataset only as the image son the Middlebury v3 datasets have different sizes.

Figure 4-6 shows the effect of the transformation and aggregation window sizes for the Census Transform. It can be seen that using transformation windows larger than 9×9 does not increase the accuracy considerably on the Middlebury and KITTI datasets while the number of required computations is increased. This figure also shows that small transformation windows get more benefit from the use of aggregation windows than large transformation windows. The accuracy is peaked at 88.1% on the KITTI dataset by using a transformation window of 15×15 and aggregation window of 7×7 running on an average of 22.25 sec. Smaller windows could be used at the cost of a decrease in accuracy.

The best combinations of window size and aggregation window are summarized in Figure 4-7.

For the recall, it can be seen that the use of aggregation windows do not have a marked impact for window sizes larger than 9×9. The decrease in the accuracy for large transformation windows and aggregation windows on the Middlebury dataset is explained by the presence of low texture areas in the images. These low textures would result in similar census strings which could get matched wrongly by chance. Although these matches could be identified by using confidence metrics they are not explored until the

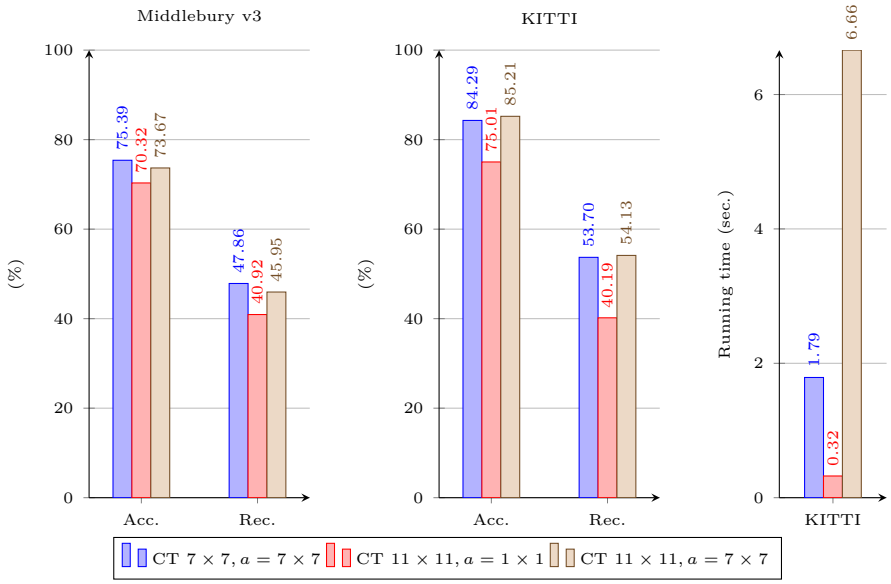


Figure 4-7: Summary of the best performing combinations of transformation and aggregation windows along with the time it takes to run the matching on a standard pc using the Census Transform as pixel descriptor. The values are an average over the dataset. Acc. and Rec. correspond to the accuracy and recall respectively. The running time shown is for the KITTI dataset only as the images in the Middlebury v3 datasets have different sizes.

following section.

By taking into account the number of required operations to perform the pixel description, it can be seen that the combinations of transform windows and aggregation windows of sizes: CT 13×13, agg 3×3; CT 9×9, agg 5×5; CT 7×7, agg 7×7 required 1512, 2000, 2352 operations respectively and produced accuracies (Middlebury, KITTI) of (75 %, 85 %); (76 %, 86 %); (77 %, 87 %) for the Middlebury and KITTI datasets respectively. The selection of the best fastest to compute combination of transformation window and aggregation window depends on the hardware used.

Figure 4-8 shows the effect of using different window sizes for transformation and aggregation on the Complete Rank Transform. Small transformation windows (3×3 and 5×5) presented an increase in accuracy and recall by using aggregation windows as more information is incorporated in the cost computation. Medium window sizes (7×7, 9×9) do not get a marked increase in accuracy or recall by the use of aggregation windows. Therefore for these window sizes it is better not to use aggregation windows. Large transformation windows (11×11 and larger) obtained an increased accuracy by using aggregation windows at the cost of decreasing the recall. This is explained as large

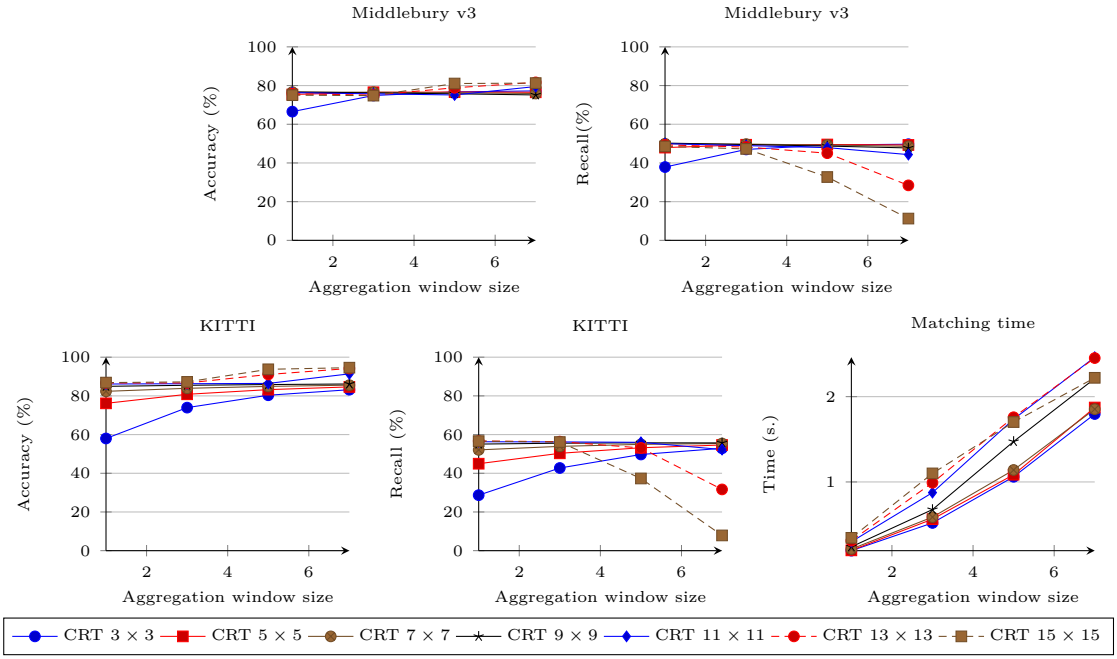


Figure 4-8: Average accuracy, recall and running time for the Complete Rank Transform at different window sizes for transformation and aggregation on the Middlebury v3 and KITTI 2015 datasets. The running time shown is for the KITTI dataset only as the images in the Middlebury v3 datasets have different sizes.

transformation windows may incorporate information which belong to different surfaces when small objects are present in the scene. These wrong matches are rejected by the LR-consistency check resulting in a reduced recall.

Regarding the number of operations and timing, the use of aggregation largely increases the number of required computations. Therefore it is preferable not to use aggregation windows. The best combinations of window size and aggregation window are summarized in Figure 4-9. For the remainder of this chapter, a transformation window of 9×9 with no aggregation window is selected as it results in a low number of operations while sacrificing 2% on accuracy on respect to a window size of 11×11 which has a larger running time.

Figure 4-10 shows the accuracy and recall of an intensity strip. Although this approach was able to obtain accuracies similar to the CRT and CT, it is expected that suffers from sensitivity to changes in illumination as it is based on intensities and it is presented only as reference. As the focus of this thesis is on depth extraction for obstacle detection, the robustness to changes in illumination of the descriptors is not addressed. The reader is referred to [25], [34] for extensive analysis on the effect of changes in illumination for different pixel descriptors.

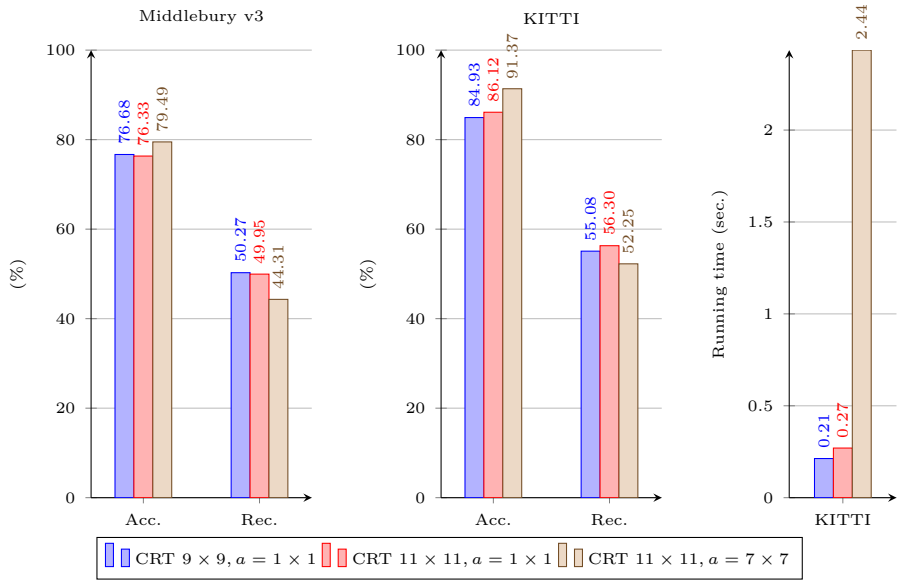


Figure 4-9: Summary of the best performing combinations of transformation and aggregation windows along with the time it takes to run the matching on a standard pc using the Complete Rank Transform as pixel descriptor. The values are an average over the dataset. Acc. and Rec. correspond to the accuracy and recall respectively. The running time shown is only for the KITTI dataset as the images in the Middlebury v3 datasets have different sizes.

Figure 4-11 shows the accuracy, recall and running time for the CT, CRT and intensity strip using the best performing combination of windows for transformation and aggregation. This figure shows that the CRT outperforms the CT and the intensity strip on recall meanwhile the running time is lower. For the remainder of this chapter the CRT with a transformation window of 9×9 is used as a pixel descriptor. This selection is based on a trade-off between accuracy and required computations.

4.5.1.2 Confidence Analysis

As the accuracy of the obtained disparity maps is not 100%, confidence measures have to be applied in order to decrease the number of wrong matches at the cost of decreasing the recall. This decrease in the recall might be a problem for dense approaches. But for SED, these matches are seeds for starting the pixel linking. The effect of the confidence metrics on the matching of the anchors is analysed in the remainder of this section.

The β_{PKR} confidence metric (see Equation (3.7)) has been shown to be able to get rid of wrong matches caused by repetitive structures [65]. This confidence metric is used in this thesis as these kinds of structures may be found on trees and footpaths. Figure 4-12 shows the effect on accuracy for different thresholds on this confidence metric. A threshold

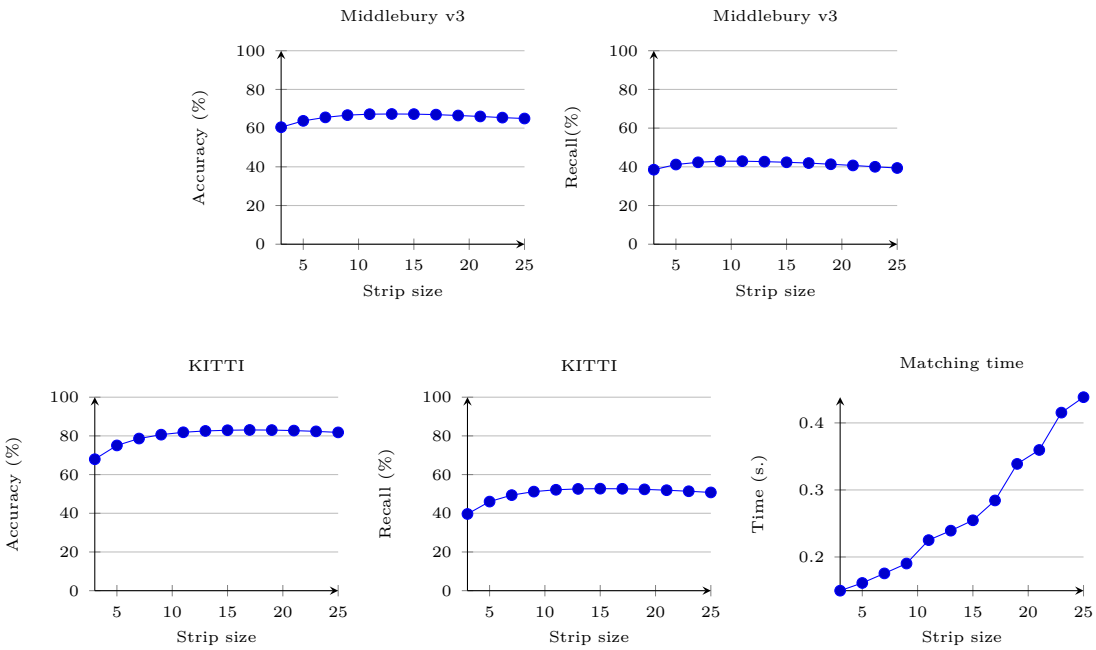


Figure 4-10: Average accuracy, recall and running time for an intensity strip of different lengths on the Middlebury v3 and KITTI 2015 datasets. The running time shown is for the KITTI dataset only as the images in the Middlebury v3 datasets have different sizes.

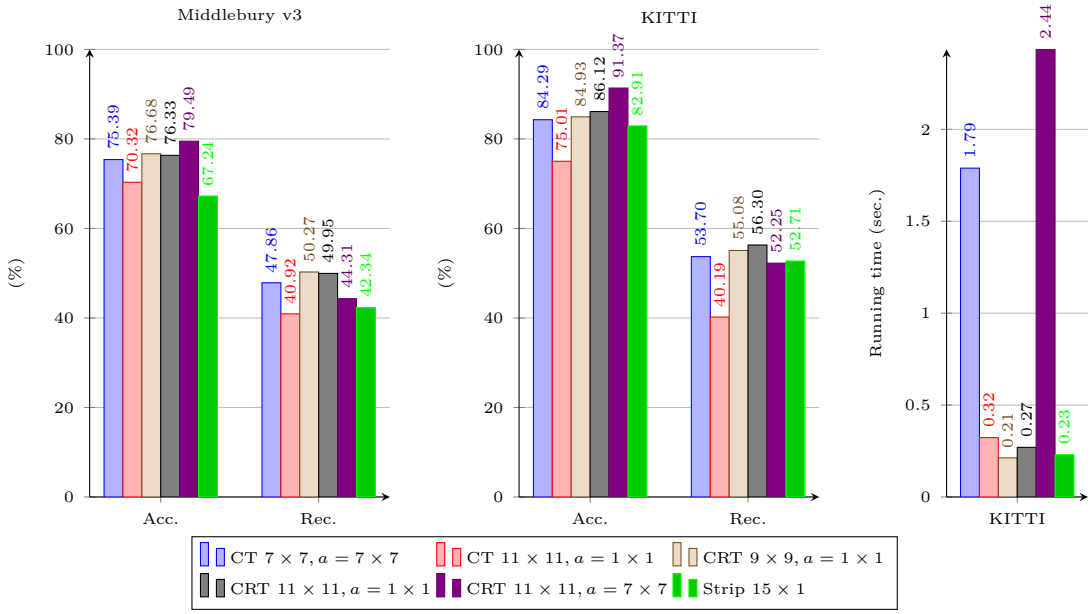


Figure 4-11: Comparison of the average accuracy, recall and running times for matching the anchor points using the CT, CRT and intensity strip. Acc. and Rec. correspond to the accuracy and recall respectively. The time is computed only for the KITTI dataset as the images in the Middlebury dataset have different sizes. a represents the size of the window used for aggregation.

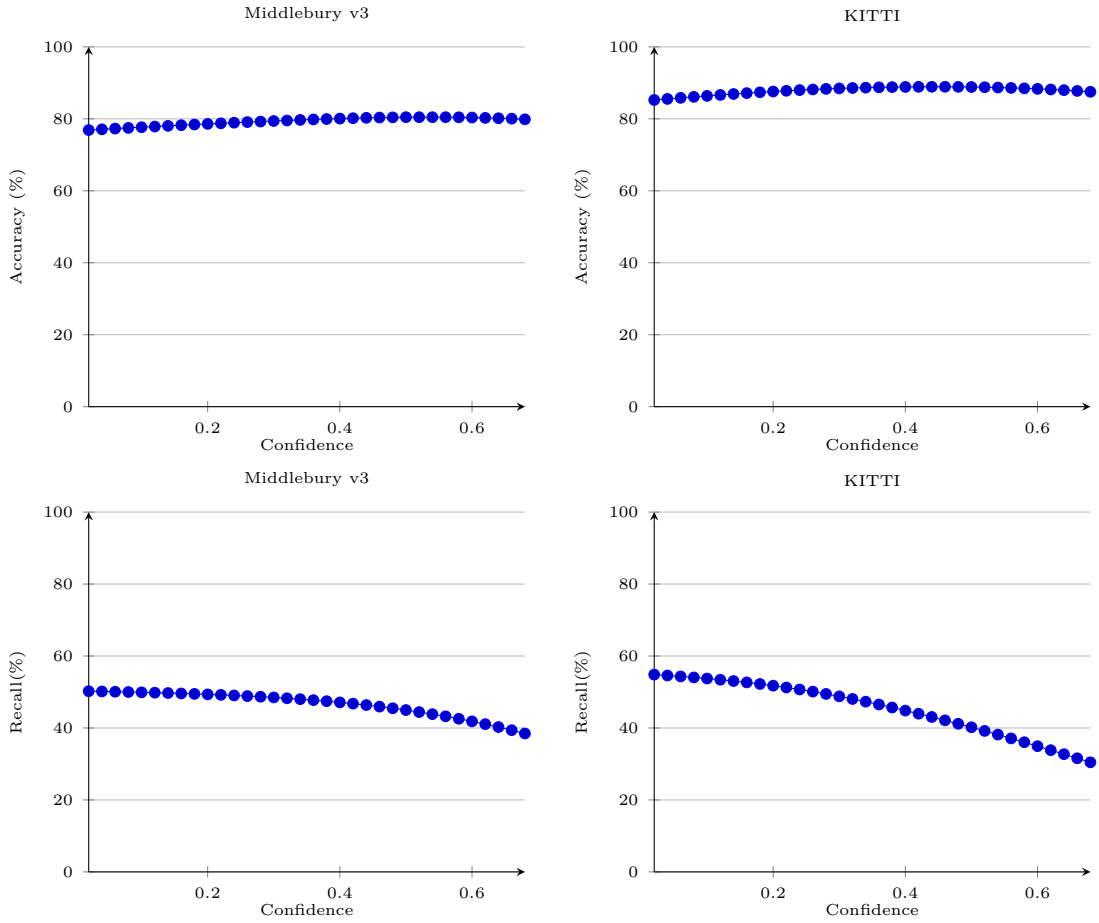


Figure 4-12: Average accuracy and recall for the β_{PKR} confidence metric using a Complete Rank Transform of size 9×9 without aggregation.

of 0.44 produced a maximum on the accuracy for the KITTI dataset at 88% accuracy and 43% recall. Although this corresponds to a 12% decrease in the recall, a human viewer is still able to recognize the contents of the scene. For the Middlebury dataset the threshold of 0.44 corresponds to 80% and 46% for accuracy and recall respectively which differ from the maximum in less than 1%. The PKR confidence metric is used with a threshold of 0.44 for the remainder of this chapter.

Figure 4-13 shows the effect on the accuracy and recall of β_{MAX} as confidence metric. This metric proved successful in removing most of the wrong disparities appearing as random points on the disparity images. The β_{MAX} confidence metric presents a marked decrease in recall as it gets close to 1. This occurs as only a few matches have a dissimilarity of 0 due to differences in the cameras and noise. The confidence metric is able to increase the accuracy up to 92% at the cost of decreasing the recall to 28% on the KITTI dataset for

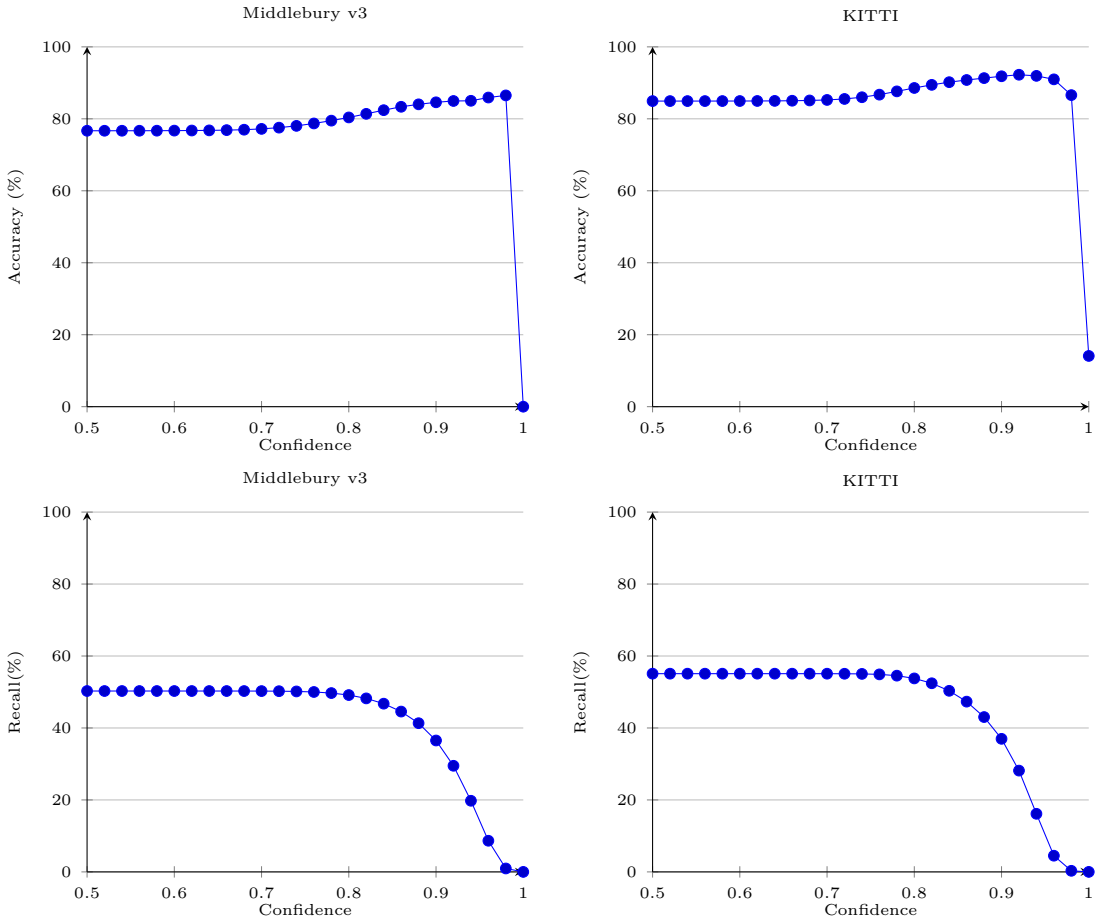


Figure 4-13: Average accuracy and recall for the β_{MAX} confidence metric using a Complete Rank Transform of size 9×9 without aggregation.

a threshold of $t_{MAX} = 0.92$. Disparity maps obtained by using this threshold $t_{MAX} = 0.92$ are not recognizable by a human viewer. Alternately a threshold $t_{MAX} = 0.88$ produced an accuracy of 91% with a recall of 43% on the KITTI dataset resulting in disparity maps recognizable by a human viewer. For the Middlebury dataset the accuracy and recall are 84% and 41% respectively using the threshold $t_{MAX} = 0.88$. The β_{MAX} confidence metric is used with a threshold of 0.88 on the remainder of this chapter.

Figure 4-14 shows an example of the obtained disparity images with and without applying the confidence metrics. It can be seen that the number of erroneous pixels decreased considerably without affecting the ability for a human viewer to identify the contents of the image. Table 4.1 shows the selected thresholds to apply for each confidence metric. By combining both confidence thresholds the average accuracy is increased to 85% and 93% on the Middlebury and KITTI datasets respectively. Meanwhile the resulting

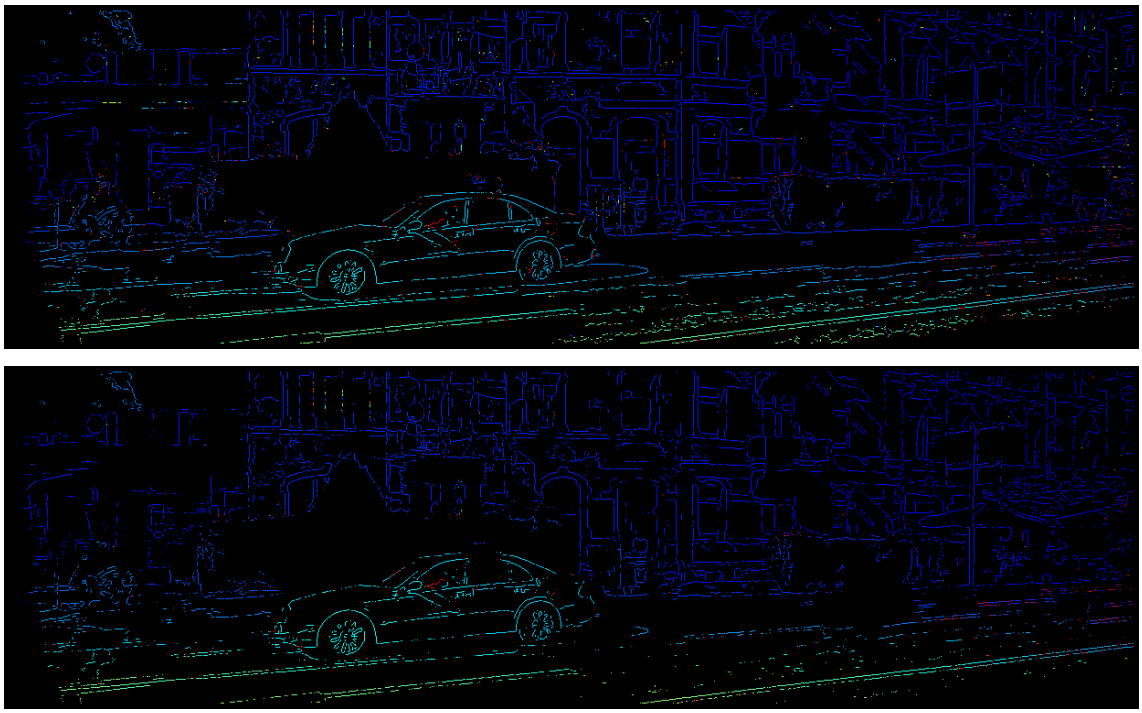


Figure 4-14: Sample disparity images with and without using confidence measures. The top image is the obtained disparities without applying the confidence thresholds. The bottom image is after applying β_{PKR} and β_{MAX} confidence measures. Red pixels correspond to erroneous matches. The disparity is color-coded from low values (light-green) to high values (blue). Black pixels correspond to non-matched or invalid matched pixels.

Confidence metric	Threshold
β_{PKR}	0.44
β_{MAX}	0.88

Table 4.1: Selected thresholds for the accuracy metrics used in this thesis.

recall is 39% and 35% on the Middlebury and KITTI datasets respectively.

4.5.1.3 Discussion

The Complete Rank Transform of size 9×9 without aggregation resulted the in best pixel descriptor for matching the anchor points of SED. The confidence metrics β_{PKR} and β_{MAX} were shown to increase the accuracy of the matches and the resulting images were still recognizable by a human viewer. Although other combinations of descriptor sizes could be used for matching the anchors they have a marked impact on the computing time. The hardware utilized for the implementation would largely influence the selection of transformation window and aggregation sizes. As an embedded environment is expected for the algorithms used in this thesis, the resources are assumed to be minimal.

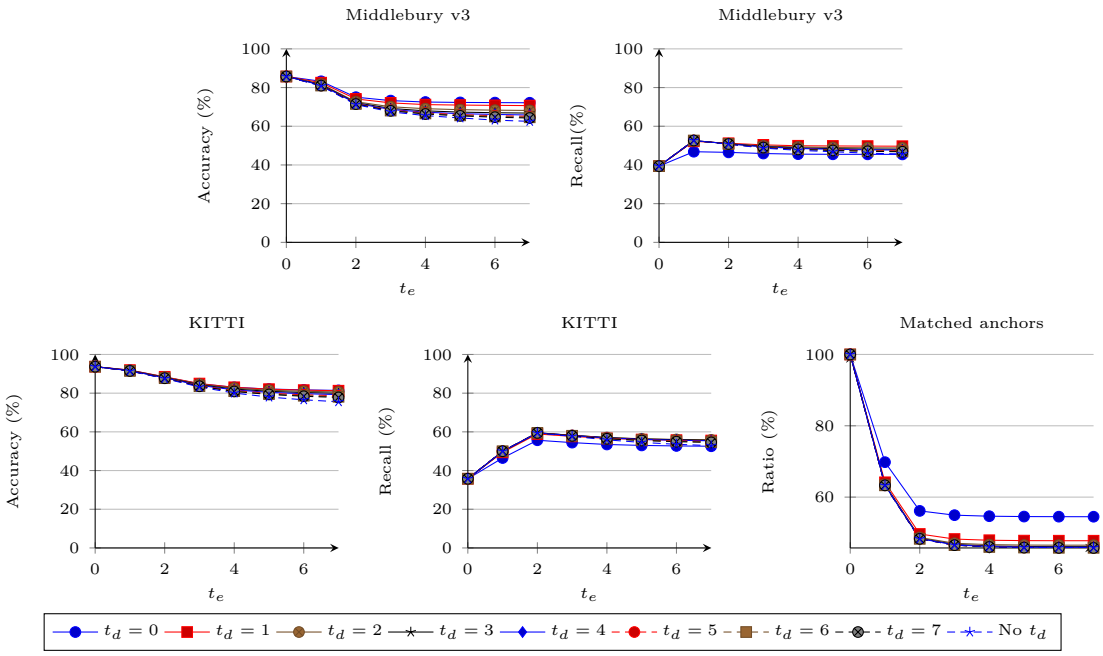


Figure 4-15: Average accuracy, recall and ratio of matched anchors for different values of the disparity and epipolar thresholds t_d and t_e.

4.5.2 Simultaneous Smart Routing

After the anchor pixels have been matched, the simultaneous smart routing procedure is evaluated on the Middlebury v3 and KITTI datasets. The simultaneous smart routing procedure propagates the disparities from the matched anchor points along the image edges producing chains of points in the form (x, y, d) where x and y are coordinates in the left image of the stereo-images and d is the disparity corresponding to the voxel of the 3D edge.

The parameters of the simultaneous smart routing procedure are: disparity threshold t_d, this indicates the amount of change in disparity allowed with respect to the matched anchor used as starting point of the linking. Epipolar threshold t_e, this creates a tolerance against errors in the vertical location of the edge which could be caused by non-perfect image rectification.

Figure 4-15 shows the effect of thresholds t_d and t_e on the accuracy, recall and number of matched anchors on the obtained disparity maps. This figure shows that there is no significant advantage in the use of the disparity threshold t_d on the accuracy and recall on small values of t_e, whereas large values of t_e obtain only a small improvement in accuracy when using small values of t_d. For the recall and number of matched anchors any value

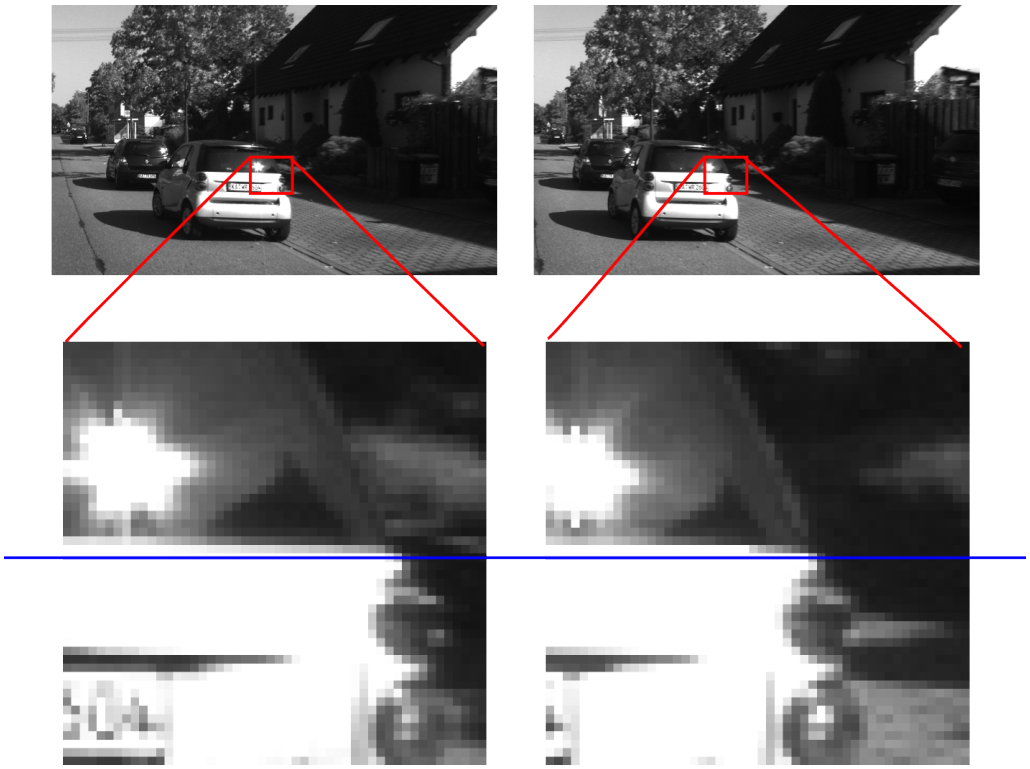


Figure 4-16: Example of vertical misalignment on rectified images. The rear window boundary is one pixel higher on the left image than in the right. The epipolar threshold t_e adds tolerance to these kind of misalignments. Images at the top correspond to the left and right views from a stereo-camera in the KITTI dataset. The bottom images show a zoomed in region where due to aliasing the edges are located at different vertical coordinates. These misalignments could also be produced by errors on the calibration, rectification and camera drift.

$t_d > 0$ produce similar results.

The analysis of the epipolar threshold t_e shows that there is a gain in recall by allowing the linked pixels to have a vertical offset in their location at the cost of a small decrease on accuracy. When the epipolar misalignment is allowed ($t_e > 0$), curves which do not lay on the same y coordinate (see Figure 4-16) may be linked.

By allowing these non-aligned curves to be linked, longer curves are produced resulting in an increase on the recall and a reduction on the number of anchors required to be matched. This decrease in the number of matched anchors translates into fewer required computations. The downside of allowing these non-aligned pixels is that the pixels could indeed be a wrong match resulting in a decrease in accuracy.

A reasonable trade-off between decrease in accuracy and number of matched anchors could be found when $t_e = 1$ which results in an accuracy of 80% and 91% for the Middlebury and KITTI datasets respectively with a recall of 52% and 50%. This compares

to the accuracy of 85% and 93% and recall of 39% and 35% obtained on the Middlebury and KITTI datasets respectively obtained when no epipolar miss-alignment is allowed. The use of $t_e = 1$ requires the matching of only 63% of the found anchors in the image meanwhile, if the miss-alignment is not allowed 99% of the anchors needs to be matched.

As can be seen in Figure 4-15 t_d has no significant influence on the accuracy, recall and number of matched anchors when $t_e = 1$. Therefore in the remainder of this thesis the disparity threshold t_d is not used and the epipolar threshold is set to $t_e = 1$.

4.5.3 Curve Merging and Length Validation

The stopping criteria of the smart routing could result in multiple short edge segments instead of a long curve for a given edge. In order to obtain longer curves a simple curve merging strategy is applied: if two end-points are located within a radius r_m and their disparity difference is within a threshold t_m they are assumed to be the same edge and are merged. If more than two end-points are located in the same radius, the first found pair of endpoints with a disparity difference within threshold t_m are merged.

In order to discard edges which might be created by texture elements, small edges are discarded as in [152]. The minimum length of an edge is set to 1% of the length of image diagonal in order keep it related to the image resolution.

The effect of the merging radius and length validation is shown in Figure 4-17. This figure shows that by deleting small curves the recall is reduced. This reduction in the recall is explained by small edges produced by texture elements which remained after applying the Gaussian Scale Space or when the stopping criteria of the simultaneous smart routing break the linking. The curve merging does not produce a change in accuracy but the recall could be increased as small edges are joined into long segments which pass the final length validation.

The merging threshold t_m does not produce any change in the accuracy or recall of the merged curves but it might allow the merging of non-related edges. In order to avoid this the merging threshold is set to $t_m = 1$ forcing the algorithm to take into account only disparities which are in the same disparity region. The merging radius r_m is set to $t_m = 5$ as no marked increase in the recall is obtained with larger values.

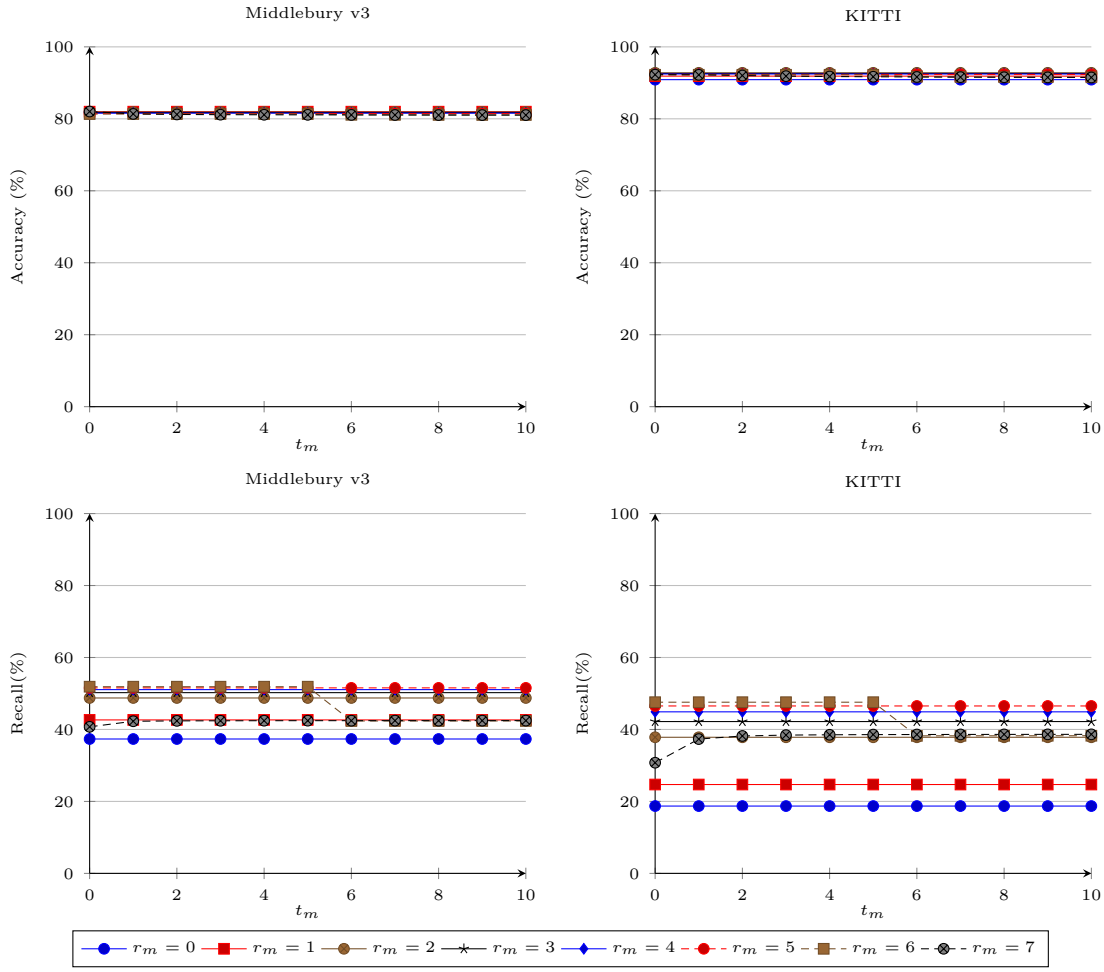


Figure 4-17: Average accuracy and recall for different values of r_m and t_m when only edges longer than the 1% of the image diagonal are kept as in [152].

4.5.4 Scan Interval Speed Up

The scan interval s is used in ED to reduce the number of obtained anchor points. The anchor point matching is one of the most computationally intensive tasks from SED. As real-time performance is required a reduction in the number of anchor points would lead to a reduction in the overall computing time.

Figure 4-18 shows the effect of different values for the scan interval s. This figure shows that by increasing the size of the scan interval the matching time is decreased. This occurs as the number of available anchor points is decreased. A value of $s = 2$ resulted in a decrease in matching time of around 50% without affecting the accuracy. The reduced recall is caused by the reduction small edges recovered. Values of $s > 3$, although faster to compute, produced disparity images in which a human viewer was unable to identify the image contents. This is explained as horizontal edges not connected to verticals are

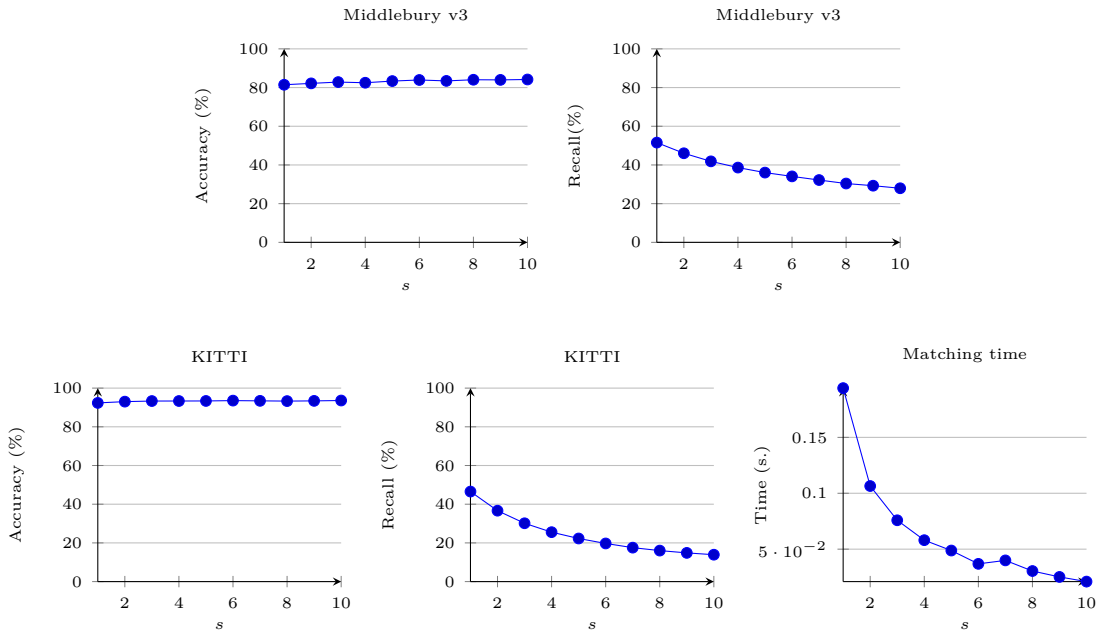


Figure 4-18: Average accuracy and recall for different values of the scan interval s inherited from ED.

not recovered by large scan intervals. A value of $s = 2$ produced a significant decrease in the number of anchors without sacrificing the visual quality of the obtained images.

4.5.5 SED Stage Timing

This section presents the obtained timings for each of the stages of Simultaneous Edge Drawing for the KITTI datasets. The Middlebury v3 dataset is not used in this timing as the images have different sizes, and an analysis per image would be required. It is important to note that no parallel processing, SIMD instructions or GPUs are used at this stage. Therefore the running times could be improved even further by using these tools when available

Table 4.2 shows that most of the time is spent on the computation of the gradient, orientation, edge candidates and anchor points. Particularly, the edge focusing algorithm [148] is the bottleneck as the scale space sampling and gradient computations require image convolutions at each iteration. After this stage, the anchor matching is the second most computationally expensive as the descriptors and dissimilarity measures must be computed for the required anchors. The simultaneous smart routing which propagates the disparity values is extremely fast as it is with ED. The total time includes all the required

SED stages	Time (ms)	
	laptop	Raspberry Pi 2
Gradient, orientation, edge candidates and anchor points	128	3550
Anchor sorting	1	23
Anchor matching	61	1182
Simultaneous smart routing	4	53
Curve merging and length validation	5	78
Total (including memory allocations)	208	5011

Table 4.2: Time required for each of the stages of SED. Time is the average for the KITTI dataset which contains images of around 1MPx resolution. The laptop implementation is using only one core @ 2.1 GHz. The Raspberry Pi 2 implementation runs on only one core @ 900 MHz.

memory allocations, initialization and copy vectors and matrices. Therefore it has some overload in respect to each of the stages.

As the target application of the proposed algorithms are embedded systems, the timing of SED is measured on the Raspberry Pi 2 Model B, resulting in a 10X reduction in the computing speed. Although the obtained times are not real-time, 1 MPx were used for the metrics. An embedded application typically uses lower resolution images, leading to a reduction in running times.

4.5.6 State-of-the-art Comparison

Only a few available edge based disparity extractors are available in the literature, namely, EBDP [28], EMCBR [54] and DP [82]. Table 4.3 compares the results obtained by SED on the same image subset from the Middlebury dataset where EBDP and EMCBR are tested. For DP [82] no information on the dataset is provided, therefore it is not compared against SED.

In order to provide sub-pixel accurate disparities as in EBDP and EMCBR parabolic fitting was applied to the edge location in order to locate the edge at sub-pixel level as in [150]. Then the disparity is taken as the difference in the sub-pixel location of the edges as in [28] and [54]. The parameters from SED are set as specified on the previous sections. These values are as follows: $\sigma_{MAX} = 4$ for the edge focusing algorithm; scan interval $s = 2$; the pixel descriptor is a 9×9 CRT with no aggregation; confidence thresholds $\beta_{PKR} = 0.44$ and $\beta_{MAX} = 0.88$; epipolar threshold $t_e = 1$, merging radius and threshold are set to $r_m = 5$ and $t_m = 1$ respectively.

Table 4.3 shows the comparison of the obtained results on the same images as provided

	Tsukuba		Teddy		Cones		Venus		Sawtooth		Running
	Matches	Errors	Matches	Errors	Matches	Errors	Matches	Errors	Matches	Errors	time (ms.)
SED (Thesis)	6178	10.9 %	9396	8.7 %	14188	7.7 %	8527	3.4 %	11694	7.2%	45 - 83, 1 core
EBDP	9920	7.6 %	11755	11.9 %	15155	5.4 %	11610	1.8 %	14614	2.7 %	60 - 85, 2 core
EMCBR	8550	8.8 %	10514	5.3 %	16147	5.3 %	11816	2.0 %	14202	2.4 %	20 - 45, 2 core

Table 4.3: Comparison between SED, EBDP and EMCBR.

in [28], [54]. This table shows that although the algorithm proposed in this thesis has a slightly higher error for some images the running time is close to EBDP using only one core. Figure 4-19 shows the obtained images by SED and EBDP. The images for EBDP were taken from [28]. It can be seen that the increase in the error rate for SED is due to the smaller number of detected edges as texture edges are ignored, whereas EBDP computes disparity for texture edges.

4.5.7 Benchmarking

In order to benchmark the proposed Simultaneous Edge Drawing algorithm it was evaluated on the Middlebury v3 and KITTI 2015 datasets. Table 4.5 and Table 4.4 show the obtained results on the Middlebury v3 and KITTI dataset 2015 respectively.

Table 4.4 summarises the top 20 of the results on the KITTI dataset using only the estimated disparities for evaluation. The table shows that SED is among the top-15 in the ranking being the only one with a running time smaller than one second. without using multiple cores or GPUs.

Table 4.5 summarises the top 7 of the entries on the Middlebury v3 dataset using only the estimated disparities for evaluation. The input images were taken as full resolution. This table shows that SED is the top performing method at the time of writing. Additionally SED is the fastest approach not using GPUs or parallel processing.

4.5.8 Discussion

This section evaluated Simultaneous Edge Drawing for estimating the disparity maps from stereo-images. The best combination of parameters was found experimentally using the Middlebury v3 and KITTI 2015 datasets while performing a trade-off between accuracy and running time.

The evaluation on the Middlebury v3 and KITTI 2015 datasets show that SED has an accuracy similar to other state-of-the-art methods meanwhile the running time is significantly lower while using only one thread for processing.

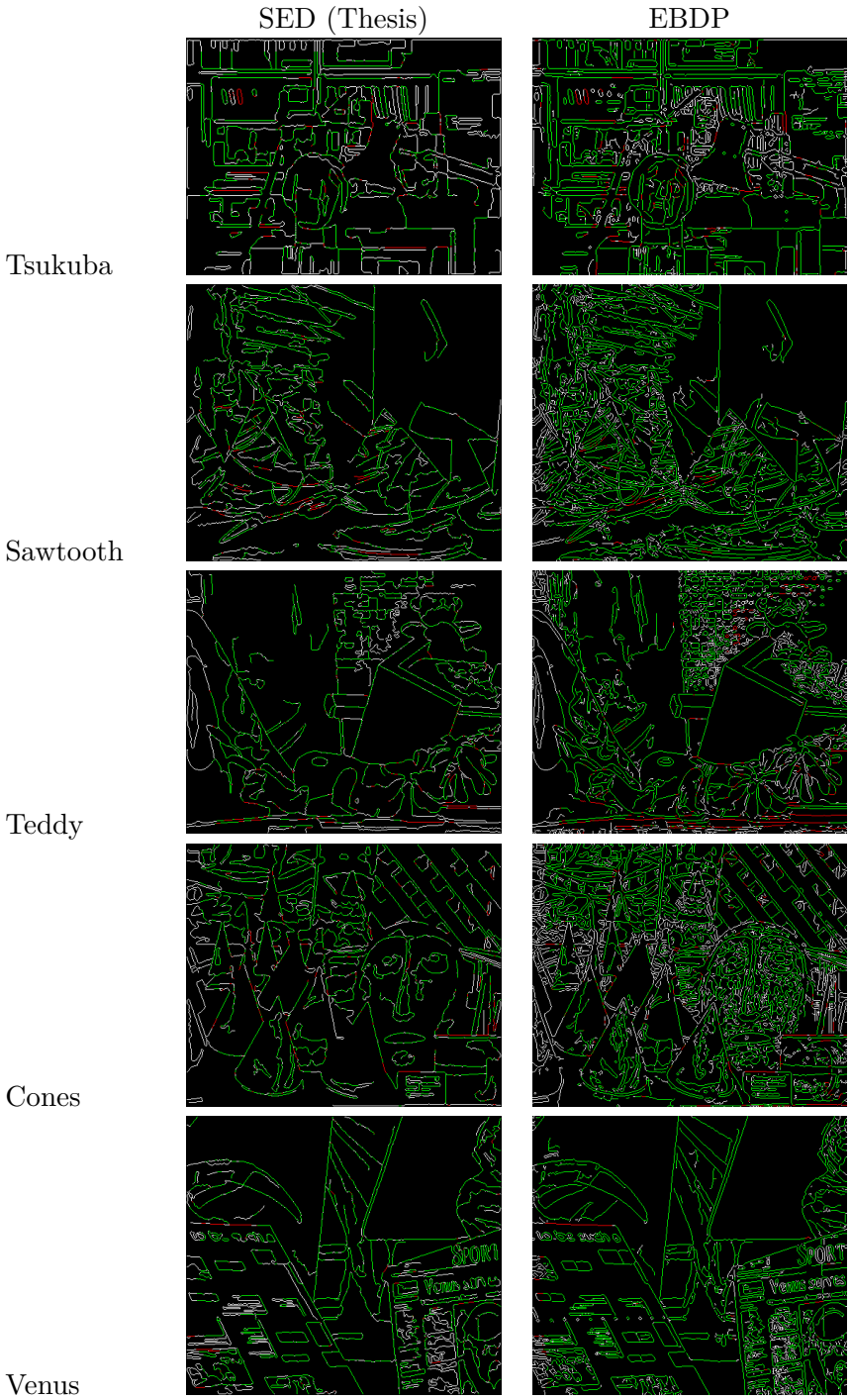


Figure 4-19: Disparity results for SED (left) and EBDP (right). Green pixels have an error threshold $\epsilon \leq 1$. Red pixels have an error threshold $\epsilon > 1$. White pixels are unmatched pixels. The images for EBDP were taken from [28]. The edges for SED are detected by ED.

Method	D1-bg	D1-fg	D1-all	Density	Time	Environment
Displets v2	3.00 %	5.56 %	3.43 %	100.00 %	265 s	>8 cores @ 3.0 Ghz (Matlab + C/C++)
PBCP	2.58 %	8.74 %	3.60 %	100.00 %	68 s	Nvidia GTX Titan X
SDM	2.56 %	10.25 %	3.65 %	62.56 %	1 min	1 core @ 2.5 Ghz (C/C++)
MC-CNN-acrt	2.89 %	8.88 %	3.88 %	100.00 %	67 s	Nvidia GTX Titan X (CUDA, Lua/Torch7)
CNN-SPS	3.30 %	7.92 %	4.07 %	100.00 %	80 s	GPU @ 2.5 Ghz (C/C++)
PRSM	3.01 %	10.52 %	4.26 %	99.99 %	300 s	1 core @ 2.5 Ghz (C/C++)
DispNetC	4.32 %	4.41 %	4.34 %	100.00 %	0.06 s	Nvidia GTX Titan X (Caffe)
Content-CNN	3.73 %	8.58 %	4.54 %	100.00 %	1 s	Nvidia GTX Titan X (Torch)
LPU	3.55 %	12.30 %	5.01 %	100.00 %	1650 s	1 core @ 2.5 Ghz (Matlab + C/C++)
SGM+CNN	3.93 %	10.56 %	5.04 %	100.00 %	2 s	Nvidia GTX 970
EEL	3.85 %	11.16 %	5.07 %	99.99 %	5 s	1 core @ 2.5 Ghz (C/C++)
SPS-St	3.84 %	12.67 %	5.31 %	100.00 %	2 s	1 core @ 3.5 Ghz (C/C++)
MDP	4.19 %	11.25 %	5.36 %	100.00 %	11.4 s	4 cores @ 3.5 Ghz (Matlab + C/C++)
CNN-MS	3.89 %	13.28 %	5.45 %	100.00 %	3 min	GPU @ TITAN X (Lua/Torch)
SED (Thesis)	4.90 %	8.29 %	5.74 %	4.02 %	0.68 s	1 core @ 2.0 Ghz (C/C++)
OSF	4.54 %	12.03 %	5.79 %	100.00 %	50 min	1 core @ 2.5 Ghz (C/C++)
OCV-SGBM	4.45 %	13.24 %	5.86 %	90.41 %	1.1 s	1 core @ 2.5 Ghz (C/C++)
CSF	4.57 %	13.03 %	5.98 %	99.99 %	80 s	1 core @ 2.5 Ghz (C/C++)
MBM	4.69 %	13.05 %	6.08 %	100.00 %	0.13 s	1 core @ 3.0 Ghz (C/C++)
PR-Sceneflow	4.74 %	13.74 %	6.24 %	100.00 %	150 s	4 core @ 3.0 Ghz (Matlab + C/C++)

Table 4.4: Results from SED on the KITTI 2015 dataset. The evaluation is performed taking into account only the estimated pixels. D1-bg = outliers on background regions, D1-fg = outliers on foreground regions, D1-all = outliers over all ground truth pixels. This would place the proposed approach in the top 15 at the time of writing, being the only one with a running time smaller than 1s. without the use of multiple cores or GPUs.

Name	Avg	Austr	AustrP	Bicyc2	Class	ClassE	Compu	Crusa	CrusaP	Djemb	DjembL	Hoops	Livgrm	Nkuba	Plants	Stairs
SED (Thesis)	0.26	0.38	0.31	0.28	0.22	0.2	0.36	0.13	0.15	0.28	0.72	0.22	0.21	0.19	0.2	0.2
R-NCC	0.39	0.24	0.64	0.34	0.05	0.09	0.66	0.24	0.13	0.42	0.33	0.19	0.76	0.53	0.2	0.88
ICSG	2.31	6.08	1.76	1.37	0.08	4.37	1.98	3.75	2.26	1.15	3	2.78	2.09	2.19	3.05	2.22
SGM	3.33	11.7	1.64	2.04	2.01	3.04	3.56	6.15	3.41	2.45	2.19	4.1	3.39	2.35	3.61	1.01
IDR	4.54	11	0.95	1.49	1.76	1.73	7.34	10.3	6.11	2.47	2.82	6.79	4.81	3.69	4.46	4.36
TMAP	5.96	8.83	1.78	2.77	3.63	8.49	6.21	9.46	6.79	2.23	7.88	8.72	8.32	6.93	6.18	6.36
SNCC	6.24	17.3	3.32	3.61	4.45	5.48	7.39	13.3	9.4	3.49	3.4	6.46	4.1	3.99	7.07	3.32
PMSC	6.87	3.46	2.68	6.19	2.54	6.92	6.54	3.96	4.04	2.37	13.1	12.3	12.2	16.2	5.88	10.8
INTS	7.29	12.9	2.09	3.81	4.28	10.6	6.78	12	7	3.04	6.69	12.4	9.4	8.37	6.42	13.2
SGM	7.29	20.7	2.62	3.58	4.11	7.62	8.79	16.3	6.9	3.38	4.78	10.8	7.62	5.76	7.06	6.03

Table 4.5: Results from SED on the Middlebury v3 stereo evaluation using only the estimated pixels. The shown results correspond to the test set. The shown results correspond to the percentage of pixels which an error larger than 2 disparities on non-occluded pixels. These values are the default for the ranking.

4.6 Discussion

This chapter introduced SED, a new, fast and accurate approach for low-level stereo-matching based on the principle of using the image edges to propagate the disparity from a few matched points. The proposed approach was evaluated on the Middlebury v3 and KITTI 2015 datasets in order to compare it to other state-of-the-art approaches. Although other approaches exist for stereo-matching based on edge information, the proposed approach is able to provide edge-disparities with connectivity information and to be as fast as approaches relying on the use of parallel processing.

The speed of SED relies on the reduction of the number of pixels matched by a search, as only a fraction of the candidates anchor-points require to be matched. In addition to the reduction in matches, SED also reduces the amount of required storage as it uses a point-chain representation of the edges, in contrast to other approaches where the whole input

images are stored. By using image-edges and an illumination-robust pixel descriptor SED incorporates robustness to lighting conditions without any extra computation. Another advantage of SED over other edge based approaches is the reduction of the number of texture edges by using of the Scale Space to keep only the most important edges.

Further research on SED would improve the performance of the anchor detection and the anchor matching as those were found to take 60% and 29% of the running time respectively. An improvement in these stages would result in a considerable increase in performance as the disparity propagation stage typically takes only 5ms on a 1 MPx image. Additionally, the accuracy of SED could be increased by the early identification of inaccurate matched anchors.

In summary, the high speed, state-of-the-art accuracy, compact representation and robustness to changes in illumination of SED open a new door to stereo-matching approaches in constrained environments. Therefore the point-chains produced are used in Chapter 5 for fast obstacle detection.

5

Edge-Based Stixels

5.1 Introduction

This chapter introduces a new approach for estimating stixels using as input edge-based disparity maps. An image usually contains only a few pixels belonging to image edges [55]. This sparsity translates into a reduction of the required storage and a speed-up as only edge-pixels are processed while incorporating more semantic and geometric meaning than feature points [16]. Stixels allow an efficient 2D representation of the occupied and free space around a mobile robot [153]. The possibility of extracting stixels from edge-based disparity images is analysed in this thesis as it would allow a reduction in the required storage and a speed up in the processing as only few pixels are processed in comparison to dense approaches where all of the image pixels are used.

A stixel is a vertical rectangle in image coordinates with an associated disparity for one of the 2D views of the stereo-camera. This middle-level representation allows the identification of free and occupied areas in the 2D views from a stereo-pair. Stixels may be represented by 5-tuples $s = (x, y, h, w, d)$. They represent the 2D projection of an obstacle in a view from the stereo pair. This compact representation allows a reduction in the required memory footprint compared to using a dense disparity map [153] for representing the obstacles.

As presented in Chapter 2 stixels have been successfully used for identifying pedestrians [154] and identifying traffic [153] at high frame-rates. Although effective, up to now they have been obtained only from dense disparity images [121], [112] or by using a dense cost-volume [115].

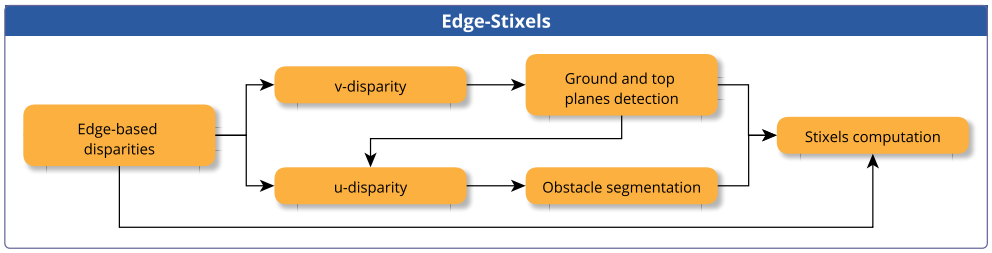


Figure 5-1: Pipeline for the computation of stixels from edge-based disparity maps

The approach in the literature closest to the approach in this thesis is that proposed by Benenson, Timofte, and Van Gool [115] where they reduce the computation time by only matching pixels located around the high gradient areas to compute the disparity and then use the gradients to locate the top and bottom boundaries of the stixel [115]. This approach, although effective and fast, has the limitation that no partial occlusion is allowed as only one stixel may exist per image column.

In order to take advantage of the compact representation of edge-based disparity maps, this chapter proposes for the first time an approach for estimating stixels using edge-based disparity maps as input. This chapter is organised as follows: first, the proposed approach for estimating stixels from edge-based disparity maps is presented. Second, the proposed approach is evaluated on different datasets by using edge-based disparity maps obtained from dense approaches and by using the output from the SED algorithm presented in Chapter 4 and by taking the edge disparities from a dense disparity map obtained by a Semi-Global Block Matching approach (see Section 2.2.3.3).

5.2 Stixels from Edge-based Disparities

Current approaches for the computation of image stixels use all of the image pixels to estimate their location. This step is computationally expensive as it requires to compute a dense disparity map or a dense cost volume. In order to cope with this limitation, a new approach based on sparse, edge-based disparity maps is proposed in this thesis.

The proposed approach for estimating stixels from sparse edge-based disparities is shown in Figure 5-1 and is as follows: First the ground plane is identified. Second, the edge-disparities are divided into top/ground-planes. Then the edge-disparities not labelled as top/ground-planes are segmented into obstacles. Finally the stixels are estimated by

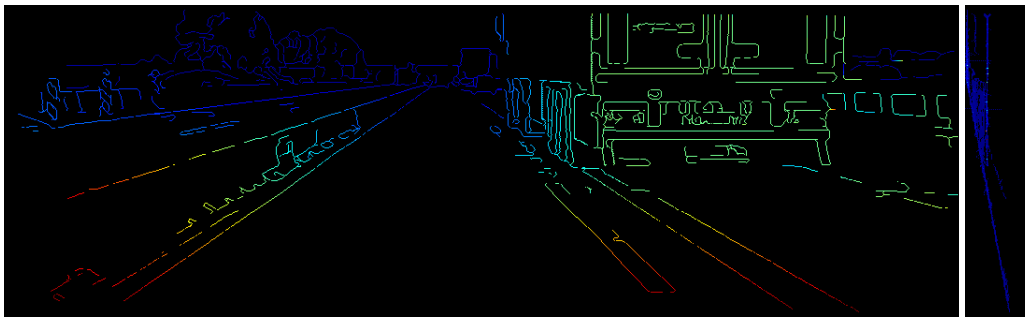


Figure 5-2: Sample v-disparity image (right) from an edge-based disparity map (left).

using the edge-disparities belonging to the obstacles. This approach does not need the computation of an occupancy grid as in [112] or assume there is only one obstacle per image column as [115].

5.2.1 Ground Plane Identification

The ground plane is estimated by using the v-disparity image (see Section 2.3.1). Labayrade, Aubert, and Tarel [93] proposed the v-disparity image to identify the ground surface from a dense disparity map. They were able to detect the ground surface effectively assuming a negligible camera roll. Kramm and Bensrhair [120] proposed an approach for computing the v-disparity image from sparse point-based disparity maps obtaining promising results, but this approach loses the semantic meaning of the scene. Additionally, [155] proposed to use only the disparities at the local maxima in the intensities to obtain robustness to changes in illumination but this work used a laser-scanner to detect the obstacles.

For this thesis the v-disparity image is computed by taking as input chains of edge-disparities instead of a dense disparity image. The v-disparity image is computed as in [93] but only those pixels which correspond to an edge-disparity are taken into account. The use of edge-disparities translates into a speed up on the computation as only the edge pixels are processed in comparison to the use of a disparity image. For example, if 640×480 images are used, the disparity image would contain $307,200$ pixels to process whereas an edge-based disparity map has to process only a fraction of this. Figure 5-2 shows an example of the obtained v-disparity images.

It is assumed that the road is flat in order to validate stixel estimation but the approach could be extended to non-flat roads similarly to [93]. The Hough transform is used to detect the ground plane on the v-disparity image as in [93]. This allows the computation of the

camera pitch θ_g and height h_o assuming negligible camera roll as in [93]. At the same time a limit is set on the height h_{MAX} of the possible obstacles as in [120].

5.2.2 3D Points Segmentation

After the ground plane has been identified the 3D points are segmented by using the v-disparity image into one of three groups: ground plane, top plane or obstacles. Any edge-point located below the ground line on the v-disparity image plus an offset h_g in meters is taken as being part of the ground plane. Similarly any edge-point located above the top-plane line in the v-disparity image is taken as being part of the top-plane. Any other edge-point is assumed to be part of an obstacle.

After the obstacle-disparities have been identified, the u-disparity image is computed by using only the obstacle-disparities (see Section 2.3.1). This is the first time a u-disparity image has been computed by using only edge-points.

After the u-disparity image is computed, the labelling approach proposed by Iloie, Giosan, and Nedevschi to identify pedestrians [101] is used to segment the edge-disparities into obstacles. This labelling approach is an extension to connected components to allow the use of custom neighbourhoods for identifying the connected pixels. As the u-disparity image is a non-uniform representation of the depth, the custom size neighbourhoods are required for the identification of obstacles at different depth levels. A small neighbourhood is used for distant objects (small disparities) whereas a large one is used for close objects (large disparities). Figure 5-3 shows the used neighbourhoods which are similar to those used in [101].

As the u-disparity is obtained from the edge-disparities, the segmentation creates a relationship between the obstacles in the u-disparity and the image edges i.e. each edge gets assigned to an obstacle. This relationship allows the drawing of the obstacles on the input images as shown in Figure 5-4.

5.2.3 Stixels Computation

After the edge-points have been segmented into obstacles, the stixels are computed assuming every obstacle touches the ground plane. Although this assumption could produce false-positive obstacles, it allows the identification of obstacles which contain only a few disparities.

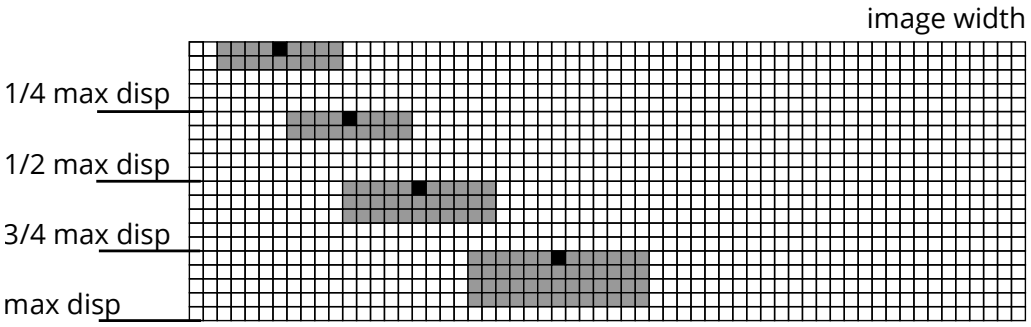


Figure 5-3: Neighbourhoods used for the labelling approach of SED. The neighbourhoods are taken from [101].

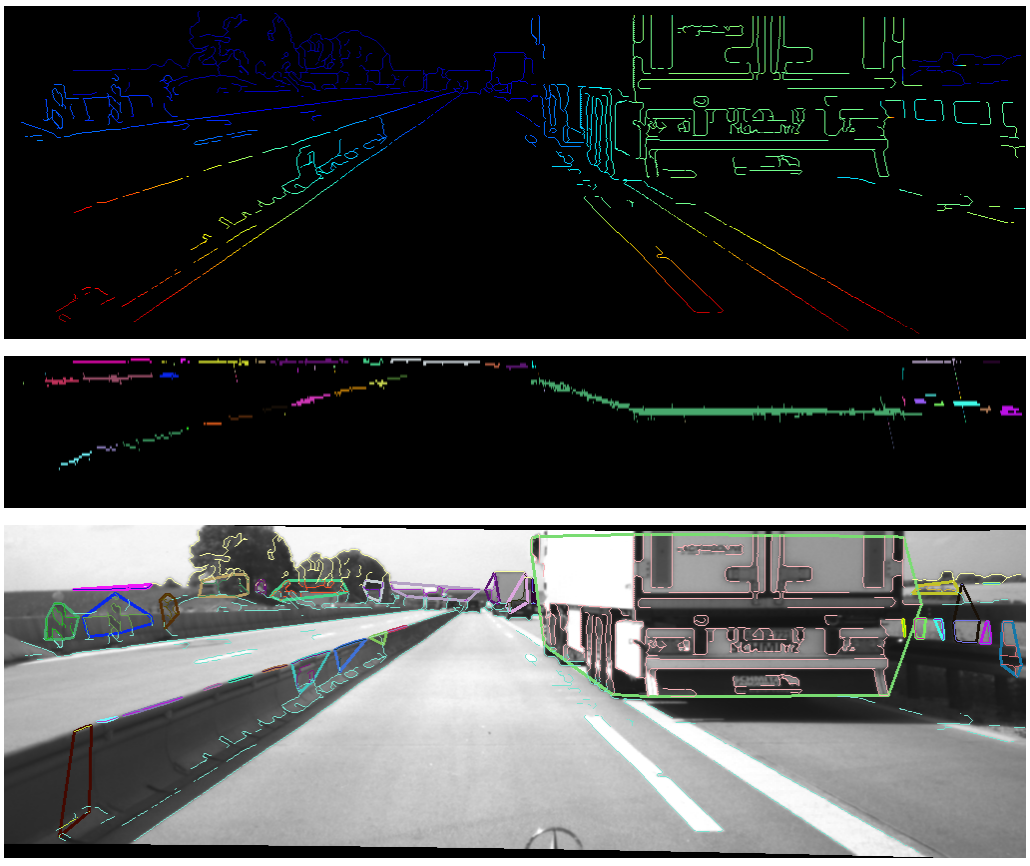


Figure 5-4: Edge-based disparity map (top) and the corresponding obstacle segmentation on the u-disparity (middle) using the labelling approach from [101]. Each colour of the segmentation image represents a different obstacle. As the u-disparity image is created from the edges, the segmentation creates a relationship between the segmented obstacles on the u-disparity and the image edges. This allows the drawing of each obstacle on the stereo-imagess (bottom).

The stixels are computed per obstacle by using all of the corresponding edge-points on one of the views of the stereo-camera. For the remainder of the chapter, the processing is performed using the left-view from the stereo-pair, but this could be performed on the right view without requiring any modification to the approach. Only obstacles with at least t_{op} 3D points are taken into account in order to remove possible noise and errors in the computation of the edge-disparity images.

The steps identifying the stixels are performed for each obstacle independently as follows:

1. Test if the obstacle has at least t_m points. If it does not, the obstacle is discarded and no further processing is performed.

2. Divide the left view of the stereo-camera into n_b column-bands b_i of width w_s. w_s is the width of the stixels.

3. Identify the column-bands b_i where each edge-point of the obstacle lies.

4. Compute the average disparity d_i for each column-band b_i.

5. For every image column in each column-band get the top row (y_t) for the edge-points.

6. Use the ground plane to compute row y_b which corresponds to disparity d_i.

7. Create a stixel for each column-band taking as disparity d_i. The top left coordinate is set as $(x_i, \bar{y_t})$ where x_i is the image coordinate where column-band b_i starts and $\bar{y_t}$ is the mean top coordinate per column-band. The height of the stixel is taken as the difference between y_b and y_t.

This procedure results in a list of stixels which are able to represent every obstacle. After all the obstacles have been processed, the stixels are drawn in the 2D view from the left camera. Figure 5-5 shows an example of the obtained stixels superimposed on the left-camera view.

5.3 Evaluation

The evaluation of the stixel estimation based on edge-disparities is performed on two datasets publicly available, the 6DVision stixels dataset [65] and the dataset provided

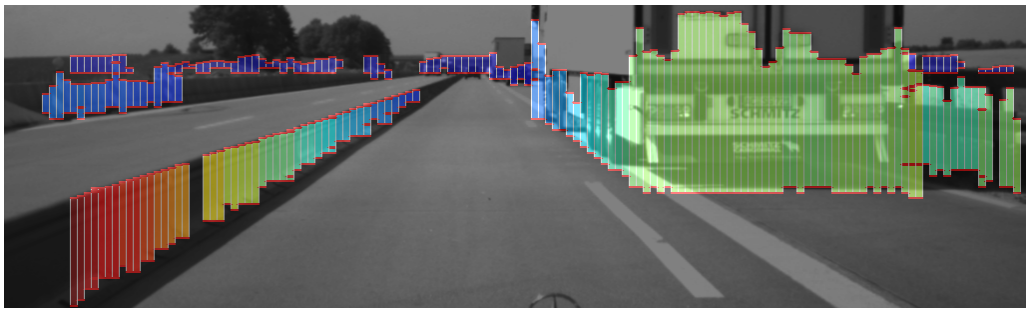


Figure 5-5: Sample view from the 6D Vision dataset with superimposed stixels on the left view. The stixels are colour coded according to disparity from high (red) to low (blue).

.

by [156] (AEss-Pedestrians) [156]. To the author's knowledge, the 6DVision dataset is the only one providing ground truth for stixels. It provides 8964 stereo pairs taken from car-on-road scenarios under different weather conditions such as sunny days, heavy rain and low lighting. The dataset proposed by Ess, Leibe, Schindler, *et al.* from the Bahnhof sequence provides 1000 stereo frames of crowded areas with manually annotated bounding boxes for pedestrians taken from a set-up similar to a baby trolley.

In order to obtain edge-disparities, the approach presented in Chapter 4 is used as it is able to produce chains of 3D points accurately and fast. The parameters for the proposed Edge Stixels are kept constant through the experimentation with the exception of the top plane height h_{MAX} which is set as $h_{MAX} = 2.5$ m for the 6DVision dataset and $h_{MAX} = 1.8$ is used for the AEss-Pedestrians dataset. This selection is according to the different medium for acquiring images and the kind of obstacles provided in the ground truth. The remaining parameters are: ground height $h_g = 0.15$ m to ignore small objects and noise in the ground plane estimation. In the ground plane estimation, the implementation used for the Hough Transform has three parameters, $rho, theta, threshold$ which correspond to the distance of the accumulator, the angle resolution and the accumulator threshold. The values are set as: $rho = 1\texttt{px}, theta = \pi/180, threshold = 30$ as these values are suggested by the author of the implementation and produced a small number of lines. Then the lines are sorted by the number of votes and the one with the most votes and positive slope is taken as the ground plane.

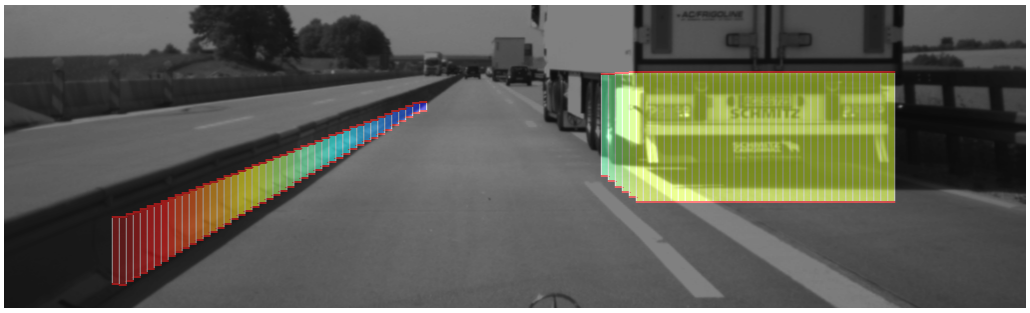


Figure 5-6: Sample ground truth provided by the 6DVision dataset. The provided stixels are only for certain regions of the image.

5.3.1 Evaluation on the 6DVision Dataset

Figure 5-6 shows an example of the ground truth stixels provided by the 6DVision dataset. It can be seen that the stixels are located at few image locations. As the approach provided in this thesis provides stixels for any possible obstacle found in the images, the stixels are filtered out in order to keep only those stixels which lie on the same columns as the ground-truth. Additionally, as it is possible to obtain more than one stixel per image column, only the ones with maximum disparity are kept for each column. As stated [65], the ground truth identifies the structures limiting the free space. Therefore false positive stixels would contain a disparity larger than that provided by the ground truth.

Unfortunately during the evaluation it was found that the 6DVision dataset provides disparities which might not be accurate. An example of this is shown in Figure 5-7. This figure shows the left and right views from the stereo pair and marks an easily distinguishable region of the image. The coordinates on the left image are $(828, 41)$ and in the right image are $(777, 41)$, this would result in a disparity of 51. Meanwhile in the provided ground truth, the disparity of the corresponding stixel is set to 44.23. This difference in the disparities could translate into a few meters in real world coordinates.

Due to these inaccuracies, this dataset was not used for the evaluation. If large error thresholds are used, the results would be misleading and if the error thresholds are small, it is not possible to discriminate between errors in the detection of the stixels or inaccuracies in the ground truth. As reference, Figure 5-5 shows an example of the stixels obtained by the approach in this thesis.

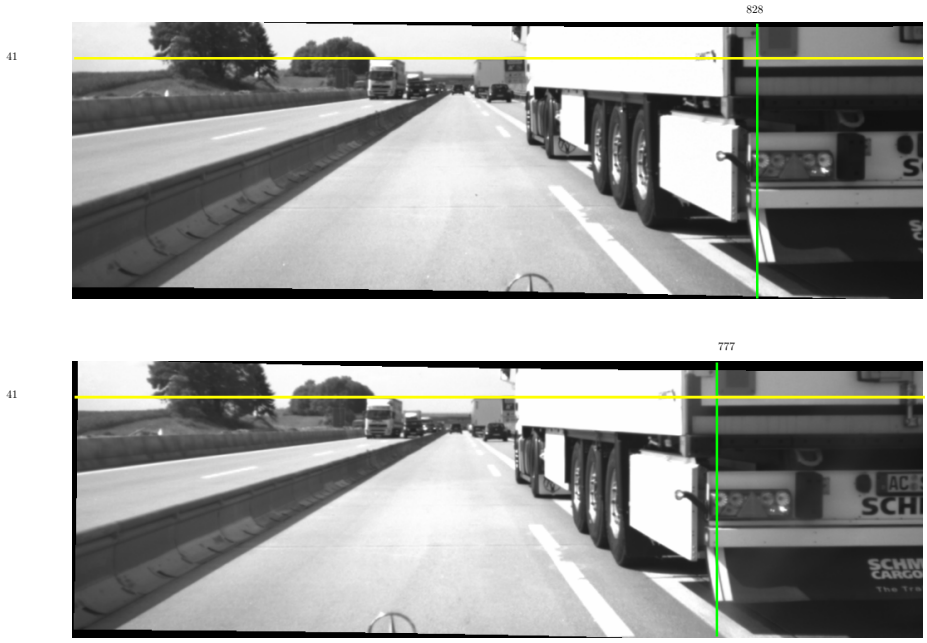
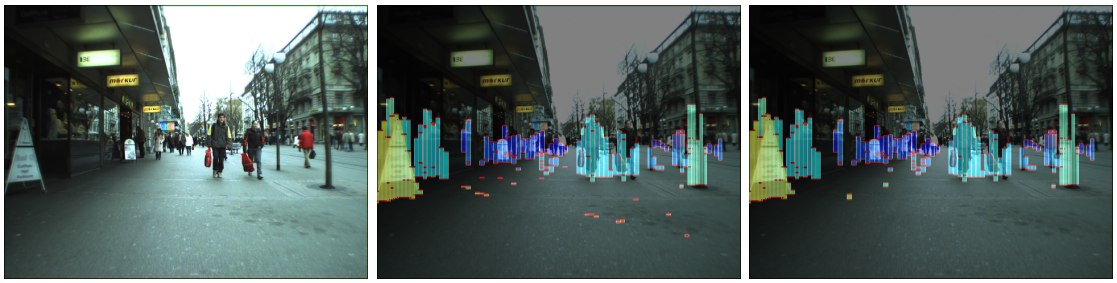


Figure 5-7: Example of a ground truth stixel with an inaccurate disparity. A highly distinguishable area of the image is used as example. The shown image corner has a disparity of approximately 51 whereas on the provided ground truth the corresponding stixel has a disparity of 44.23. This error would place the obstacle further away than it really is.

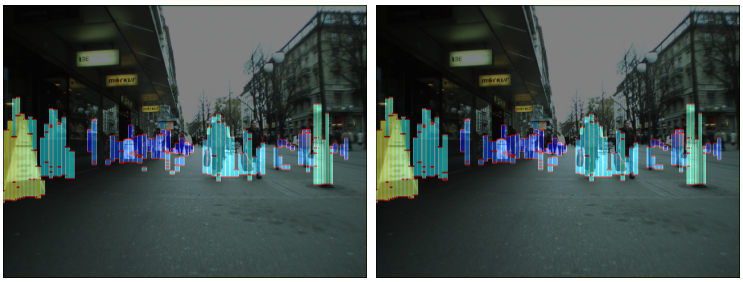
5.3.2 Evaluation on the AEss-Pedestrians Dataset

The stixels are evaluated on the AEss-Pedestrians dataset [156] using the Bahnhof sequence as in [115]. This dataset does not provide stixel information, instead it provides bounding boxes for pedestrians identified manually. In order to use this dataset for computing the detection rate of the approach from this thesis the following methodology is used: obstacles are marked as detected if the bottom of a stixel is found to coincide with the bottom of a ground-truth bounding box measured at the middle. This approach is the same as that used in [115].

The first step in the evaluation is the selection of the parameters for the computation of the stixels. The first parameter to be selected is threshold t_h which is the minimum height of an obstacle in meters. Figure 5-8 shows the effect of different values of t_h. This figure shows that at a minimum height of $t_h = 0m$ some ground pixels are taken as obstacles resulting in false obstacles due to inaccuracies in the estimation of the ground plane. In order to take into account these inaccuracies a value $t_h = 0.10$ is selected as it does not impose a heavy assumption on the minimum height of the obstacles for the dataset. In



(a) Left view. (b) $t_h = 0.00$ m. (c) $t_h = 0.05$ m.



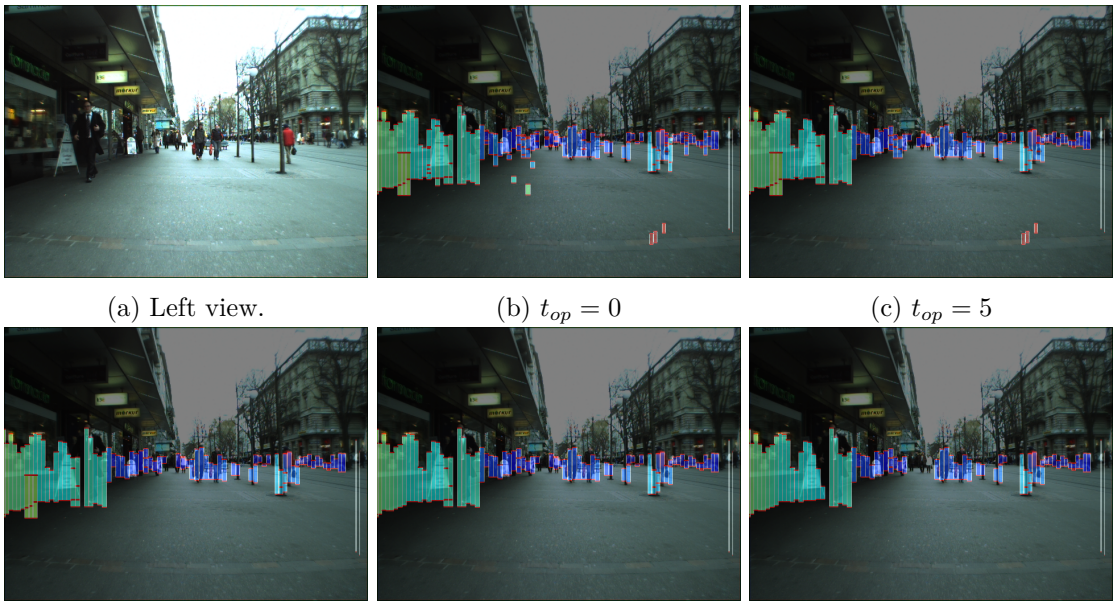
(d) $t_h = 0.10$ m. (e) $t_h = 0.15$ m.

Figure 5-8: Stixels computed for different values of the threshold t_h. Small values of t_h are more sensitive to inaccuracies in the disparities and the detection of the ground plane resulting in the creation of non-existent obstacles.

general in the autonomous cars environment there is a distance larger than $0.1m$ between the car bottom and the ground.

The remaining parameter to tune in Edge-Stixels is the minimum number of points per obstacle t_{op}. Figure 5-9 and Figure 5-10 shows the effect of the threshold t_{op} on the detected stixels. Figure 5-9 illustrates how the number of false stixels decreases as the threshold t_{op} is increased. Figure 5-10 shows the detection rates for different values of t_{op} at various levels of the absolute error between the bottom of the bounding boxes (provided by the ground truth) and the stixels. Figure 5-10 shows that this approach is able to get a recall above 90% with a threshold slightly below 20 pixels, whereas the approach from [115] is able to get 90% at a threshold of 30 pixels (see Figure 5-11). This means that the stixels obtained by the approach of this thesis are closer to the ones provided by the ground truth. At an error threshold of 30 (which is selected as the best in [115]) the approach from this thesis is able to get recall rates larger than 95% whereas the approach of [115] gets only 90%.

It is assumed that the larger number of edge pixels per obstacle, the less likely the edge pixels to be a product of aberrations. Therefore by increasing the minimum number of edge pixels per obstacle it is expected that the number of false detected stixels is decreased.



(a) Left view. (b) $t_{op} = 0$ (c) $t_{op} = 5$

(d) $t_{op} = 10$ (e) $t_{op} = 15$ (f) $t_{op} = 20$

Figure 5-9: Effect of the threshold t_{op} in the detection of false positive stixels. Large values reduce the number of stixels created by small edges. Please refer to Figure 5-10 for the error measurements at the different values of t_{op}.

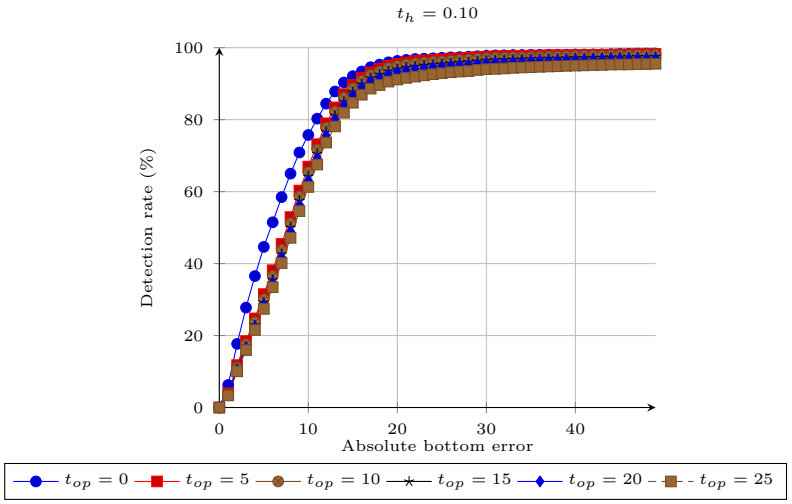


Figure 5-10: Rate of detected obstacles per absolute bottom error for different values for the minimum point count per obstacle t_m.

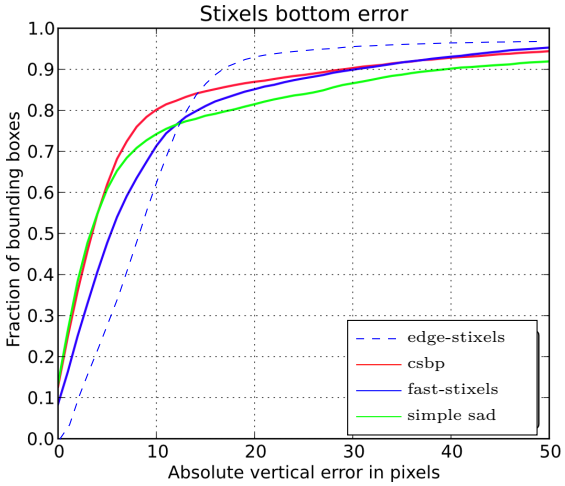


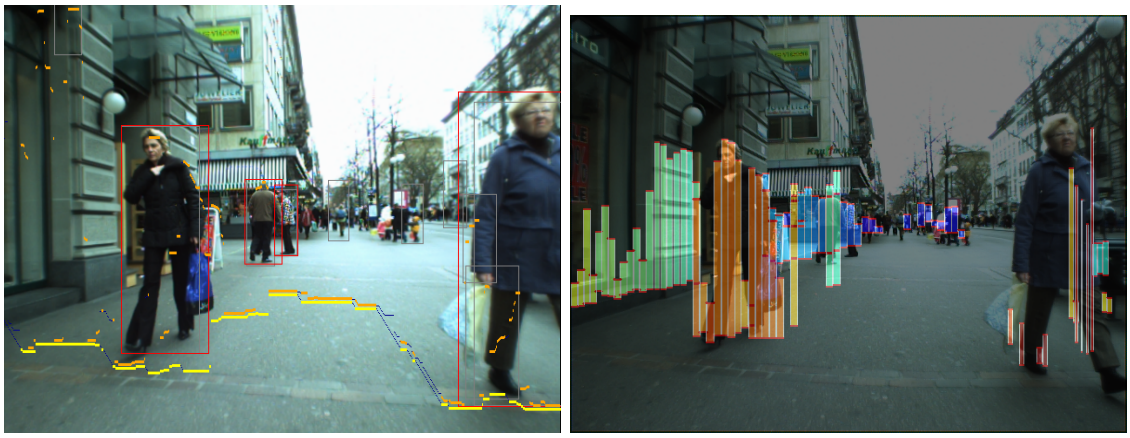
Figure 5-11: Recall for detection of bounding boxes using different error thresholds for the bottom boundary of the bounding boxes. Figure taken from [115] with the obtained results (edge-stixels) overlapped. The recall of the proposed Edge-Stixels goes higher at a smaller error threshold.

On the other hand, this increase in the minimum number of obstacle points would result in a decrease in the recall as fewer stixels are being created. A trade-off must be applied between the minimum number of obstacle points and the recall. A value of $t_{op} = 20$ still produces a recall larger than 95% at an error level of 30 pixels which is the baseline in [115], therefore this value is selected for the remainder of the evaluation.

Figure 5-11 shows a comparison of the recall of the approach proposed in this thesis and the results from [115]. It can be seen that the approach in this thesis is able to obtain a higher recall rate at a lower error level. This is a desired quality as it means that the proposed stixels are closer to the ground truth in comparison with the approach from [115] for the AEss-Pedestrians.

The addition of a threshold for the minimum height for the obstacles adds robustness to errors in the detection of the ground plane which results in the creation of zero-height stixels. Figure 5-12 shows an example of these produced zero-height stixels. Figure 5-12a shows zero-height stixels produced by the approach in [115]. As this approach does not support multiple obstacles on the same column, the detection of zero height stixels results in non-detected obstacles. Figure 5-12b shows that by applying a minimum height to the obstacles these zero-height obstacles are avoided. Even if present, the approach in this thesis is able to detect more than one obstacle per column therefore true obstacles are still recovered.

As mentioned in Section 5.2.2 the obstacle segmentation keeps the connectivity rela-



(a) (b)

Figure 5-12: Example of erroneous stixels detected by [115] are correctly identified by the approach in this thesis. a) Figure taken from [115], it shows erroneous stixels with zero-height (adjacent orange and yellow lines). b) No zero-height stixels are found in the proposed approach. Obstacles on the boundaries are detected as long as edges with disparities are present. For the lady on the b) it is detected as obstacle even though there are stixels only on some sections of her body.

Stages	Time (ms.)
Edge disparities	258.7
3D Point segmentation	9.8
Stixel Computation	0.2

Table 5.1: Average timing for each of the stages of the Edge-Stixels on the AEss dataset [156].

tionship of the edges, therefore it is possible to represent this segmentation on the input images as shown in Figure 5-13.

Table 5.1 shows the timing for each of the stages of the proposed stixels computation. The timing is performed on a standard laptop using only one core @ 2.1 GHz. No SIMD instructions are used in the implementation. This table shows the average over the entire AEss dataset which consists of 999 stereo-images. The computation of the edge-disparities is the slowest part as expected, whereas the obstacle segmentation and stixels computation runs in less than 10ms.

5.3.2.1 Discussion

The AEss-pedestrians dataset presents challenging environments for a robot navigating on a foot-path among pedestrians. The proposed approach was shown to obtain a higher accuracy and recall in comparison to the approach from [115] which represents the state-of-the-art.

Figure 5-13: Sample image of the AEss-pedestrians datasets with the segmentation resulting from the approach in this thesis.

Compared to the approach in [115], the approach proposed in this thesis has the following strengths:

- Support for more than one stixel per image column. In the approach presented by [115] it is assumed there is only one stixel per image column. This assumption becomes a problem when zero-height stixels are detected as in the examples in Figure 5-12. Zero-height stixels would result in possible obstacles not being detected whereas in the approach of this thesis, even though zero-height or close to zero-height stixels may be obtained, any other obstacles located in the same image column as the zero-height are detected.

- Partial dependence on the accuracy of the identification of the ground plane. The approach from [115] relies on the proper identification of the ground plane for computing the disparity of the stixels and their location. In contrast, the approach in this thesis uses the ground plane to filter-out ground pixels from the computation of stixels. If the ground plane is inaccurately detected in the proposed approach, it results in the creation of false stixels with small height which may be identified and

discarded by applying a threshold on the minimum height of the obstacles.

- No dependence on GPUs of FPGAs for fast processing. The approach in this thesis relies on the indirect use of geometric information for speeding up the processing, thus avoiding the computation of unneeded data. In contrast, although the approach in [115] also tries to minimise the used information, it still requires the solving of several DP problems which take into account all of the image pixels and therefore they require the use of GPU and multi-cores for achieving fast processing. The approach in [65] relies on the use of FPGAs for computing the disparity maps at high speed. This lack of a dependence on GPUs or FPGAs for fast processing enables the easy implementation of the approach proposed in this thesis on an embedded system where the resources are limited. It is important to note that the approach from this thesis could also take advantage of GPUs or FPGAs, if available, providing a further increase in speed but these implementations are out of the scope of this thesis.

After analysing the obtained results, the main sources of error of the proposed approach were identified as:

- Inaccuracies on the input edge-disparities. Inaccurate disparities obtained by the stereo-matching would result in "flying" objects resulting in the creation of false-positive stixels. Although the threshold t_m reduces the effect of these wrong disparities, some of those false-positive stixels are still present. These aberrations do not appear over a sequence of frames, therefore tracking would help to reduce them significantly.

- Inaccurate detection of the ground plane. The ground plane is used to determine the bottom boundary of the stixels. For some cases, the line fitted to the ground is not accurate, resulting in the incorporation of ground edges as obstacle edges. By assuming the geometry of the ground changes smoothly (high frame rates would result in small changes in the ground) it would be possible to carry information between sequential frames in order to discard wrong estimations of the ground plane.

5.4 Discussion

This chapter introduced a new approach for computing stixels from sparse edge-based disparities. The proposed approach showed a higher recall rate and a higher accuracy on the AEss dataset [156] compared to the approach from Benenson, Timofte, and Van Gool which represents the current state-of-the-art. This increase in accuracy and recall is obtained by the use of edge-disparities as they are able to accurately represent the obstacle boundaries.

Additionally the proposed approach showed that the obstacle segmentation and stixel computation reached a performance of 100Hz on a standard laptop without the use of GPUs, multiple cores and SIMD instructions. This speed-up is a product of the use of only edge information for the processing of the data.

The proposed compact and fast-to-compute approach allows a reduction in the amount of resources required to classify the image contents into obstacles or free area while also increasing the accuracy and recall against other state-of-the-art approaches without requiring the use of GPUs or FPGAs. Therefore it could be expected that by using any of those tools the speed of the system could be further increased.

The obtained results show that by using edge-disparities only it is possible to reduce the amount of required storage and processing as the amount of processed information is lower than for approaches based on dense disparity maps.

Future work would include: a) the tracking of the obstacles to produce an estimate of their displacement; b) reducing the number of false stixels created by inaccuracies in the input edge-disparities and c) modifying the ground plane detector to allow curved surfaces.

6

Conclusions and Future Work

This thesis proposed the detection of obstacles for navigation using edges from images as backbone. The obtained results show that the edges in the stereo-imagess provide enough information to estimate the distance of the objects in the scene and to classify their projection as obstacle or free space. Figure 6-1 shows the pipeline of the proposed approach. First the calibrated and rectified stereo-images are used to extract the edge-based disparities. Then the obtained chains of edge-disparities are used to draw the stixels over the input images. This allows the identification of the obstacle and free space on the input images, segmentation of the objects, detection of the ground plane and provides an edge based representation of the world. The remaining of the chapter details how the contributions of this thesis fit into this output.

The proposed approach turned out to have both a higher accuracy and recall compared to other state-of-the-art approaches for obstacle detection at a higher speed while requiring

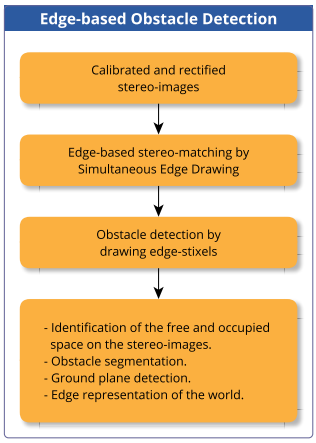


Figure 6-1: Pipeline of the proposed obstacle detection based on image-edges.

a smaller amount of resources. By using illumination-robust pixel descriptors and the image edges, the obtained approach inherits robustness against changes in illumination. Additionally, as no assumption is imposed on the shape or location of the obstacles the obtained approach is able to identify any kind of obstacle which could be found in the real world.

It is important to note that no re-projection is used in the proposed approach, therefore errors associated to it do not affect the performance of the obstacle detection. If required, the obtained edge-representation could be translated into real world coordinates by using the intrinsic and extrinsic parameters of the camera, resulting in a 3D edge representation of the world.

The obtained results rely on the efficient representation of the image contents by the use of image edges as they keep the amount of information to the minimum while keeping the image semantics. The following paragraphs review the results obtained at each stage of the creation of this novel edge-based approach and its application for obstacle detection. Finally, the directions for further research are discussed.

6.1 Summary of Contributions

This thesis proposes three main novel contributions: the extension of the Census Transform to incorporate edge information; the creation of a new approach for computing 3D curves from stereo-images and the identification of the obstacles in this 3D curve world. A secondary contribution is the evaluation of the performance of the Complete Rank Transform in the stereo-matching context as up to now it has been used only for the calculation of Optical Flow. An additional minor contribution is the creation of libraries that implement the proposed algorithms and other tools required by the thesis.

6.1.1 New Pixel Descriptors for Stereo-Matching, Chapter 3

The first contribution from this thesis is the extension of the Census Transform to incorporate a similarity measure allowing the identification of pixels around the edges. This similarity measure behaves like a threshold on the gradient resulting in the creation of zero-strings for pixels which are not located around the image edges. This information is used in a local stereo-matching approach to reduce the test candidates without incurring

a significant increase in computation.

A secondary contribution is the evaluation of the Complete Rank Transform in a local stereo-matching approach. This image transform was shown to be able to increase the accuracy of the computed disparity maps (dense) at the cost of increasing the computations.

6.1.2 Disparity by Simultaneous Edge Drawing, Chapter 4

The second major contribution is the design of a new approach to extracting 3D curves from stereo-images. The core of this approach is the extension of Edge Drawing, an efficient and accurate edge detector, to run simultaneously across the stereo-images. By doing this, only a few pixels require to be tested in a local stereo-matching approach and then the image gradient is used to propagate the disparities along the image edges while they are localized and connected.

This approach showed it was able to reduce the number of computations at the cost of increasing the overall error by about 2% against the best performing approach on the KITTI 2015 benchmark. As the computation of the disparities is an intermediate step in the overall obstacle detector, the obstacle detection stage is able to diminish the effect of this slightly reduced accuracy.

6.1.3 Edge Based Stixels, Chapter 5

The third major contribution is the creation of an approach to identify the obstacles in the stereo-images by computing stixels from a 3D curve representation of the world. This approach was shown to be very fast to compute without using SIMD, GPUs or multiple cores as only 3D curves are processed. Additionally this approach was shown to be able to identify obstacles with an accuracy higher than other state-of-the-art obstacle detectors.

The proposed approach is also able to provide segmentation of the 3D curves into obstacles using only spatial information. This information is very valuable for classification tasks such as pedestrian detection.

6.1.4 Further Research

The success of using the edges to identify obstacles using a stereo-camera showed that by carefully selecting the information to be processed it is possible to speed up the process and

reduce the required resources without sacrificing the accuracy. In the following paragraphs some problems that would benefit from this compact and efficient representation are:

6.1.4.1 Autonomous Navigation

The identification of the obstacles is one of the first steps required to achieve autonomous navigation. By applying the proposed approaches in this thesis the following problems may be solved:

- **Obstacle tracking.** As one of the by-products of the proposed approaches is the obstacle segmentation, this information could be used in the sequences of frames obtained from the stereo-camera to track the obstacles.
- **Visual Odometry.** During the obstacle detection stage the 3D curves are classified as belonging to the ground plane or above it. These 3D curves located on the ground plane could be used to match a sequence of frames and estimate the displacement of the camera.
- **Simultaneous Localization and Mapping (SLAM):** Once the displacement of the camera in time is known, the obtained 3D curves could be merged to create a 3D curve representation of the world. By using this compact 3D curve representation the amount of resources required to create large scale maps are decreased in comparison to the use of dense point clouds.
- **Obstacle classification.** As the edges are able to keep the image semantics, they could be used along with the obtained segmentation of the obstacles to classify them by using approaches similar to HOG-SVM which is based on the usage of gradient information. Other approaches could use the edge information along with CNN to perform classification.

6.1.4.2 Improvements to the Proposed Approaches

Although the proposed approaches were able to increase computation speed and reduce resources, there is still some room for improvement. The main areas of improvement per proposed algorithm are:

- Thresholded Census Transform

  - **Adaptive similarity measure.** Information from the image could be used in order to improve its response to the presence of high or low gradient areas in the image.
  - **Use of sparse or non-square windows for performing the transform.** Currently only square windows are used to compute the image transform. Windows with different geometries could be used in order to reduce even further the number of computations and reduce the redundant operations in a stereo-matching process.

- Simultaneous Edge Drawing:
  - **Use confidence information during the disparity propagation.** Currently no confidence information is used during the propagation of the disparity from the anchor points. This information could be used to reduce even further the possibility of following the wrong edge during the simultaneous smart routing.
  - **Edge validation.** The validation of edges by the Helmholtz principle may help to reduce the number of texture edges as it does for a revised version of the Edge Drawing algorithm.
  - **Curve fitting.** Currently chains of points are used to describe the 3D curves. Curve fitting or sub-sampling could be used to reduce the number of points used to represent the curves reducing even further the required storage at a possible cost of increasing the number of computations.
  - **Improved use of the Scale Space.** Currently the computation of the Scale Space for the identification of the anchors is the most computationally expensive stage of the approach. An alternate way to compute this would significantly reduce the required computations.

- Edge-based Stixels
  - **Non-planar ground plane.** The current approach for identifying the ground plane assumes it is planar in the neighbourhood of the ground robot. This assumption could be problem for surfaces found in rural areas. By fitting a

curve instead of a line to the ground profile on the v-disparity image it would be possible to represent a wider range of ground surfaces.

- **Different neighbourhoods for labelling.** Currently the neighbourhoods used on the labelling approach are fixed. Different shapes could be used in order to better represent the non-linear correspondence of the u-disparity image to the real world.
- **Tracking.** By tracking the identified obstacles it may be possible to identify erroneous 3D curves and discard them from further processing.
- **Extension to flying robots.** Currently the ground plane is used to estimate the camera pitch and height. If this information could be obtained from other means, the 3D curve information could be used to identify the object boundaries instead of relying on the assumption that they touch the ground.

Glossary

u-disparity Image that represents an histogram obtained from a disparity image where each row corresponds to a row in the disparity image, the columns correspond to the disparity values and the intensities correspond to the number of occurrences of each disparity value per row. 41, 42, 46, 115, 133

v-disparity Image that represents an histogram obtained from a disparity image where each column corresponds to a column in the disparity image, the rows correspond to the disparity values and the intensities correspond to the number of occurrences of each disparity value per column. 41, 42, 45, 46, 114, 115, 133

aggregation window Support region used to aggregate the matching cost of pixels around p and p' in the left and right image respectively. See Equation (2.26). 30, 31, 58–67, 70, 71, 73, 77

cost volume 3D image where the intensity represents the matching cost of pixel (x, y) at disparity d. 66

depth Difference in the z-coordinates between a 3D point in the space around the robot and the stereo-camera measured in real world coordinates. 11, 12, 23, 26, 27, 38, 44–46, 48, 96, 115

disparity map Image where the intensity represents the disparity of a pixel (x, y) referenced on one of the images in the stereo-images. 4, 8, 13, 21, 29, 31, 34, 36–40, 42, 44, 45, 50, 55–57, 59, 61, 66, 68, 73–75, 77, 78, 81–83, 97, 99, 100, 102, 110, 112–114, 126, 130

stereo-camera Camera pair aligned horizontally with their capture sensors laying on the same plane. 11, 12, 24, 44, 45, 134, 135

stereo-images Synchronized images captured by a stereo-camera. 7–14, 17, 18, 21, 24, 25, 29, 30, 57, 82–88, 102, 110, 116, 124, 128–130, 134

stixel Vertical rectangle with an associated disparity drawn over the images captured by the stereo-camera. 45–48, 112–114, 117–127

Bibliography

[1] D. Gallup, "Efficient 3d reconstruction of large-scale urban environments from street-level video," p. 123, 2011, [Online]. Available: http://dc.lib.unc.edu/cdm/ref/collection/etd/id/3559.

[2] S. Sivaraman and M. M. Trivedi, "Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773–1795, Dec. 2013, ISSN: 1524-9050, DOI: 10.1109/TITS.2013.2266661.

[3] K. H. Lin, C. H. Chang, A. Dopfer, and C. C. Wang, "Mapping and Localization in 3D Environments Using a 2D Laser Scanner and a Stereo Camera," *Journal of Information Science and Engineering*, vol. 28, no. 2012, pp. 131–144, 2012, [Online]. Available: http://www.iis.sinica.edu.tw/page/jise/2012/201201%7B%5C_%7D09.pdf.

[4] P. Pinggera, U. Franke, and R. Mester, "Highly Accurate Depth Estimation for Objects at Large Distances," in *35th German Conference, GCPR 2013, Saarbrücken, Germany, September 3-6, 2013. Proceedings*, ser. Lecture Notes in Computer Science, vol. 8142, Springer Berlin Heidelberg, 2013, pp. 21–30, DOI: 10.1007/978-3-642-40602-7_3.

[5] P. Pinggera, D. Pfeiffer, U. Franke, and R. Mester, "Know Your Limits: Accuracy of Long Range Stereoscopic Object Measurements in Practice," in *13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II*, 2014, pp. 96–111, DOI: 10.1007/978-3-319-10605-2_7.

[6] A. Wedel, U. Franke, J. Klappstein, T. Brox, and D. Cremers, "Realtime Depth Estimation and Obstacle Detection from Monocular Video," in *Pattern Recognition*, 2006, pp. 475–484, ISBN: 978-3-540-44414-5, DOI: 10.1007/11861898_48.

[7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, p. 15, 2015, DOI: 10.1109/TRO.2015.2463671arXiv: 1502.00956.

[8] J. M. Carranza, "Efficient Monocular SLAM by Using a Structure-Driven Mapping," PhD thesis, University of Bristol, May 2012, [Online]. Available: http://www.cs.bris.ac.uk/Publications/Papers/2001577.pdf.

[9] J. Zhang, G. Kantor, M. Bergerman, and S. Singh, "Monocular visual navigation of an autonomous vehicle in natural scene corridor-like environments," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3659–3666, Oct. 2012, DOI: 10.1109/IROS.2012.6385479.

[10] R. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Second. Cambridge University Press, ISBN: 0521540518, 2004, ISBN: 0521540518.

[11] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, T. Graf, and R. Schmidt, "Driving environment perception using stereovision," *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2005, pp. 331–336, 2005, DOI: 10.1109/IVS.2005.1505124.

[12] J. Gluckman and S. K. Nayar, "Catadioptric Stereo Using Planar Mirrors," *International Journal of Computer Vision*, vol. 44, no. 1, pp. 65–79, 2001, ISSN: 09205691, DOI: 10.1023/A:1011172403203.

[13] J. Gluckman and S. Nayar, "Rectified catadioptric stereo sensors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 224–236, 2002, ISSN: 01628828, DOI: 10.1109/34.982902.

[14] M. El-gayar, H. Soliman, and N. Meky, "A comparative study of image low level feature extraction algorithms," *Egyptian Informatics Journal*, vol. 14, no. 2, pp. 175–181, Jul. 2013, ISSN: 11108665, DOI: 10.1016/j.eij.2013.06.003.

[15] S. Gauglitz, T. Höllerer, and M. Turk, "Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking," *International Journal of Computer Vision*, vol. 94, no. 3, pp. 335–360, Sep. 2011, ISSN: 0920-5691, DOI: 10.1007/s11263-011-0431-5.

[16] M. Al-Shahri and A. Yilmaz, "Line Matching in Wide-Baseline Stereo: A Top-Down Approach," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 1–1, Jun. 2014, ISSN: 1057-7149, DOI: 10.1109/TIP.2014.2331147.

[17] D. Rao, S.-J. Chung, and S. Hutchinson, "CurveSLAM: An approach for vision-based navigation without point features," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Oct. 2012, pp. 4198–4204, ISBN: 978-1-4673-1736-8, DOI: 10.1109/IROS.2012.6385764.

[18] M. Hofer, M. Donoser, and H. Bischof, "Semi-Global 3D Line Modeling for Incremental Structure-from-Motion," in *Proceedings of the British Machine Vision Conference*, M. Valstar, A. French, and T. Pridmore, Eds., BMVA Press, 2014, DOI: 10.13140/2.1.4591.7443.

[19] T. Koletschka, L. Puig, and K. Daniilidis, "MEVO: Multi-environment stereo visual odometry," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Sep. 2014, pp. 4981–4988, ISBN: 978-1-4799-6934-0, DOI: 10.1109/IROS.2014.6943270.

[20] H. Trinh, "Efficient Stereo Algorithm using Multiscale Belief Propagation on Segmented Images," in *Procedings of the British Machine Vision Conference 2008*, British Machine Vision Association, 2008, pp. 33.1–33.10, ISBN: 1-901725-36-7, DOI: 10.5244/C.22.33.

[21] H. Bay, V. Ferrari, and L. Van Gool, "Wide-Baseline Stereo Matching with Line Segments," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, IEEE, 2005, pp. 329–336, ISBN: 0-7695-2372-2, DOI: 10.1109/CVPR.2005.375.

[22] R. Fabbri and B. Kimia, "3D curve sketch: Flexible curve-based stereo reconstruction and calibration," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2010, pp. 1538–1545, ISBN: 978-1-4244-6984-0, DOI: 10.1109/CVPR.2010.5539787.

[23] C. Pal, A. Chakrabarti, and R. Ghosh, "A Brief Survey of Recent Edge-Preserving Smoothing Algorithms on Digital Images," *Computer Vision and Pattern Recogni-*

tion, vol. 00, pp. 1–40, Mar. 2015, arXiv: 1503.07297v1[Online]. Available: http://arxiv.org/abs/1503.07297.

[24] J. Mohan, V. Krishnaveni, and Y. Guo, "A survey on the magnetic resonance image denoising methods," *Biomedical Signal Processing and Control*, vol. 9, no. 1, pp. 56–69, Jan. 2014, ISSN: 17468094, DOI: 10.1016/j.bspc.2013.10.007.

[25] H. Hirschmuller and D. Scharstein, "Evaluation of Stereo Matching Costs on Images with Radiometric Differences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1582–1599, Sep. 2009, ISSN: 0162-8828, DOI: 10.1109/TPAMI.2008.221.

[26] H. Hirschmüller, P. R. Innocent, and J. Garibaldi, "Real-Time Correlation-Based Stereo Vision with Reduced Border Errors," *International Journal of Computer Vision*, vol. 47, no. 1/3, pp. 229–246, 2002, ISSN: 09205691, DOI: 10.1023/A:1014554110407.

[27] S. Nedevschi, R. Schmidt, R. Danescu, D. Frentiu, T. Marita, T. Graf, F. Oniga, and C. Pocol, "High accuracy stereo vision system for far distance obstacle detection," in *IEEE Intelligent Vehicles Symposium, 2004*, IEEE, 2004, pp. 292–297, ISBN: 0-7803-8310-9, DOI: 10.1109/IVS.2004.1336397.

[28] J. Witt and U. Weltin, "Sparse stereo by edge-based search using dynamic programming," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012, pp. 3631–3635, ISBN: 9784990644109, [Online]. Available: http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=6460951.

[29] Fusiello, "Efficient Stereo with Multiple Windowing Andrea," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 14, no. 8, p. 1053, 2000, ISSN: 02180014, DOI: 10.1016/S0218-0014(00)00069-6.

[30] G. Saygili, L. van der Maaten, and E. A. Hendriks, "Adaptive stereo similarity fusion using confidence measures," *Computer Vision and Image Understanding*, vol. 135, pp. 95–108, Jun. 2015, ISSN: 10773142, DOI: 10.1016/j.cviu.2015.02.005.

[31] F. Schaffalitzky and A. Zisserman, "Viewpoint invariant texture matching and wide baseline stereo," in *Proceedings Eighth IEEE International Conference on Computer*

Vision. ICCV 2001, vol. 2, IEEE Comput. Soc, 2001, pp. 636–643, ISBN: 0-7695-1143-0, DOI: 10.1109/ICCV.2001.937686.

[32] Y. Hou, J. Yao, B. Zhou, and Y. Liu, "Fusing color and texture features for stereo matching," in *2013 IEEE International Conference on Information and Automation (ICIA)*, IEEE, Aug. 2013, pp. 620–625, ISBN: 978-1-4799-1334-3, DOI: 10.1109/ICInfA.2013.6720371.

[33] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Computer Vision - ECCV '94*, ser. Lecture Notes in Computer Science, J.-O. Eklundh, Ed., vol. 801, Berlin/Heidelberg: Springer-Verlag, 1994, pp. 151–158, ISBN: 978-3-540-57957-1, DOI: 10.1007/BFb0028345.

[34] O. Demetz, D. Hafner, and J. Weickert, "The complete rank transform: A tool for accurate and morphologically invariant matching of structures," in *Proceedings of the 2013 British Machine Vision Conference*, Bristol: BMVA Press, 2013, [Online]. Available: http://www.mia.uni-saarland.de/Publications/demetz-bmvc13.pdf.

[35] B. Froba and A. Ernst, "Face detection with the modified census transform," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, IEEE, 2004, pp. 91–96, ISBN: 0-7695-2122-3, DOI: 10.1109/AFGR.2004.1301514.

[36] F. Stein, "Efficient Computation of Optical Flow Using the Census Transform," in *Pattern Recognition*, ser. Lecture Notes in Computer Science, vol. 3175, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 79–86, ISBN: 978-3-540-22945-2, DOI: 10.1007/b99676.

[37] D. Bhat and S. Nayar, "Ordinal measures for visual correspondence," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Comput. Soc. Press, 1996, pp. 351–357, ISBN: 0-8186-7258-7, DOI: 10.1109/CVPR.1996.517096.

[38] Hui Ding, Mengyin Fu, and Meiling Wang, "Shift-Invariant Contourlet Transform and Its Application to Stereo Matching," in *First International Conference on Inno-*

vative Computing, Information and Control - Volume I (ICICIC'06), vol. 3, IEEE, 2006, pp. 87–90, ISBN: 0-7695-2616-0, DOI: 10.1109/ICICIC.2006.519.

[39] G. Egnal, "Mutual Information as a Stereo Correspondence Measure," Tech. Rep. January, 2000, p. 113, [Online]. Available: http://repository.upenn.edu/cis%7B%5C_%7Dreports/113/.

[40] Junhwan Kim, Kolmogorov, and Zabih, "Visual correspondence using energy minimization and mutual information," in *Proceedings Ninth IEEE International Conference on Computer Vision*, IEEE, 2003, 1033–1040 vol.2, ISBN: 0-7695-1950-4, DOI: 10.1109/ICCV.2003.1238463.

[41] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, Feb. 2008, ISSN: 0162-8828, DOI: 10.1109/TPAMI.2007.1166.

[42] G. Kuschk and D. Cremers, "Fast and Accurate Large-Scale Stereo Reconstruction Using Variational Methods," in *2013 IEEE International Conference on Computer Vision Workshops*, IEEE, Dec. 2013, pp. 700–707, ISBN: 978-1-4799-3022-7, DOI: 10.1109/ICCVW.2013.96.

[43] M. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, Dec. 2005, ISSN: 1057-7149, DOI: 10.1109/TIP.2005.859376.

[44] W. S. Fife and J. K. Archibald, "Improved Census Transforms for Resource-Optimized Stereo Vision," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 1, pp. 60–73, Jan. 2013, ISSN: 1051-8215, DOI: 10.1109/TCSVT.2012.2203197.

[45] M. Humenberger, C. Zinner, M. Weber, W. Kubinger, and M. Vincze, "A fast stereo matching algorithm suitable for embedded real-time systems," *Computer Vision and Image Understanding*, vol. 114, no. 11, pp. 1180–1202, Nov. 2010, ISSN: 10773142, DOI: 10.1016/j.cviu.2010.03.012.

[46] S.-c. Pei and Y.-y. Wang, "Color invariant census transform for stereo matching algorithm," in *2013 IEEE International Symposium on Consumer Electronics*

(ISCE), IEEE, Jun. 2013, pp. 209–210, ISBN: 978-1-4673-6199-6, DOI: 10.1109/ISCE.2013.6570188.

[47] S. Perri, P. Corsonello, and G. Cocorullo, "Adaptive Census Transform: A novel hardware-oriented stereovision algorithm," *Computer Vision and Image Understanding*, vol. 117, no. 1, pp. 29–41, Jan. 2013, ISSN: 10773142, DOI: 10.1016/j.cviu.2012.10.003.

[48] C. Zinner, M. Humenberger, K. Ambrosch, and W. Kubinger, "An Optimized Software-Based Implementation of a Census-Based Stereo Matching Algorithm," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, PART 1, G. Bebis, R. Boyle, B. Parvin, D. Koracin, P. Remagnino, F. Porikli, J. Peters, J. Klosowski, L. Arns, Y. K. Chun, T.-M. Rhyne, and L. Monroe, Eds., vol. 5358 LNCS, 2008, pp. 216–227, ISBN: 3540896384, DOI: 10.1007/978-3-540-89639-5_21.

[49] J. Meltzer and S. Soatto, "Edge descriptors for robust wide-baseline correspondence," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2008, pp. 1–8, ISBN: 978-1-4244-2242-5, DOI: 10.1109/CVPR.2008.4587684.

[50] A. Geiger, M. Roser, and R. Urtasun, "Efficient Large-Scale Stereo Matching," in *Proceedings of the 10th Asian conference on Computer vision - Volume Part I*, ser. ACCV'10, Berlin, Heidelberg: Springer-Verlag, 2011, pp. 25–38, ISBN: 978-3-642-19314-9, DOI: 10.1007/978-3-642-19315-6_3.

[51] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 794–805, Oct. 2013, ISSN: 10473203, DOI: 10.1016/j.jvcir.2013.05.006.

[52] Wei Wei and King Ngi Ngan, "Disparity estimation with edge-based matching and interpolation," in *2005 International Symposium on Intelligent Signal Processing and Communication Systems*, vol. 2, IEEE, 2005, pp. 153–156, ISBN: 0-7803-9266-3, DOI: 10.1109/ISPACS.2005.1595369.

[53] M. Tomono, "Robust 3D SLAM with a stereo camera based on an edge-point ICP algorithm," in *2009 IEEE International Conference on Robotics and Automation*, IEEE, May 2009, pp. 4306–4311, ISBN: 978-1-4244-2788-8, DOI: 10.1109/ROBOT.2009.5152529.

[54] J. Witt and U. Weltin, "Robust Real-Time Stereo Edge Matching by Confidence-Based Refinement," in *Intelligent Robotics and Applications*, ser. Lecture Notes in Computer Science, C.-Y. Su, S. Rakheja, and H. Liu, Eds., vol. 7508, Springer Berlin Heidelberg, 2012, pp. 512–522, ISBN: 978-3-642-33502-0, DOI: 10.1007/978-3-642-33503-7_50.

[55] H. H. Baker and T. O. Binford, "Depth from Edge and Intensity Based Stereo," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, pp. 631–636, [Online]. Available: http://dl.acm.org/citation.cfm?id=1623264.1623271.

[56] F. Mokhtarian and S. Abbasi, "Affine Curvature Scale Space with Affine Length Parametrisation," *Pattern Analysis & Applications*, vol. 4, no. 1, pp. 1–8, Mar. 2001, ISSN: 1433-7541, DOI: 10.1007/PL00010984.

[57] F. Mai and Y. S. Hung, "3D Curves Reconstruction from Multiple Images," *2010 International Conference on Digital Image Computing: Techniques and Applications*, pp. 462–467, Dec. 2010, DOI: 10.1109/DICTA.2010.84.

[58] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and Xiaopeng Zhang, "On building an accurate stereo matching system on graphics hardware," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, IEEE, Nov. 2011, pp. 467–474, ISBN: 978-1-4673-0063-6, DOI: 10.1109/ICCVW.2011.6130280.

[59] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, "High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth," in *Pattern Recognition*, ser. Lecture Notes in Computer Science 1, X. Jiang, J. Hornegger, and R. Koch, Eds., vol. 8753, Berlin, Heidelberg: Springer International Publishing, 2014, pp. 31–42, ISBN: 978-3-319-11751-5, DOI: 10.1007/978-3-319-11752-2_3.

[60] M. Lhuillier and Long Quan, "Match propagation for image-based modeling and rendering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1140–1146, Aug. 2002, ISSN: 0162-8828, DOI: 10.1109/TPAMI.2002.1023810.

[61] F. Cheng, H. Zhang, D. Yuan, and M. Sun, "Stereo matching by using the global edge constraint," *Neurocomputing*, vol. 131, pp. 217–226, May 2014, ISSN: 09252312, DOI: 10.1016/j.neucom.2013.10.022.

[62] 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, no. 4, pp. 401–406, Apr. 1998, ISSN: 01628828, DOI: 10.1109/34.677269.

[63] Xiaoyan Hu and P. Mordohai, "A Quantitative Evaluation of Confidence Measures for Stereo Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2121–2133, Nov. 2012, ISSN: 0162-8828, DOI: 10.1109/TPAMI.2012.46.

[64] X. Hu and P. Mordohai, "Evaluation of stereo confidence indoors and outdoors," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2010, pp. 1466–1473, ISBN: 978-1-4244-6984-0, DOI: 10.1109/CVPR.2010.5539798.

[65] D. Pfeiffer, S. Gehrig, and N. Schneider, "Exploiting the Power of Stereo Confidences," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2013, pp. 297–304, ISBN: 978-0-7695-4989-7, DOI: 10.1109/CVPR.2013.45.

[66] M. Gong, R. Yang, L. Wang, and M. Gong, "A Performance Study on Different Cost Aggregation Approaches Used in Real-Time Stereo Matching," *International Journal of Computer Vision*, vol. 75, no. 2, pp. 283–296, Feb. 2007, ISSN: 0920-5691, DOI: 10.1007/s11263-006-0032-x.

[67] R. Haeusler and R. Klette, "Analysis of KITTI Data for Stereo Analysis with Stereo Confidence Measures," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, PART

2, vol. 7584 LNCS, 2012, pp. 158–167, ISBN: 9783642338670, DOI: 10.1007/978-3-642-33868-7_16.

[68] G. Egnal, M. Mintz, and R. P. Wildes, "A stereo confidence metric using single view imagery with comparison to five alternative approaches," *Image and Vision Computing*, vol. 22, no. 12 SPEC. ISS. Pp. 943–957, 2004, ISSN: 02628856, DOI: 10.1016/j.imavis.2004.03.018.

[69] D. Scharstein and R. Szeliski, "Stereo Matching with Nonlinear Diffusion," *International Journal of Computer Vision*, vol. 28, no. 2, pp. 155–174, 1998, ISSN: 09205691, DOI: 10.1023/A:1008015117424.

[70] R. Haeusler and R. Klette, "Disparity Confidence Measures on Engineered and Outdoor Data," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, ser. Lecture Notes in Computer Science, L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo, Eds., vol. 7441, Springer Berlin Heidelberg, 2012, pp. 624–631, ISBN: 978-3-642-33274-6, DOI: 10.1007/978-3-642-33275-3_77.

[71] P. Mordohai, "The self-aware matching measure for stereo," *Proceedings of the IEEE International Conference on Computer Vision*, no. Iccv, pp. 1841–1848, 2009, ISSN: 1550-5499, DOI: 10.1109/ICCV.2009.5459409.

[72] K.-J. Yoon and I. S. Kweon, "Distinctive Similarity Measure for stereo matching under point ambiguity," *Comp Vis Imag Under*, vol. 112, no. 2, pp. 173–183, 2008, ISSN: 1077-3142, DOI: DOI:10.1016/j.cviu.2008.02.003.

[73] N. Sabater, A. Almansa, and J.-M. Morel, "Meaningful Matches in Stereovision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 930–942, May 2012, ISSN: 0162-8828, DOI: 10.1109/TPAMI.2011.207arXiv: 1112.1187.

[74] P. F. Felzenszwalb and R. Zabih, "Dynamic Programming and Graph Algorithms in Computer Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 721–740, Apr. 2011, ISSN: 0162-8828, DOI: 10.1109/TPAMI.2010.135.

[75] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Belief Propagation for Early Vision," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 41–54, Oct. 2006, ISSN: 0920-5691, DOI: 10.1007/s11263-006-7899-4.

[76] K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun, "Continuous Markov Random Fields for Robust Stereo Estimation," in *Computer Vision – ECCV 2012*, ser. Lecture Notes in Computer Science, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., vol. 7576, Springer Berlin Heidelberg, Apr. 2012, pp. 45–58, ISBN: 978-3-642-33714-7, DOI: 10.1007/978-3-642-33715-4_4arXiv: 1204.1393.

[77] R. Scharstein, Daniel and Szeliski, D. Scharstein, R. Szeliski, and R. Zabih, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *International Journal of Computer Vision*, vol. 47, no. 1, pp. 7–42, 2002, ISSN: 0920-5691, DOI: 10.1023/A:1014573219977.

[78] M. Sarkis and K. Diepold, "Sparse stereo matching using belief propagation," in *2008 15th IEEE International Conference on Image Processing*, IEEE, 2008, pp. 1780–1783, ISBN: 978-1-4244-1765-0, DOI: 10.1109/ICIP.2008.4712121.

[79] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001, ISSN: 01628828, DOI: 10.1109/34.969114arXiv: åDĄäÿÃæœĹ.

[80] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, "Fast Approximate Energy Minimization with Label Costs," *International Journal of Computer Vision*, vol. 96, no. 1, pp. 1–27, Jan. 2012, ISSN: 0920-5691, DOI: 10.1007/s11263-011-0437-z.

[81] S. Gehrig, M. Reznitskii, N. Schneider, U. Franke, and J. Weickert, "Priors for Stereo Vision under Adverse Weather Conditions," in *2013 IEEE International Conference on Computer Vision Workshops*, IEEE, Dec. 2013, pp. 238–245, ISBN: 978-1-4799-3022-7, DOI: 10.1109/ICCVW.2013.39.

[82] Y. Ohta and T. Kanade, "Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*,

vol. PAMI-7, no. 2, pp. 139–154, Mar. 1985, ISSN: 0162-8828, DOI: 10.1109/TPAMI.1985.4767639.

[83] R. Ranftl, S. Gehrig, T. Pock, and H. Bischof, "Pushing the limits of stereo using variational stereo estimation," in *2012 IEEE Intelligent Vehicles Symposium*, IEEE, Jun. 2012, pp. 401–407, ISBN: 978-1-4673-2118-1, DOI: 10.1109/IVS.2012.6232171.

[84] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2012, pp. 3354–3361, ISBN: 978-1-4673-1228-8, DOI: 10.1109/CVPR.2012.6248074.

[85] Y. Swirski, Y. Y. Schechner, and T. Nir, "Variational stereo in dynamic illumination," in *2011 International Conference on Computer Vision*, IEEE, Nov. 2011, pp. 1124–1131, ISBN: 978-1-4577-1102-2, DOI: 10.1109/ICCV.2011.6126360.

[86] L. Bagnato, P. Frossard, and P. Vandergheynst, "Optical flow and depth from motion for omnidirectional images using a TV-L1 variational framework on graphs," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, IEEE, Nov. 2009, pp. 1469–1472, ISBN: 978-1-4244-5653-6, DOI: 10.1109/ICIP.2009.5414552.

[87] H. Hirschmuller, "Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, IEEE, 2005, pp. 807–814, ISBN: 0-7695-2372-2, DOI: 10.1109/CVPR.2005.56.

[88] X. Sun, X. Mei, S. Jiao, M. Zhou, Z. Liu, and H. Wang, "Real-time local stereo via edge-aware disparity propagation," *Pattern Recognition Letters*, vol. 49, pp. 201–206, Nov. 2014, ISSN: 01678655, DOI: 10.1016/j.patrec.2014.07.010.

[89] A. Gonçalves, A. Godinho, and Joao Sequeira, "Low cost sensing for autonomous car driving in highways," in *ICINCO-RA*, vol. 2, 2007, pp. 370–377, [Online]. Available: http://www.researchgate.net/publication/221645687%7B%5C_%7DLow%7B%5C_%7Dcost%7B%5C_%7Dsensing%7B%5C_%7Dfor%7B%5C_%7Dautonomous%

7B%5C_%7Dcar%7B%5C_%7Ddriving%7B%5C_%7Din%7B%5C_%7Dhighways/file/9c96051c2dd5565b46.pdf.

[90] T. Pollard and J. L. Mundy, "Change Detection in a 3-d World," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2007, pp. 1–6, ISBN: 1-4244-1179-3, DOI: 10.1109/CVPR.2007.383073.

[91] K. Sebesta and J. Baillieul, "Animal-inspired agile flight using optical flow sensing," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, IEEE, Dec. 2012, pp. 3727–3734, ISBN: 978-1-4673-2066-5, DOI: 10.1109/CDC.2012.6426163arXiv: arXiv:1203.2816v1.

[92] Yu-Chen Lin, Che-Tsung Lin, Wei-Cheng Liu, and Long-Tai Chen, "A vision-based obstacle detection system for parking assistance," in *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, IEEE, Jun. 2013, pp. 1627–1630, ISBN: 978-1-4673-6322-8, DOI: 10.1109/ICIEA.2013.6566629.

[93] R. Labayrade, D. Aubert, and J.-P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation," in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 2, IEEE, 2002, pp. 646–651, ISBN: 0-7803-7346-4, DOI: 10.1109/IVS.2002.1188024.

[94] Z. Hu and K. Uchimura, "U-V-disparity: an efficient algorithm for stereovision based scene analysis," in *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, vol. 2005, IEEE, 2005, pp. 48–54, ISBN: 0-7803-8961-1, DOI: 10.1109/IVS.2005.1505076.

[95] A. Broggi, C. Caraffi, R. Fedriga, and P. Grisleri, "Obstacle Detection with Stereo Vision for Off-Road Vehicle Navigation," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, vol. 3, IEEE, 2005, pp. 65–65, ISBN: 0-7695-2372-2, DOI: 10.1109/CVPR.2005.503.

[96] N. Soquet, M. Perrollaz, D. Aubert, N. Soquet, M. Perrollaz, D. Aubert, and F. Space, "Free Space Estimation for Autonomous Navigation," in *5th International Conference on Computer Vision Systems*, 2007, ISBN: 9783000209338, [Online]. Available: https://hal.inria.fr/hal-00780658.

[97] M. Perrollaz, J.-D. Yoder, A. Spalanzani, and C. Laugier, "Using the disparity space to compute occupancy grids from stereo-vision," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Oct. 2010, pp. 2721–2726, ISBN: 978-1-4244-6674-0, DOI: 10.1109/IROS.2010.5649690.

[98] M. Perrollaz, J.-D. Yoder, A. Negre, A. Spalanzani, and C. Laugier, "A Visibility-Based Approach for Occupancy Grid Computation in Disparity Space," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1383–1393, Sep. 2012, ISSN: 1524-9050, DOI: 10.1109/TITS.2012.2188393.

[99] B. Musleh, D. Martin, A. de la Escalera, and J. M. Armingol, "Visual ego motion estimation in urban environments based on U-V disparity," in *2012 IEEE Intelligent Vehicles Symposium*, vol. 1, IEEE, Jun. 2012, pp. 444–449, ISBN: 978-1-4673-2118-1, DOI: 10.1109/IVS.2012.6232183.

[100] N. Fakhfakh, D. Gruyer, and D. Aubert, "Weighted V-disparity approach for obstacles localization in highway environments," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, Jun. 2013, pp. 1271–1278, ISBN: 978-1-4673-2755-8, DOI: 10.1109/IVS.2013.6629641.

[101] A. Iloie, I. Giosan, and S. Nedevschi, "UV disparity based obstacle detection and pedestrian classification in urban traffic scenarios," in *2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)*, IEEE, Sep. 2014, pp. 119–125, ISBN: 978-1-4799-6569-4, DOI: 10.1109/ICCP.2014.6936963.

[102] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 3, pp. 249–265, Jun. 1987, ISSN: 0882-4967, DOI: 10.1109/JRA.1987.1087096.

[103] M. Yguel, O. Aycard, and C. Laugier, "Efficient GPU-based Construction of Occupancy Girds Using several Laser Range-finders," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Oct. 2006, pp. 105–110, ISBN: 1-4244-0258-1, DOI: 10.1109/IROS.2006.281817.

[104] L. Matthies and A. Elfes, "Integration of sonar and stereo range data using a grid-based representation," in *Proceedings. 1988 IEEE International Conference*

on Robotics and Automation, IEEE Comput. Soc. Press, 1988, pp. 727–733, ISBN: 0-8186-0852-8, DOI: 10.1109/ROBOT.1988.12145.

[105] M. Brandao, R. Ferreira, K. Hashimoto, J. Santos-Victor, and A. Takanishi, "Integrating the whole cost-curve of stereo into occupancy grids," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Nov. 2013, pp. 4681–4686, ISBN: 978-1-4673-6358-7, DOI: 10.1109/IROS.2013.6697030.

[106] M. Perrollaz, A. Spalanzani, and D. Aubert, "Probabilistic representation of the uncertainty of stereo-vision and application to obstacle detection," in *2010 IEEE Intelligent Vehicles Symposium*, IEEE, Jun. 2010, pp. 313–318, ISBN: 978-1-4244-7866-8, DOI: 10.1109/IVS.2010.5548010.

[107] H. Badino, U. Franke, and R. Mester, "Free Space Computation Using Stochastic Occupancy Grids and Dynamic Programming," in *Iccvw*, 2007, pp. 1–12.

[108] Z. Zhang, "A stereovision system for a planetary rover: calibration, correlation, registration, and fusion," *Machine Vision and Applications*, vol. 10, no. 1, pp. 27–34, May 1997, ISSN: 0932-8092, DOI: 10.1007/s001380050056.

[109] F. Oniga and S. Nedevschi, "Processing Dense Stereo Data Using Elevation Maps: Road Surface, Traffic Isle, and Obstacle Detection," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 3, pp. 1172–1182, Mar. 2010, ISSN: 0018-9545, DOI: 10.1109/TVT.2009.2039718.

[110] A. Vatavu, R. Danescu, and S. Nedevschi, "Stereovision-Based Multiple Object Tracking in Traffic Scenarios Using Free-Form Obstacle Delimiters and Particle Filters," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 498–511, Feb. 2015, ISSN: 1524-9050, DOI: 10.1109/TITS.2014.2366248.

[111] M. Vergauwen, M. Pollefeys, and L. Van Gool, "A stereo-vision system for support of planetary surface exploration," *Machine Vision and Applications*, vol. 14, no. 1, pp. 5–14, Apr. 2003, ISSN: 0932-8092, DOI: 10.1007/s00138-002-0097-7.

[112] H. Badino, U. Franke, and D. Pfeiffer, "The Stixel World - A Compact Medium Level Representation of the 3D-World," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in*

Bioinformatics), vol. 5748 LNCS, 2009, pp. 51–60, ISBN: 3642037976, DOI: 10.1007/978-3-642-03798-6_6.

[113] J. Rebut, G. Toulminet, and A. Bensrhair, "Road obstacles detection using a self-adaptive stereo vision sensor: a contribution to the ARCOS french project," in *IEEE Intelligent Vehicles Symposium, 2004*, IEEE, 2004, pp. 738–743, ISBN: 0-7803-8310-9, DOI: 10.1109/IVS.2004.1336476.

[114] S. Kubota, T. Nakano, and Y. Okamoto, "A Global Optimization Algorithm for Real-Time On-Board Stereo Obstacle Detection Systems," in *Proc. IEEE Intelligent Vehicles Symposium*, IEEE, Jun. 2007, pp. 7–12, ISBN: 1-4244-1067-3, DOI: 10.1109/IVS.2007.4290083.

[115] R. Benenson, R. Timofte, and L. Van Gool, "Stixels estimation without depth map computation," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, IEEE, Nov. 2011, pp. 2010–2017, ISBN: 978-1-4673-0063-6, DOI: 10.1109/ICCVW.2011.6130495.

[116] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Fast Stixel Computation for Fast Pedestrian Detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, PART 3, vol. 7585 LNCS, 2012, pp. 11–20, ISBN: 9783642338847, DOI: 10.1007/978-3-642-33885-4_2.

[117] T. Williamson and C. Thorpe, "A specialized multibaseline stereo technique for obstacle detection," in *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*, IEEE Comput. Soc, 1997, pp. 238–244, ISBN: 0-8186-8497-6, DOI: 10.1109/CVPR.1998.698615.

[118] R. Shade and P. Newman, "Discovering and mapping complete surfaces with stereo," in *2010 IEEE International Conference on Robotics and Automation*, IEEE, May 2010, pp. 3910–3915, ISBN: 978-1-4244-5038-1, DOI: 10.1109/ROBOT.2010.5509337.

[119] M. Bleyer, C. Rhemann, and C. Rother, "Extracting 3D Scene-Consistent Object Proposals and Depth from Stereo Images," in *Computer Vision – ECCV 2012*, ser. Lecture Notes in Computer Science, A. Fitzgibbon, S. Lazebnik, P. Perona, Y.

Sato, and C. Schmid, Eds., vol. 7576, Springer Berlin Heidelberg, 2012, pp. 467–481, ISBN: 978-3-642-33714-7, DOI: 10.1007/978-3-642-33715-4_34.

[120] S. Kramm and A. Bensrhair, "Obstacle detection using sparse stereovision and clustering techniques," in *2012 IEEE Intelligent Vehicles Symposium*, IEEE, Jun. 2012, pp. 760–765, ISBN: 978-1-4673-2118-1, DOI: 10.1109/IVS.2012.6232283.

[121] F. Erbs, A. Barth, and U. Franke, "Moving vehicle detection by optimal segmentation of the dynamic stixel world," *Intelligent Vehicle Symposium (IV)*, no. Iv, pp. 951–956, 2011, ISSN: 1931-0587, DOI: 10.1109/IVS.2011.5940532.

[122] A. Barth and U. Franke, "Estimating the driving state of oncoming vehicles from a moving platform using stereo vision," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 560–571, 2009, ISSN: 15249050, DOI: 10.1109/TITS.2009.2029643.

[123] S. Bota and S. Nedevschi, "Vision based obstacle tracking in urban traffic environments," in *2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing*, IEEE, Aug. 2011, pp. 231–238, ISBN: 978-1-4577-1479-5, DOI: 10.1109/ICCP.2011.6047874.

[124] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and Tracking the Driving Environment With a Particle-Based Occupancy Grid," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1331–1342, Dec. 2011, ISSN: 1524-9050, DOI: 10.1109/TITS.2011.2158097.

[125] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proceedings of Imaging Understanding Workshop*, vol. 130, pp. 121–129, 1981, DOI: 10.1.1.49.2019.

[126] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, Aug. 1981, ISSN: 00043702, DOI: 10.1016/0004-3702(81)90024-2.

[127] S. Vedula, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 475–480, Mar. 2005, ISSN: 0162-8828, DOI: 10.1109/TPAMI.2005.63.

[128] K. Yamaguchi, D. McAllester, and R. Urtasun, "Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation," in *Computer Vision–ECCV 2014*, ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8693, Cham: Springer International Publishing, 2014, pp. 756–771, ISBN: 978-3-319-10601-4, DOI: 10.1007/978-3-319-10602-1_49.

[129] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2015, pp. 3061–3070, ISBN: 978-1-4673-6964-0, DOI: 10.1109/CVPR.2015.7298925.

[130] Z. Talai and A. Talai, "A fast edge detection using fuzzy rules," *2011 International Conference on Communications, Computing and Control Applications (CCCA)*, pp. 1–5, Mar. 2011, DOI: 10.1109/CCCA.2011.6031499.

[131] M. Tomono, "3D localization based on visual odometry and landmark recognition using image edge points," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Oct. 2010, pp. 5953–5959, ISBN: 978-1-4244-6674-0, DOI: 10.1109/IROS.2010.5649853.

[132] N. Kazakova, M. Margala, and N. Durdle, "Sobel edge detection processor for a real-time volume rendering system," in *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*, vol. 2, IEEE, 2004, pp. II–913–16, ISBN: 0-7803-8251-X, DOI: 10.1109/ISCAS.2004.1329421.

[133] C. Lopez-Molina, B. De Baets, H. Bustince, J. Sanz, and E. Barrenechea, "Multiscale edge detection based on Gaussian smoothing and edge tracking," *Knowledge-Based Systems*, vol. 44, pp. 101–111, 2013, ISSN: 09507051, DOI: 10.1016/j.knosys.2013.01.026.

[134] A. F. Bobick and S. S. Intille, "Large occlusion stereo," *International Journal of Computer Vision*, vol. 33, no. 3, pp. 181–200, 1999, ISSN: 09205691, DOI: 10.1023/A:1008150329890.

[135] D. Hafner, O. Demetz, and J. Weickert, "Why Is the Census Transform Good for Robust Optic Flow Computation?" In *Scale Space and Variational Methods in Computer Vision*, ser. Lecture Notes in Computer Science, vol. 7893, Berlin,

Heidelberg: Springer Berlin Heidelberg, 2013, pp. 210–221, ISBN: 978-3-642-38266-6, DOI: 10.1007/978-3-642-38267-3.

[136] G. Bradski, "The OpenCV Library," *Dr Dobbs Journal of Software Tools*, vol. 25, pp. 120–125, 2000, ISSN: 1044-789X, DOI: 10.1111/0023-8333.50.s1.10.

[137] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 1, pp. I–195 – I–202, 2003, ISSN: 1063-6919, DOI: 10.1109/CVPR.2003.1211354.

[138] H. Hirschmuller and D. Scharstein, "Evaluation of Cost Functions for Stereo Matching," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2007, pp. 1–8, ISBN: 1-4244-1179-3, DOI: 10.1109/CVPR.2007.383248.

[139] D. Scharstein and C. Pal, "Learning Conditional Random Fields for Stereo," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2007, pp. 1–8, ISBN: 1-4244-1179-3, DOI: 10.1109/CVPR.2007.383191.

[140] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986, ISSN: 0162-8828, DOI: 10.1109/TPAMI.1986.4767851.

[141] C. Akinlar and C. Topal, "Edpf: a Real-Time Parameter-Free Edge Segment Detector With a False Detection Control," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, no. 01, p. 1 255 002, Feb. 2012, ISSN: 0218-0014, DOI: 10.1142/S0218001412550026.

[142] C. Topal and C. Akinlar, "Edge Drawing: A combined real-time edge and segment detector," *Journal of Visual Communication and Image Representation*, vol. 23, no. 6, pp. 862–872, Aug. 2012, ISSN: 10473203, DOI: 10.1016/j.jvcir.2012.05.004.

[143] D. E. Knuth, *The Art of Computer Programming. Volume III: Sorting and Searching*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1998, ISBN: 0-201-03803-X.

[144] C. Schmid and A. Zisserman, "Automatic line matching across views," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern*

Recognition, IEEE Comput. Soc, 1997, pp. 666–671, ISBN: 0-8186-7822-4, DOI: 10.1109/CVPR.1997.609397.

[145] C. Topal, C. Akinlar, and Y. Genc, "Edge Drawing: A Heuristic Approach to Robust Real-Time Edge Detection," *2010 20th International Conference on Pattern Recognition*, pp. 2424–2427, Aug. 2010, DOI: 10.1109/ICPR.2010.593.

[146] D. Yuan, F. Cheng, and H. Zhang, "Dense stereo matching based on edge constraint and variable windows," in *2011 IEEE International Conference on Robotics and Biomimetics*, IEEE, Dec. 2011, pp. 1912–1917, ISBN: 978-1-4577-2138-0, DOI: 10.1109/ROBIO.2011.6181570.

[147] D. Teney and J. Piater, "Sampling-Based Multiview Reconstruction without Correspondences for 3D Edges," in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, IEEE, Oct. 2012, pp. 160–167, ISBN: 978-0-7695-4873-9, DOI: 10.1109/3DIMPVT.2012.28.

[148] F. Bergholm, "Edge Focusing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 6, pp. 726–741, Nov. 1987, ISSN: 0162-8828, DOI: 10.1109/TPAMI.1987.4767980.

[149] Farhan, *Counting Sort Algorithm*, V. Pieterse and P. E. Black, Eds., 2012, [Online]. Available: https://scholar.google.cz/scholar?hl=en%7B%5C&%7Dq=Dictionary+of+Algorithms+and+Data+Structures%7B%5C&%7DbtnG=%7B%5C&%7Das%7B%5C_%7Dsdt=1,5%7B%5C&%7Das%7B%5C_%7Dsdtp=%7B%5C#%7D0%20http://www.code2learn.com/2012/01/counting-sort-algorithm-with-example.html.

[150] F. Devernay, "A Non-Maxima Suppression Method for Edge Detection with Sub-Pixel Accuracy," INRIA, Tech. Rep., 1995, DOI: 10.1.1.35.1863.

[151] J. Witt and U. Weltin, "Robust stereo visual odometry using iterative closest multiple lines," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4164–4171, Nov. 2013, DOI: 10.1109/IROS.2013.6696953.

[152] M. Hofer, A. Wendel, and H. Bischof, "Incremental Line-based 3D Reconstruction using Geometric Constraints," in *Procedings of the British Machine Vision*

Conference 2013, British Machine Vision Association, 2013, pp. 92.1–92.11, ISBN: 1-901725-49-9, DOI: 10.5244/C.27.92.

[153] D. Pfeiffer and U. Franke, "Efficient representation of traffic scenes by means of dynamic stixels," in *2010 IEEE Intelligent Vehicles Symposium*, IEEE, Jun. 2010, pp. 217–224, ISBN: 978-1-4244-7866-8, DOI: 10.1109/IVS.2010.5548114.

[154] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian detection at 100 frames per second," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2012, pp. 2903–2910, ISBN: 978-1-4673-1228-8, DOI: 10.1109/CVPR.2012.6248017.

[155] R. Labayrade, D. Gruyer, C. Royere, M. Perrollaz, and D. Aubert, "Obstacle Detection Based on Fusion Between Stereovision and 2D Laser Scanner," in *Mobile Robots: Perception & Navigation*, February, S. Kolski, Ed., Pro Literatur Verlag, 2007, ISBN: 3-86611-283-1, [Online]. Available: http://hal.inria.fr/hal-00683758/%20http://hal.inria.fr/hal-00683758.

[156] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "A mobile vision system for robust multi-person tracking," *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, 2008, ISSN: 1063-6919, DOI: 10.1109/CVPR.2008.4587581.