FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Design and Evaluation of Stereo Matching Techniques for Silicon Retina Cameras

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der technischen Wissenschaften

by

Dipl.-Ing. (FH) Jürgen Kogler, MSc.
Registration Number 0826833

to the
Faculty of Informatics at the Vienna University of Technology

Advisor: Ao. Univ.Prof. Dipl.-Ing. Mag.rer.nat Dr.techn. Margrit Gelautz

The dissertation has been reviewed by:

(Ao. Univ.Prof. Dr. Margrit Gelautz)	(Ao. Univ.Prof. Dr. Josef Scharinger)

Wien, 29.02.2016

(Dipl.-Ing. (FH) Jürgen Kogler, MSc.)

Erklärung zur Verfassung der Arbeit

Dipl.-Ing. (FH) Jürgen Kogler, MSc.
Wilhelmstraße 27/8, 1120 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)	(Unterschrift Verfasser)

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Margrit Gelautz for the continuous support and motivation during my PhD study. Her guidance helped me throughout my research and during the writing of this thesis. Besides my supervisor, I would like to thank Josef Scharinger, from the Johannes Kepler University in Linz, who kindly agreed to review my PhD as a secondary supervisor, and enhanced the quality of this thesis with his insightful comments.

Special thanks to Florian Eibensteiner for being my companion in investigating the world of silicon retina sensors and having numerous meetings with me, resulting in many fruitful discussions. I would like to show my appreciation to Christoph Sulzbachner and Martin Humenberger, for their endless support in improving the quality of this PhD from different perspectives. Additionally, I would like to thank Wilfried Kubinger and Manfred Gruber, from the Austrian Institute of Technology, for offering me the opportunity to start this PhD and develop it within different projects. Furthermore, I want to thank my colleagues Mattias Schörghuber, Christian Zinner and Maria Satzinger for contributing to my work with their constructive feedback.

Finally, I want to express special gratitude to my wife Souzan, who handled every phase of this PhD.

Dedication

I would like to dedicate this scientific work
to *Armin Kinzl*, who was at the beginning
of his scientific career, but unfortunately
died in a tragic accident.
(*13 September 1986; †3 September 2009)

Widmung

Ich möchte diese wissenschafliche Arbeit
Armin Kinzl widmen, der am Beginn seiner
wissenschaflichen Karriere stand und
bedauerlicherweise bei einem tragischen
Unfall ums Leben gekommen ist.
(*13. September 1986; †3. September 2009)

Abstract

Nowadays, techniques for 3D reconstruction that are used in a variety of computer vision applications need to account for the 3D structure of a real-world scene. This task is often performed using a stereo vision system which consists of two digital cameras observing the same scene from two different viewing angles. A major challenge in stereo vision is the stereo matching problem, which involves finding corresponding pixels that are projections of the same scene point in the image pair. While stereo matching of images delivered by conventional cameras has been the subject of intense research for many years, this thesis focuses on the analysis of stereo data delivered by a different type of digital camera - a *Silicon Retina* sensor - whose stereo processing capabilities have been addressed by only few publications thus far.

The special analog pixel design of a silicon retina camera enables a high dynamic range of light and very fast pixel updates. Unlike a conventional camera, the silicon retina camera's sensor pre-processes the information on-chip, and only transmits pixels that capture a change of light. This significantly reduces the amount of data that must be transferred and processed. However, as the process yields visual information different to a normal digital image, the data poses new challenges for solving the correspondence problem occurring in a silicon retina stereo set-up. In this thesis, we first analyze the data from a silicon retina stereo sensor and study its behavior in order to assess the impact of various algorithms on this data. Then, based on these results, we design and implement new kinds of stereo matching algorithms to overcome the imposed challenges of silicon retina data. Besides the core stereo matching algorithms, we develop and evaluate different approaches to improve the accuracy of the stereo matching algorithms. Additionally, we design a method to generate ground truth data to better evaluate the calculated depth data; this enables meaningful discussions and interpretation of the generated stereo matching output.

Kurzfassung

Heutzutage werden in verschiedensten Anwendungen der Bildverarbeitung Techniken zur 3D-Rekonstruktion herangezogen, um die Tiefeninformationen der realen Umgebung abzubilden. Zu diesem Zweck werden häufig Stereo Vision Systeme, bestehend aus zwei Kameras, welche die Szene aus zwei unterschiedlichen Blickwinkeln aufnehmen, verwendet. Eine wesentliche Herausforderung in der Stereobildverarbeitung ist die Lösung des Korrespondenzproblems, welches darauf abzielt, korrespondierende Pixel - welche Projektionen des gleichen Punkts der 3D-Szene darstellen - im Bildpaar zu finden. Im Gegensatz zur Stereoverarbeitung von Bildern herkömmlicher Kameras, welche bereits seit vielen Jahren Gegenstand intensiver Forschung ist, konzentriert sich die vorliegende Arbeit auf die in der bisherigen Literatur kaum behandelte Analyse von Stereodaten, welche mit einer *Silicon Retina* Kamera aufgenommen wurden.

Das spezielle analoge Pixeldesign einer Silicon Retina Kamera ermöglicht einen hohen Dynamikbereich sowie schnelle Pixel Updates. Im Gegensatz zu konventionellen Kameras wird beim Silicon Retina Sensor eine Vorverarbeitung der Daten auf dem Chip durchgeführt, und Pixelinformationen werden nur dann weitergeleitet, wenn eine Helligkeitsänderung stattgefunden hat. Diese Funktionsweise reduziert die Menge an Daten, die transferiert und weiter verarbeitet werden muss. Aufgrund der unterschiedlichen Bildinformation im Vergleich zu normalen digitalen Kameras stellt ein Silicon Retina Stereosystem neue Herausforderungen an die Lösung des Korrespondenzproblems. In der vorliegenden Arbeit untersuchen wir zunächst die Daten eines Silicon Retina Stereosystems, um dann neue Stereo Matching Ansätze zu entwickeln, welche auf die Besonderheiten der Silicon Retina Daten zugeschnitten sind. Zusätzlich zu den eigentlichen Matching Algorithmen entwickeln und vergleichen wir verschiedene Verfahren zur Verbesserung der Tiefenergebisse. Des Weiteren wurde zur Evaluierung der berechneten 3D-Daten eine Methode zur Generierung von Referenzdaten (Ground Truth) entwickelt, die quantitative Aussagen über die berechneten Resultate ermöglicht, welche die Basis für eine ausführliche Diskussion und Interpretation der erzielten Stereoergebnisse bilden.

Contents

List of Abbreviations

2SF	Two-Stage Filter
AER	Address-Event Representation
ATIS	Asynchronous, Time-based Image Sensor
BP	Belief Propagation
BRIEF	Binary Robust Independent Elementary Features
CCD	Charge Coupled Device
CF	Corner Feature
CMOS	Complementary Metal-Oxide-Semiconductor
COG	Center of Gravity
DSI	Disparity Space Image
DSP	Digital Signal Processor
EISATS	Environment perception and driver assistance Image Sequence Analysis Test Site
ET	Event Transform
FPGA	Field Programmable Gate Array
FPS	Frames per Second
IDS	Imaging Development Systems
KITTI	Karlsruhe Institute of Technology and Toyota Technological Institute
LIDAR	Light Detection and Ranging
LSAD	Locally Scaled Sum of Absolute Differences
LSSD	Locally Scaled Sum of Squared Differences

NSAD	Normalized Sum of Absolute Differences
SAD	Sum of Absolute Differences
SSD	Sum of Squared Differences
SURF	Speeded Up Robust Features
SVD	Singular Value Decomposition
TC	Time Correlation
TFS	Time to First Spike
TOF	Time of Flight
WTA	Winner-Takes-All
ZSAD	Zero-mean Sum of Absolute Differences
ZSSD	Zero-mean Sum of Squared Differences

CHAPTER 1

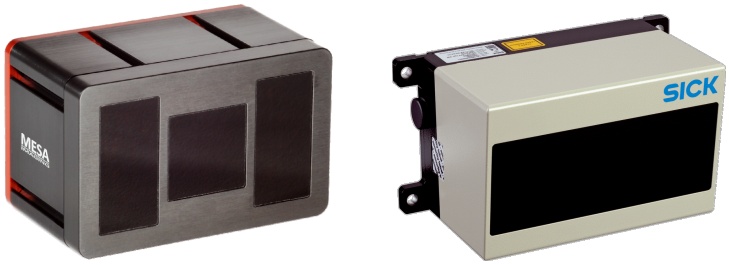
Introduction

1.1 Motivation

Automation is present in our daily lives. The cars we drive are assembled almost completely autonomously. Driver assistance systems improve safety in traffic. Our food and other articles of daily use come from large factories where production is carried out by robotic systems. For safe and accurate operation, these systems need to have knowledge about their environments. In many cases, a variety of different sensors are involved that analyze the surroundings of autonomously operating systems. Typically, such sensors also retrieve depth information for correct perception of the three dimensional world. The sensors for depth measurement can be grouped into active sensors, such as laser scanners (light detection and ranging (LIDAR)) or time-of-flight (TOF) cameras, and passive sensors, like stereo vision which consists of two cameras in a stereo configuration. Figure 1.1(a) shows as example a TOF camera from MESA Imaging[1] and Figure 1.1(b) depicts a LIDAR from SICK[2]. State-of-the-art laser scanners emit laser pulses and measure the round-trip time between the emission and the received pulse which bounced back from an object within the beam's directional path. Other laser scanners measure the phase difference between emitted and received laser beams. Time-of-flight sensors actively illuminate the scene with modulated infrared light and measure the time of travel or phase difference between emission of the light and the sensor's detection of the returned light beams. Both active sensors (TOF and laser scanner) deliver accurate depth information, but depend on the active emission of signals in order to measure depth.

Therefore, passive sensors such as stereo vision sensors are considered an alternative. In an ideal stereo vision setup, two cameras are mounted coplanar with parallel optical axes and the observed scene points are projected onto the cameras' image planes.

[1] http://www.mesa-imaging.ch
[2] http://www.sick.com)



(a) (b)

Figure 1.1: (a) Time-of-flight camera (model SR4500) from MESA Imaging AG and (b) 3D laser scanner (model LD-MRS400001) from SICK AG.

One goal in stereo vision is to find corresponding pixels within the image pair and to calculate the horizontal displacement, also known as the *disparity*, of the pixels using stereo matching algorithms. The distance from a scene point to the camera is inversely proportional to the corresponding pixel's disparity, which allows the calculation of the 3D point in camera coordinates, if the geometry of the calibrated cameras is known. For each pixel of the camera, a dense stereo matching algorithm tries to find the corresponding pixel in the other camera's image, which ultimately leads to a 3D point cloud of the observed scene. Figure 1.2 shows a typical workflow of a stereo vision system for retrieving 3D information. Even though many stereo vision systems are in use, the computational effort of calculating the 3D depth maps is still a challenging task depending on the resolution and frame rate chosen for the stereo vision system. Computing the stereo correspondences for each image pixel and frame pair of a stereo video sequence may lead to a considerable amount of redundant work, since in many applications, only information about the changing areas of a scene is required. This means the processing of all correspondences for each image pair is a loss of valuable resources. In such cases, it would be helpful to have a camera which only captures the changing parts of the scene, which are then subject to further analysis and 3D reconstruction.

Such an alternative camera sensor is the *Silicon Retina* sensor. A silicon retina sensor efficiently transmits the data from the observed scene in the form of sparse events according to the model of nature [67]. This sensor differs in its construction from the conventional complementary metal oxide semiconductor (CMOS) or charge coupled device (CCD) sensors with respect to the chip architecture and electronic circuits around the pixels. The silicon retina technology also brings about new challenges regarding the processing of sparse data within stereo matching algorithms. Within this thesis we analyze the aforementioned sparse data delivered by the silicon retina sensor, and explore the ways in which stereo matching can be applied to sparse input data. In this context, we also adapt the classic stereo vision workflow to be used for silicon retina data with the goal to benefit from the sensor's advantages and to overcome its drawbacks.

1.2 Contributions

For stereo vision using conventional (*conventional* refers to the usage of monochrome/color cameras) cameras, many different stereo matching approaches have been developed. Most of these algorithms aim to calculate an accurate and dense disparity map of the captured image pair. Important peculiarities of the silicon retina sensor include a sparse data representation as well as a reduction of the information stored within one pixel. This exemplifies the challenge of establishing stereo matching correspondences using silicon retina cameras. We summarize the contributions of this thesis as follows.

Since classical stereo matching algorithms use intensity information as matching criterion and silicon retina sensors only deliver whether or not a pixel (event) has changed, pre-processing is necessary to adapt, for example, a standard SAD (sum of absolute differences) algorithm for silicon retina data. This pre-processing could be the generation of a standard image by aggregating several events over time. Even if this is an interesting strategy, we developed a novel event-based approach using the time difference between the received pixels of the left and right camera as matching criterion. Another algorithmic contributions of this work are two novel approaches for silicon retina-based stereo matching. The first method uses a global optimization scheme specifically designed to deal with sparse data in order to minimize the matching costs. For this we use the *Belief Propagation* (BP) approach, which uses the disparity information

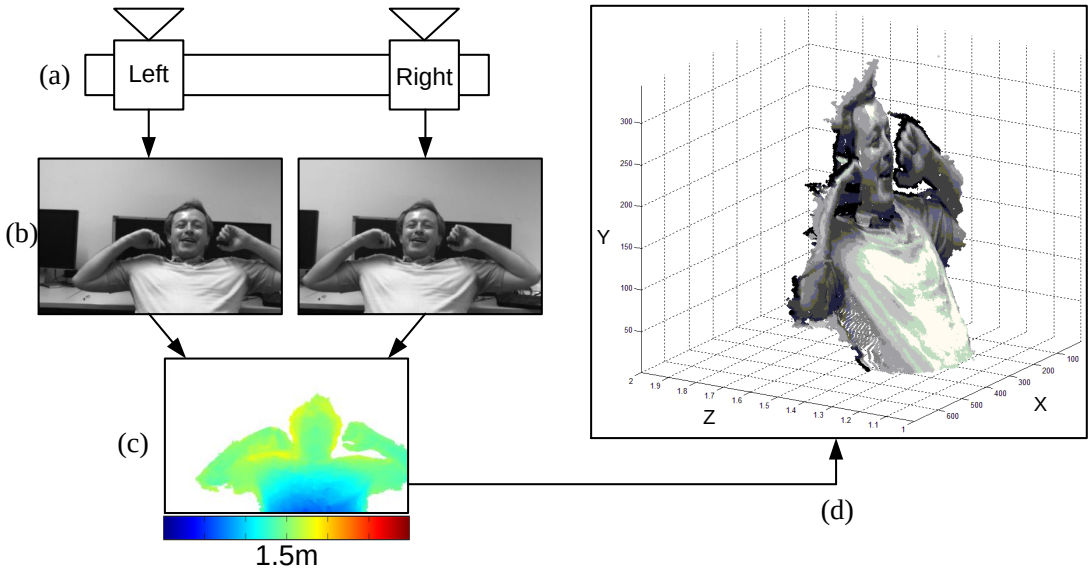


Figure 1.2: Overview of the workflow using a stereo vision camera system for 3D reconstruction. (a) Stereo camera system with both cameras. (b) Captured grayscale images from the left and right camera. (c) Calculated disparity image from the stereo matching algorithm. (d) 3D representation of the captured scene.

between neighbors to gain an overall matching result. The second approach is a filter which consists of two stages and analyzes the disparities around a considered pixel in order to improve its own disparity level.

For evaluation of the algorithms developed in this thesis, we first used planar objects at known distances. This pragmatic approach is easy to implement but is not accurate enough since not all pixels can be evaluated. This was the motivation for us to construct and utilize, for the first time, ground truth data sets which represent the real-world environment more closely, and are suitable for sparse silicon retina data. Using these ground truth data sets for the evaluation of the different algorithms implemented allows us to better interpret the algorithms' performance and provide a comparison with other algorithms.

Many classical stereo vision algorithms require calibrated cameras and a subsequently rectified image pair before the stereo matching is applied. Even though in principle, using the silicon retina camera in a stereo setup follows the same optical laws, no detailed calibration and rectification procedure for silicon retina data is available. Therefore, a calibration method suitable for silicon retina cameras was designed within this thesis, including a rectification procedure that was tailored to deal with sparse data. The calibration-based data preparation and the generated ground truth data set provided a valuable basis for our algorithmic development and evaluation.

1.3 Resulting Publications

The publications below have resulted from the work on this thesis. During my PhD thesis, I collaborated with Florian Eibensteiner, from University of Applied Sciences Hagenberg, whose PhD [24] topic dealt with the establishment of a stereo matching work flow on a *field programmable gate array* (FPGA) using data from silicon retina cameras. The calibration, rectification, and generation of ground truth data described in Sections 5.1 and 6.1.3 of this thesis are the result of this collaboration, which also led to several joint publications.

Journal

- Jürgen Kogler, Florian Eibensteiner, Martin Humenberger, Christoph Sulzbachner, Margrit Gelautz and Josef Scharinger: *Enhancement of sparse silicon retina-based stereo matching using belief propagation and two-stage post-filtering*. Journal of Electronic Imaging (SPIE), volume 23, issue 4, number 043011, pp 1-15, 2014.

Book Chapter

- Jürgen Kogler, Christoph Sulzbachner, Martin Humenberger, Florian Eibensteiner: *Address-event based stereo vision with bio-inspired silicon retina imagers*. Published in the Book *Advances in Theory and Applications of Stereo Vision*, Editor: Asim Bhatt and published from InTech, pp 165-188, 2011.

Conferences

- Jürgen Kogler, Florian Eibensteiner, Martin Humenberger, Margrit Gelautz, Josef Scharinger: *Ground truth evaluation for event-based silicon retina stereo data*. In the Proceedings of the 9th IEEE Embedded Vision Workshop (held in conjuction with IEEE CVPR), pp. 649-656, 2013.
- Jürgen Kogler, Martin Humenberger, Christoph Sulzbachner: *Event-based stereo matching approaches for frameless address event stereo data*. In the Proceedings of the 7th International Symposium on Visual Computing (ISVC), pp. 674-685, 2011.
- Jürgen Kogler, Christoph Sulzbachner, Florian Eibensteiner, Martin Humenberger: *Address event matching for a silicon retina based stereo vision system*. In the Proceedings of the 4th International Conference from Scientific Computing to Computational Engineering (IC-SCCE), pp 17-24, 2010.
- Jürgen Kogler, Christoph Sulzbachner, Erwin Schoitsch, Wilfried Kubinger, Martin Litzenberger: *ADOSE - New bio-inspired in-vehicle sensor technology for active safety*. In the Proceedings of the 14th International Forum on Advanced Microsystems for Automotive Applications (AMAA), pp 155-164, 2010.
- Jürgen Kogler, Christoph Sulzbachner, Wilfried Kubinger: *Bio-inspired stereo vision system with silicon retina imagers*. In the Proceedings of the 7th International Conference on Computer Vision Systems (ICVS), pp 174-183, 2009.

Magazine Article

- Jürgen Kogler, Christoph Sulzbachner, Erwin Schoitsch, Wilfried Kubinger: *ADOSE: New in-vehicle sensor technology for vehicle safety in road traffic*. Published in the European Research Consortium for Informatics and Mathematics (ERCIM) Magazine, number 78, pp. 47-48, 2009.

1.4 Structure of the Thesis

The remainder of this thesis is organized as follows. In Chapter 2 the concept of a camera and the derivation of the principle from the human eye is explained, which leads to the description of the silicon retina technology. Chapter 3 explains fundamentals of stereo vision relevant for this work and the challenges which must be addressed regarding stereo matching. Chapter 4 presents the related work regarding the topic of stereo matching using conventional cameras as well as silicon retina cameras. In Chapter 5, the stereo matching algorithms developed in this thesis are explained in detail, including relevant pre-processing steps, if needed, such as the conversion of events to images. Chapter 6 presents the evaluation method, followed by the results and their discussion. Chapter 7 finally concludes the work and provides an outlook on future work.

CHAPTER 2

Fundamentals of Vision Sensors

Before starting our discussion of stereo camera systems, we describe relevant fundamentals of the human visual system and historical camera development. After that, we review in Section 2.2 the functionality of modern digital cameras, in general, before we present in Section 2.3 the description of the silicon retina sensor technology. First, we talk about the historical development of the silicon retina technology, and second, we describe more specifically the differences between them and conventional cameras.

2.1 Human Eye and Camera History

Cameras, we use on a daily basis, seek to imitate nature where eyes serve as important organs for humans and animals to perceive their environment. The eye is considered the most important sensing organ of human beings, because ~80% of all sensory input is perceived by the eyes. The eyes work reliably under different environmental conditions and may adjust to a large dynamic range of 100dB [72].The size of the area in the brain which is contributing to visual processing indicates the importance of the visual sense. Approximately 60% of the cerebral cortex is involved in the task of visual processing, and in total, more than 30 different areas of the brain are active during the processing of visual input [31].

A closer look at the human eye is given in Figure 2.1. It illustrates the eye more specifically, including the path of the light through the cornea. In order to cope with changing light conditions, the pupil can control the amount of light passing the lens. The retina consists of different receptors converting the incoming light to signals which can be interpreted by the human brain. A human is equipped with ~5 million cones which are responsible for recognizing the three basic colors: red, green, and blue, and ~100 million rods for the detection of monochromatic light [5]. Not all of these more than 100 million signals are transferred to the brain directly. That means the eye applies a pre-processing to the receptor input and reduces the number of signals transferred

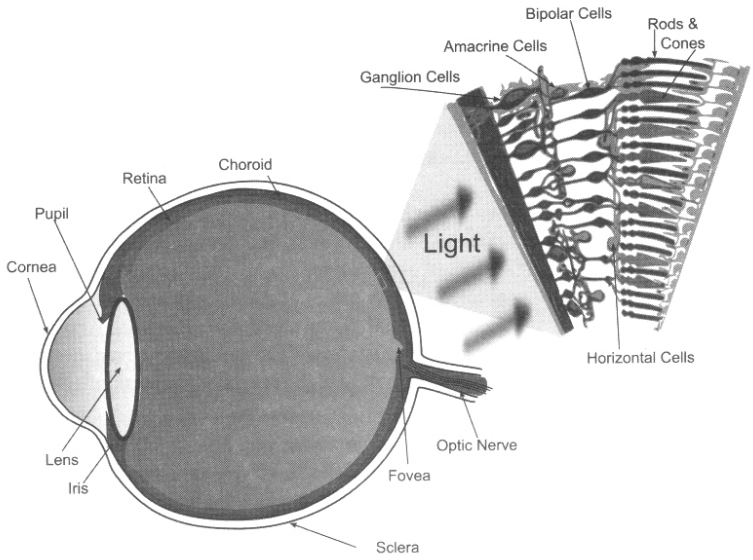


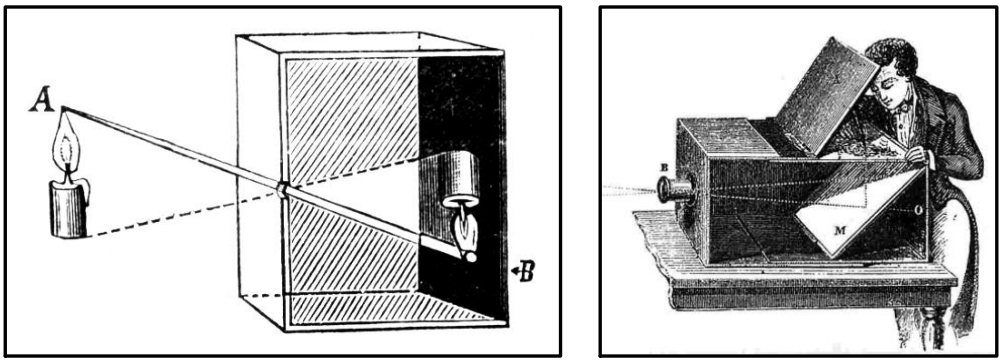
Figure 2.1: Left: A cross section of the eye with all of its components is shown. Right: A cross section of the retina is shown with the light path through the retina to the photoreceptors (rods and cones). Image source: *Next Generation Artificial Vision Systems* [5].

to the brain to ~1.5 million. This mechanism of natural vision processing has been improved over thousands of years of evolutionary development and forms the basics of cameras currently found on the market.

Before the camera was invented, the philosopher Aristotle (384–322 BC) discovered some optical principles used in today's cameras. Aristotle discovered that light passing through a small hole and entering a dark room projects an up side down picture of the scene outside the hole at the opposite end of the room [95]. Based on this discovery, the idea of the *camera obscura* was created, also known as *pinhole camera*. Figure 2.2(a)[1] shows the principal idea of the camera obscura. The light reflected from an object (candle) passes a hole in the box and produces a turned copy of the object on the backside of the box. Later, Leonardo Da Vinci (1452–1519) made investigations regarding the optical path of the camera obscura and discovered similarities with the human eye. This research brought up the idea of equipping the camera obscura with optical lenses instead of having a simple hole. In 1686, Johann Zahn (1641–1707) developed the first portable camera obscura, which is shown in Figure 2.2(b)[2]. This camera obscura was equipped with a mirror, located inside the box, that was turned at a 45° angle to the lens, which projected the captured image to a focusing screen at the top of the camera

[1]Wikimedia commons public domain: `https://commons.wikimedia.org/wiki/Category:Camera_obscura#/media/File:Camera_obscura_1.jpg` - Original: Fizyka z 1910

[2]Wikimedia commons public domain: `https://commons.wikimedia.org/wiki/Category:Camera_obscura#/media/File:Camera_Obscura_box18thCentury.jpg` - Original : By unknown illustrator (19th Century Dictionary Illustration)



(a) (b)

Figure 2.2: (a) Shows the principle behind the pinhole camera. (b) Usage of the camera obscura to capture the scene by drawing the projected image on paper.

where it could be drawn on paper. Since this time, the camera obscura was used by painters and photographers to capture the real world and record the observed scene. This was a starting point for today's camera which are digital and are used in different fields to capture the real world in images.

2.2 Conventional Digital Cameras

The digital cameras which are currently used on a daily basis contain either monochrome sensors which only capture intensity images, or sensors with the ability to capture colors. Additionally, the cameras can be differentiated based on their internal hardware architecture. The differences are described in more detail in the following subsections.

2.2.1 Intensity Capturing Method

Within intensity cameras, the light is not divided into different wavelengths (red, green and blue), as it is in cameras with color capturing abilities. Cameras which only capture intensity images are very often used in the industry, where in order for the processing of the image to occur, the color information is not needed, but rather, higher sensitivity of the sensor is necessary. The sensor has a higher sensitivity because more light contributes to each single pixel as result of the missing color separation.

2.2.2 Color Capturing Methods

For capturing color images, three different approaches can often be found in modern cameras.

The first method works with three sensors, where one sensor for each of the colors red, green and blue exists within the camera. Figure 2.3 shows the principle of the three chip sensors. Here, the light goes through a prism and is split into the spectral

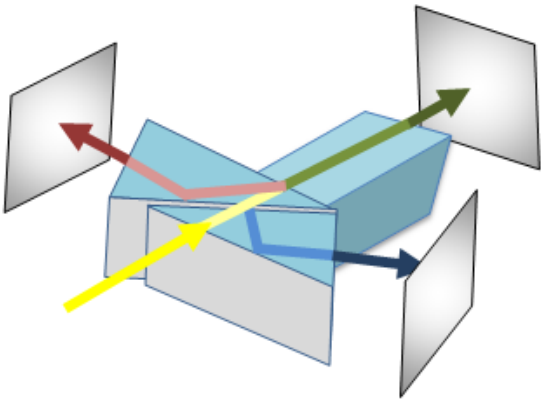


Figure 2.3: Color capturing principle with a three chip camera sensor, where a prism splits the light into its respective wavelengths.

colors red, green and blue. The camera is equipped with three sensor chips where each color is captured by a single chip and converted into digital values. The color information from the three chips is then combined again into the full color image. Since the mechanical placement of all components has to be very precise, this technology is normally expensive.

The second method to capture color images relies on a principle designed by Foveon[1] [48]. This approach from Foveon takes into account the fact that the penetration depth of light in silicon is different depending on the wavelength of the light. Figure 2.4 shows the schematic which demonstrates how the color is separated. The

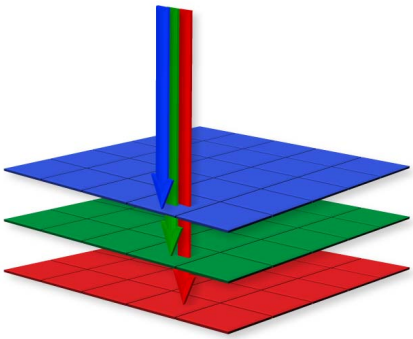


Figure 2.4: Color capturing principle which takes into account the fact that light with different wavelength (energy) is able to penetrate the silicon in different depths.

camera sensor chip has three layers, because the light shall be separated in the three colors red, green, and blue. Blue light has the shortest wavelength therefore has a

[1]Foveon, Inc. created and produced the Foveon X3 Sensor

high energy and penetrates the silicon in the deepest layer, followed by the color green and red. After the conversion of the red, green and blue light into digital values, the information can be combined to get the full color image.

The third method is based on filtering the light for each pixel using the Bayer color filter pattern [2] and one sensor chip. Figure 2.5 shows a Bayer pattern consisting of red, green and blue pixels. The spectral range of the color green lies in-between red and blue

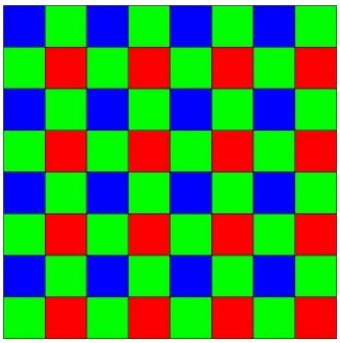


Figure 2.5: The Bayer pattern which filters the light based on the color filter each pixel has. Interpolation between the pixels can reconstruct the full color image.

and is the wavelength where the eye of humans has its highest sensitivity [72]. Therefore, the Bayer pattern is equipped with more pixels sensitive to green light, rather than red or blue. The missing color information of a pixel position can be estimated by evaluating the neighbor pixel values using different demosaicking (interpolation) techniques [71]. This color imaging method is simple in contrast to the methods mentioned above. An additional advantage is that a camera with Bayer pattern can have similar data rates to intensity cameras.

2.2.3 CCD Cameras

Digital cameras can often be assigned to two different groups, according to the image sensor used. The first type is called *Charged Coupled Device* (CCD) [55, 56]. In the CCD chip, the light is measured by a photo sensor which carries out the photon-to-charge conversion. A capacitor accumulates the charge, which is proportional to the quantity of light. Figure 2.6 shows the schematic of a Full-Frame CCD architecture with the main components. While the shutter is closed, the light is prevented from reaching the sensor. During that time, the pixel information (charge) is read out by shifting it vertically through the chip into a register where the charge is then converted into a voltage level. This analog signal is transmitted to the camera electronics, where it is converted into a digital signal that represents the digital image. This read out procedure makes the sensor slow and is responsible for *motion blur* if fast motion is captured. Another side effect caused by the architecture of the CCD sensor is *blooming*, where saturated pixels leak charges to neighboring pixels and generate spots with saturated pixels. The *smearing* effect is also a disadvantage of the CCD sensor, because

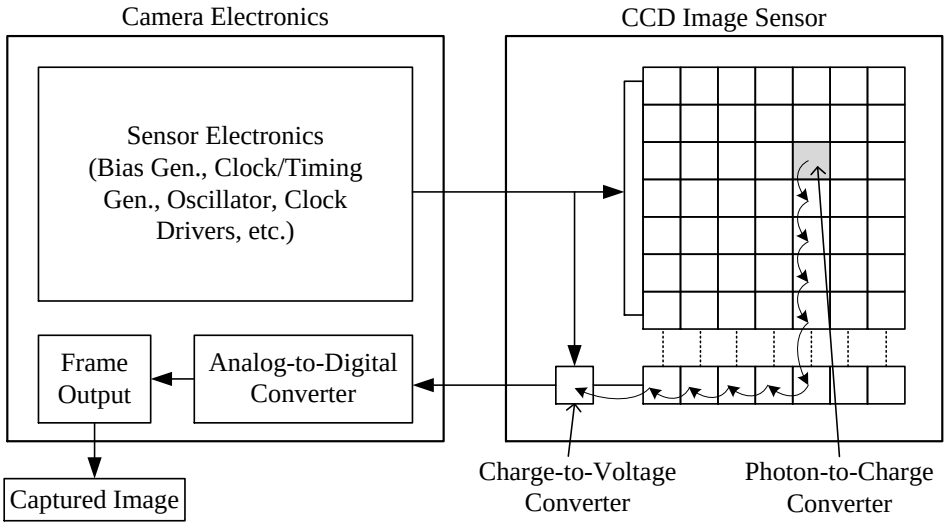


Figure 2.6: The schematic of a Full-Frame CCD-based camera sensor. The sensor electronics located in the camera electronics is steering the image capturing process. On the right, the CCD image sensor is depicted such that for each pixel, photon-to-charge conversion takes place. After the shutter is closed, the charge is vertically shifted into a register. The charge is shifted through the register to an output and converted to an analog signal (voltage). Afterwards, the analog signal is transmitted to the camera electronics where it is converted into a digital image.

it causes bright vertical lines. These smearing lines are generated by the transportation of saturated pixel charges collecting additional charges during the vertical transport. The main advantage of the CCD sensor in comparison to the CMOS sensor is the higher sensitivity and the capability to capture good images in low light conditions.

2.2.4 CMOS Cameras

The second type of image sensor is based on the *Complementary Metal Oxide Semiconductor* (CMOS) [55,56] technology. In the CMOS chip, the light is measured and converted into a charge by a photo sensor, which amplifies the measured charge with a transistor circuit in order to quantify the amount of light. After the shutter closes, the pixel charge can be read directly, which is the main difference between the CMOS and the CCD design. Figure 2.7 shows the schematic of a CMOS-based camera chip with the main components. Each pixel is equipped with an amplifier and the electronic circuit necessary to convert the charge directly into an analog signal. All pixels can be read parallel and then transferred to the analog-to-digital converter. The main advantage of CMOS chip architecture is the fast direct access to the pixels which enables the read out of only the certain regions of interest on the chip-level. The disadvantages are that it allows variations in pixel behavior because of fabrication tolerances, and its need for

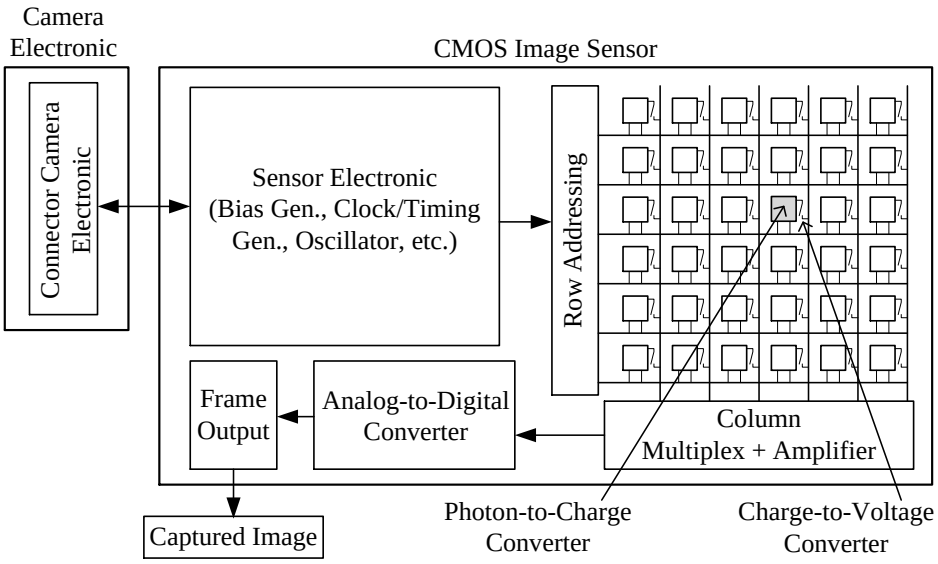


Figure 2.7: The schematic of a CMOS-based digital camera. The sensor electronic is located in the image sensor, where each pixel is amplified and read directly after the shutter closes. The charges are converted into analog signals, which are then transferred through the output logic to the analog-to-digital converter, where they are converted into digital images.

more light to retrieve better images.

Both the CMOS and CCD architectures have their benefits and drawbacks and can be found in current camera models.

2.3 Silicon Retina Camera

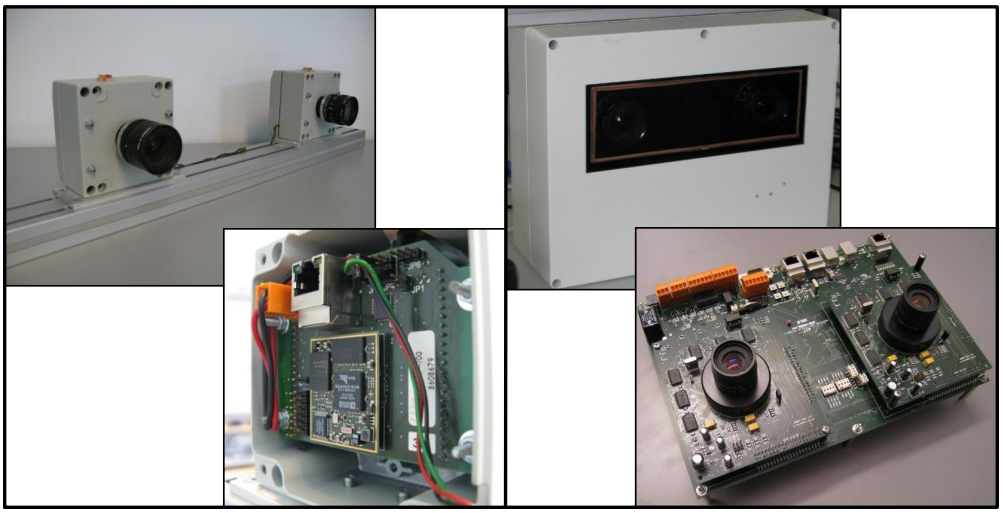
Cameras based on silicon retina sensors use a similar design to CMOS sensors, but have differences in architecture and functional behavior, which we shall explain in the following subsections.

2.3.1 Silicon Retina Development

The research on silicon retina camera sensors started about 30 years ago. In 1988, Mead and Mahowald [61] executed the first integration of a silicon retina sensor on a single chip. This model differed in its function from conventional cameras in that its goal was to imitate some basic steps of the human visual system. In a follow-up work one year later, Mahowald and Mead [59] introduced the term *Silicon Retina*. Since that time, different silicon retina architectures and related processing technologies have been developed. These approaches range from simple light to variable impulse rate transformation [21], time-to-first-spike encoding (TFS) [82, 83, 96], motion sensing

and computation systems [9], sensing spatial contrast by doing more on-chip signal processing [20,74] to a model for a mammalian retina [101,102].

In this thesis we use two different models of silicon retina sensors, which are different in their spatial and temporal resolution, as well as the dynamic range in which they operate. The first sensor has a resolution of 128×128 pixels and a temporal resolution up to 1ms. This sensor provides a dynamic range of up to 120dB, and is described in the work of Lichtsteiner *et al.* [53,54]. The second sensor, which is used throughout most parts of this thesis, represents the newer generation of silicon retina sensors, and is called ATIS (Asynchronous, Time-based Image Sensor). This sensor provides a spatial resolution of 304×240 pixels and a temporal resolution up to 10ns. Additionally, this sensor has an increased dynamic range of 143dB, which enables its operation in different applications with varying lighting conditions. A detailed description of the ATIS sensor can be found in the work of Posch *et al.* [68–70]. We use both of these sensor types in stereo configurations for the stereo matching and 3D reconstruction task. Figure 2.8 shows the stereo silicon retina sensors used in this work. The left images (a) show



(a) (b)

Figure 2.8: (a) Small silicon retina sensors with a resolution of 128×128 mounted on a rigid bar to build up a stereo head. (b) Silicon retina stereo system equipped with two 304×240 sensors, which are integrated directly into a common hardware with a fixed baseline.

the lower resolution silicon retina sensor. For our experiments we used two of them mounted on a rigid bar to form a stereo camera system. In Figure 2.8(b) the silicon retina stereo system with the higher resolution sensor is shown. Two of these sensors were integrated directly into a common hardware to form a stereo camera head with a fixed baseline.

2.3.2 Silicon Retina Sensor - Technical Overview

The silicon retina sensor differs from conventional camera chips, but the base is a CMOS chip with adaptations made to achieve the specific behavior of a silicon retina camera. Most conventional sensors are considered cameras with frame rates up to 200fps (frames per second), but a silicon retina sensor is a frame-free, asynchronous, time-continuous photoreceptor. Every pixel independently delivers data based only on changes of the luminance. Hence, this kind of vision sensor offers three potential advantages:

- The construction of the sensor chip and the event-based processing of the visual information leads to a very high temporal resolution.
- The high dynamic range of the sensor is achieved by a logarithmic measurement of the photo current, which makes the sensor suitable for use in situations where large fluctuations in amount of light occur.
- The asynchronous and illumination-change-dependent event data generation significantly reduces the data which must be transmitted, because only dynamic parts of the scene induce a data transfer.

Figure 2.9 illustrates these three advantages with the help of image examples. The

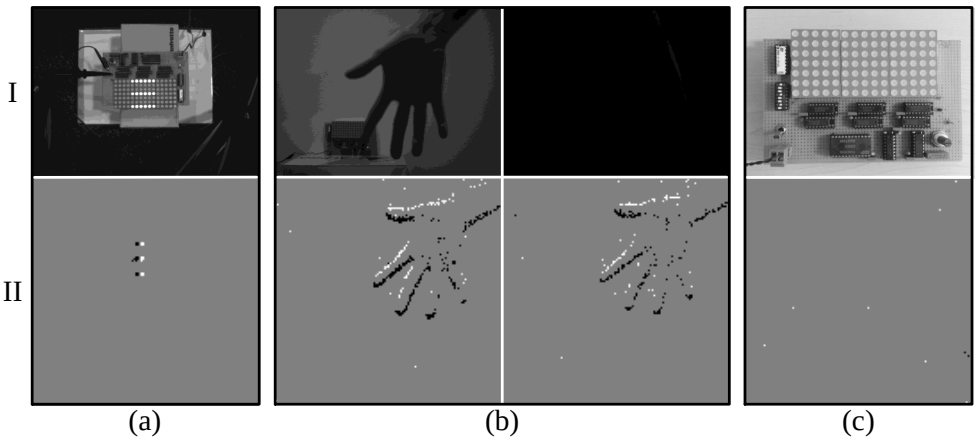


Figure 2.9: Differences between conventional image sensor (I) and a silicon retina sensor (II). (a) Benefit of the temporal resolution (conventional sensor has motion blur). (b) High dynamic range (conventional sensor needs adaption on shutter to operate). (c) Efficient data transmission (conventional sensor transfers the whole image without any changes).

first row (I) shows the images of a conventional camera sensor and the second row (II) displays the output of a silicon retina sensor. In Figure 2.9(a) the benefit of a high temporal resolution is demonstrated. Here, the disadvantage observed is that fast movements within a scene will, in contrast to silicon retina sensors, lead to motion

blur when using conventional cameras. A board equipped with lights running from right to left is placed in the scene, but at any given point in time, only three vertically oriented lights are active. As the lights change with increasing speed, a point will occur where the shutter time of the conventional camera is not able to capture only three lights, and motion blur occurs. In this case the silicon retina sensor with the high temporal resolution is able to capture fast movements in the scene in detail, and only the three active lights are shown within the silicon retina image. Figure 2.9(b) shows the advantage of the high dynamic range in the example of a moving hand that was observed under office light (left) and nearly dark surroundings (right). Under the office light, both sensors see the moving hand. But if the light is reduced to nearly complete darkness, the conventional sensor cannot capture an image of the hand any more, unless a shutter or aperture change is made. In contrast, the silicon retina sensor is able to capture the moving hand without changing its configuration. In Figure 2.9(c) the difference in data transmission is shown. The scene shows the same board with the lights in (a), but in this case, it is switched off, which means no activity of light changes (static scene) occur over time. For each new frame, the conventional sensor captures all pixels of the sensor in order to generate a new image. This results in redundant data transmission. A silicon retina sensor transfers data only if changes of pixels, also known as *events*, were detected. This means that data is transferred only when intensity changes occur. In this case, the silicon retina sensor is observing the board with switched off lights, and that means no data is transferred for the static scene, except a few events which are considered to be noise.

As mentioned, the data delivered by the silicon retina sensor are called events, and are generated inside the retina sensor by the illumination change detector which functions as follows. Formally, an event can be defined as $e(x, y, t)$ [73], where (x, y) is the spatial location of the pixel firing the event, and t is the time of occurrence given in the unit of timestamps. One timestamp corresponds to the temporal resolution of the silicon retina sensor (1 timestamp $\triangleq$ 10ns in the case of ATIS). Depending on the polarity of the change of illumination I over a period of time Δt, an event can either be positive +1 (on-event) or negative −1 (off-event):

$$e(x, y, t) = \begin{cases} +1 & \text{if} \quad I(x, y, t) - I(x, y, t - \Delta t) > \Delta I \\ -1 & \text{if} \quad I(x, y, t) - I(x, y, t - \Delta t) < -\Delta I \end{cases}, \tag{2.1}$$

with the adjustable on- and off-threshold ΔI. Each pixel of the sensor measures the changes of illumination in a logarithmic manner, and works asynchronously and time continuously. In contrast to conventional frame-based image sensors, which generate frames of intensity or color values representing the observed area, these kinds of event-based neuromorphic visual sensors only deliver events on intensity changes caused by the dynamic parts of a scene.

The event data is transferred from the silicon retina sensor to the subsequent processing system by using a protocol which is called address-event representation (AER) . This kind of data representation is well-suited for nervous systems and frameless data

transmission [58, 84]. Boahen [10] proposed a first application using AER for point-to-point communication between neuromorphic chips. Since the silicon retina sensor does not need to transfer the information of the whole sensor matrix, the AER sends the individual address of the firing pixel as a packet within the protocol. In addition to the address represented by the x- and y-coordinate, the time of occurrence and the polarity (on- or off-event) are transferred within such a packet. In Figure 2.10, the construction of the AER protocol used for the stereo silicon retina system from Figure 2.8(b) is shown in more detail. The packets sent from the stereo camera can be divided into times-

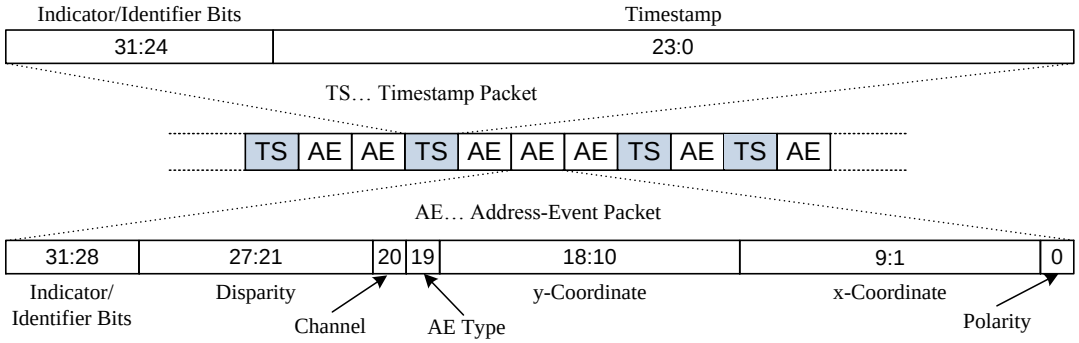


Figure 2.10: The AER protocol of the silicon retina stereo camera (see Figure 2.8(b)). Timestamp packets and address-event packets, both with a length of 32 Bit, are shown with the information inside them. The numbers within the packets describe the number of bits used for the different information.

tamp packets and address-event packets. After transmission of a timestamp packet, all address-events belonging to this timestamp are transferred, and afterwards the next timestamp is sent. In contrast to other AER protocols, the channel information is encoded in the address-event packet, which gives us the opportunity to identify whether an event was received from the left or right camera. Further information that can be encoded within the address-event packet is the disparity, but those bits are only used if the stereo matching takes place directly within the stereo camera head. In our case, we calculate the stereo matching results separately and therefore the disparity bits are not used.

2.4 Summary

This chapter described the principle of how the human eye works and the development of the first camera, also known as *camera obscura*. After reviewing the principles of modern digital cameras based on CCD or CMOS technology, we explained the design of a silicon retina sensor, which is inspired by human visual processing. The silicon retina camera, with its benefits of a high temporal resolution, a high dynamic range, and an efficient data transmission, is used for the development of suitable algorithms and evaluation experiments within this work.

CHAPTER 3

Fundamentals of Stereo Vision

The field of stereo vision addresses the derivation of depth information from a scene by observing the scene from two different points of view. Similar to the two human eyes, two cameras are placed next to each other in a stereo vision system to retrieve 3D information, as shown in Figure 3.1. The distance between the two cameras is called

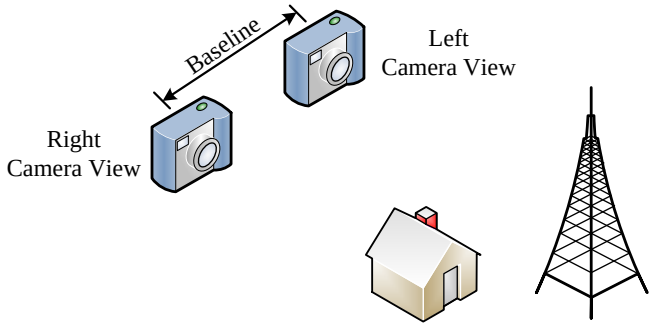


Figure 3.1: Stereo view with two cameras observing the same scene.

baseline and is an important parameter of a stereo camera set-up because it directly relates to the achievable depth accuracy. Our explanation of a stereo vision scenario is based on the epipolar geometry, which is introduced in Section 3.1. After that, the task of correspondence search, which is one of the key topics within stereo vision, is explained in Section 3.2. In Section 3.3 we review the calibration and rectification steps necessary for an accurate and precise calculation of stereo vision results. Finally, in Section 3.4 the calculation of the depth information is presented.

3.1 Epipolar Geometry

In our experiments we use a pinhole camera model [1], which is also used to explain the epipolar geometry [36]. Figure 3.2 shows the projection of a scene point S onto the left I_L and the right I_R image plane. The projected scene point S is represented by the

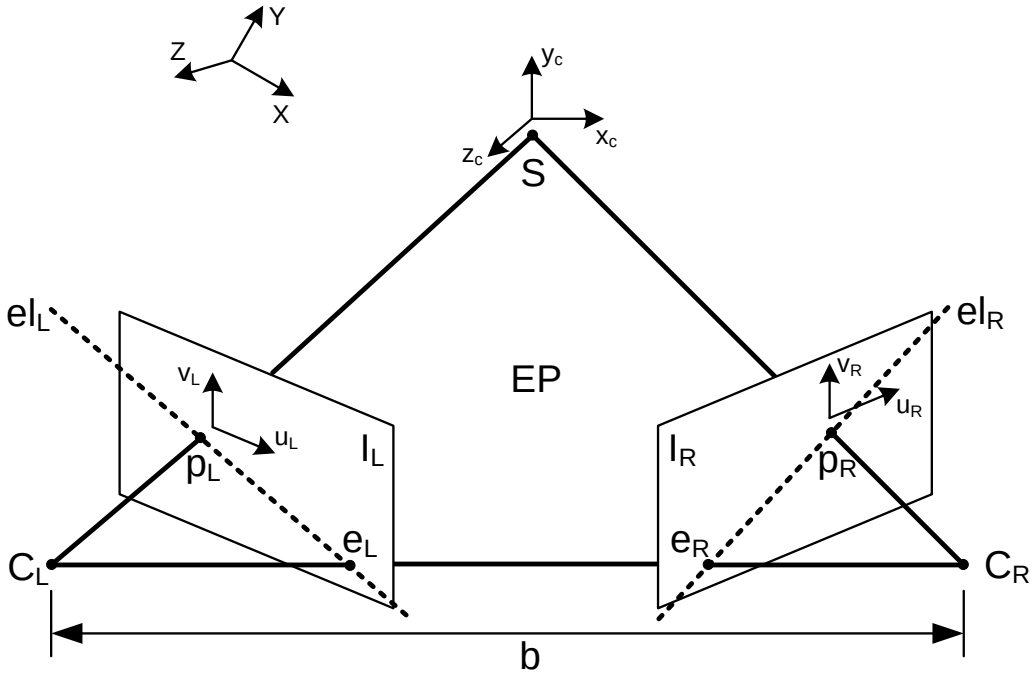


Figure 3.2: Epipolar geometry showing the projection of the scene point S onto the left image plane I_L and right image plane I_R.

image points p_L and p_R on the image planes. The distance between the optical centers C_L and C_R of the left and right camera is defined by the baseline b. Both image points $p_L = (u_L, v_L)^T$ and $p_R = (u_R, v_R)^T$ are given in pixels, and the scene point $S = (x_c, y_c, z_c)^T$ is given in meters, because the scene point is described in the camera coordinate system and the image points in the image coordinate systems. The world coordinate system is described by $(X, Y, Z)^T$ in meters. The epipolar geometric relation between points in the left and right image is described by the fundamental matrix F according to

$$p_R^T \, F \, p_L = 0. \tag{3.1}$$

The epipolar plane EP, spanned by the scene point S and the optical centers C_L and C_R, intersects the left and right image planes at the epipolar lines el_L and el_R. The epipoles e_L and e_R represent the points where the baseline b is intersecting the image planes. As a consequence, the corresponding point of point p_L is located on the epipolar line el_R in the right image and vice versa. Using the fundamental matrix F, the epipolar line el_L

can be calculated by

$$el_L = F^T \, p_R, \tag{3.2}$$

and the epipolar line el_R is accordingly calculated by

$$el_R = F \, p_L. \tag{3.3}$$

These relationships reduce the search space of corresponding pixels from the whole image to the epipolar lines. Once the epipolar geometry is known, a rectification can take place which further restricts the search space to a horizontal line by transforming the image planes in such a way that the epipolar lines are parallel. In Figure 3.3 the rectified epipolar geometry is shown, where the epipolar lines are parallel and the epipoles are located at infinity. Details about the rectification step are presented in Section 3.3.

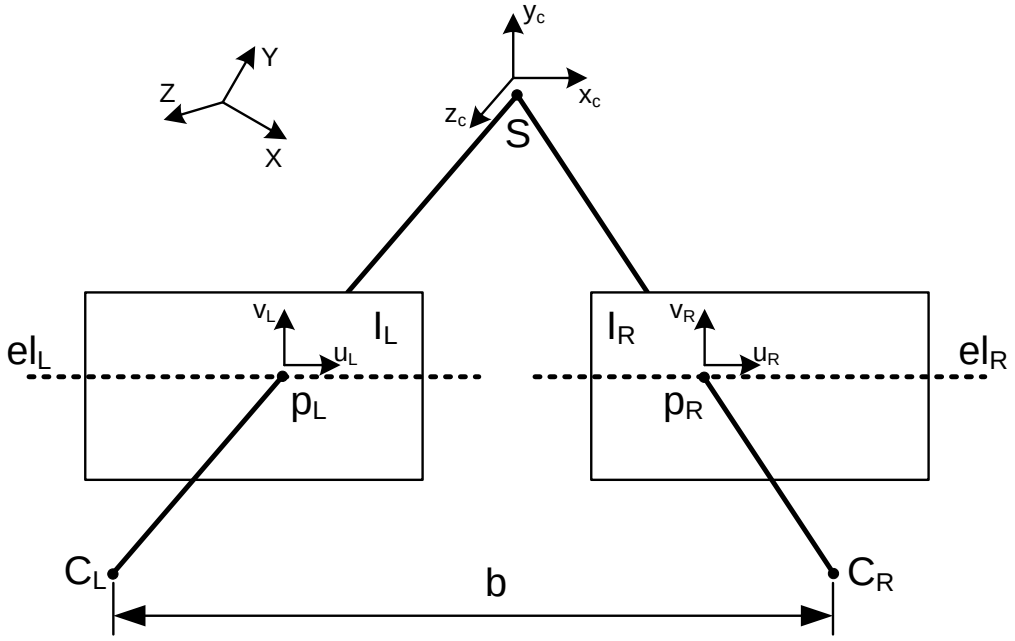
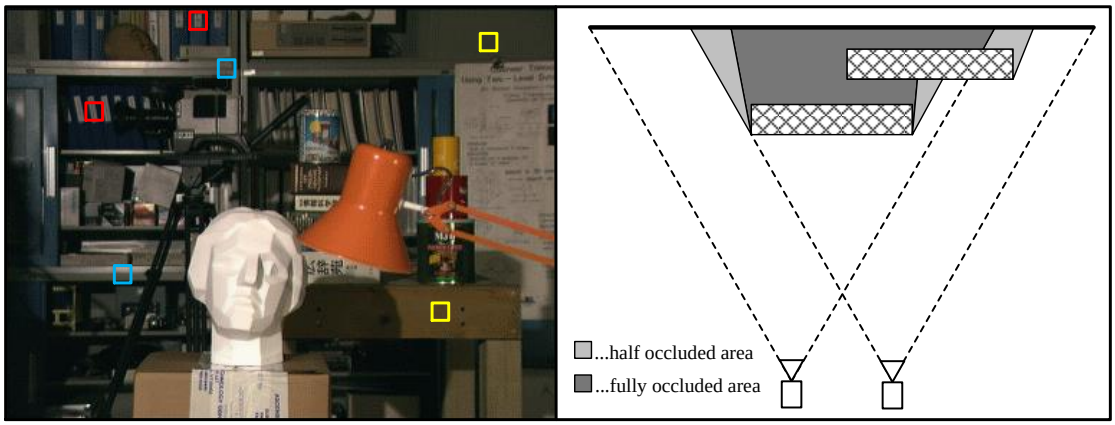


Figure 3.3: The rectified epipolar geometry with parallel epipolar lines.

3.2 Correspondence Problem

As shown before, the epipolar geometry can be exploited to restrict the search of corresponding pixels between the left and right image to the epipolar lines, which are, in case of an epipolarly rectified geometry, parallel horizontal lines. Even if the search is restricted to a horizontal line, the determination of the correct match is difficult. The reason is that pixels in the left image can have more the one matching candidate in the

right image, because of pixels with identical gray/color values. Another factor are occluded areas where based on the stereo geometry areas visible from the left camera are not seen from the right. Figure 3.4 summarizes major challenges of the correspondence search. Figure 3.4(a) shows the left image of the Tsukuba test set from the Middlebury[1]



(a) (b)

Figure 3.4: (a) Shows the left image of the Tsukuba test set from the Middlebury test database where yellow squares mark texture-less region, blue squares mark horizontal oriented texture and red squares mark repetitive pattern. (b) Shows the bird view of a scene with multiple objects observed from two cameras. Fully occluded areas are marked in dark gray and half-occluded areas are marked in light gray.

data set to demonstrate some challenges of the correspondence search, which are:

Texture-less regions are regions with similar color or intensity information which increase the difficulty to identify exact corresponding matches, because most of the pixels look the same. In Figure 3.4(a) (yellow marks) the background with the monochrome wall and the table in the foreground are representatives of texture-less regions.

Horizontally oriented textures are regions where less texture is available along the horizontal axis, which is at the same time also the search axis for the corresponding matches. Figure 3.4(a) (blue marks) shows an example of this challenge. The pixel rows of the shelf have nearly the same color. This leads to the texture-less region problem when correspondences are searched horizontally.

Repetitive patterns have grayscale differences, but this grayscale information is continually repeated as shown in Figure 3.4(a) (red marks). The pattern confuses the search because within a certain disparity range many matches can be found but only one match is correct.

[1] `http://vision.middlebury.edu/stereo`

Thin and small objects are problematic if instead of single pixels, blocks of pixel are used for matching. Block-matching can increase the quality of the matching process at low-texture areas, but fails if the background information within the block is much more dominant than the foreground object.

Environmental factors are the influence which comes from the cameras used. Here the camera and its settings can reduce the matching quality. Cameras, e.g., have a limited dynamic range to capture scenes with bright and dark areas. Other influences can be moving objects which cause motion blur captured differently from the left and right camera.

Figure 3.4(b) shows occluded areas which are another challenge the correspondence search has to deal with. Occlusions are areas which are not seen from the cameras. It can be distinguished between fully occluded areas (marked in dark gray) which are not seen from any camera and half-occluded areas (marked in light gray) seen only from one camera. The correspondence search is not able to find matches for these areas. If the occlusions are detected correctly from the stereo matching algorithm, then the disparity map shows holes where no depth information can be retrieved.

Additional challenges for stereo matching are transparent or non-Lambertian surfaces (e.g. with strong reflections). In this case it is difficult to find the corresponding match, because the matching algorithm cannot distinguish if a pixel is viewed through a transparent object or if it is observed as a reflection from a mirror. Those are special cases and do not represent fundamental matching problems we focus in this thesis.

3.3 Calibration and Rectification

As mentioned before a rectified image pair can reduce the search space for corresponding pixels, but this requires a calibration before the stereo matching takes place. The goal of the calibration is to determine the intrinsic and extrinsic parameters [1]. The calibration procedure is a well-covered research topic, thus, a comprehensive number of related work exists.

In the calibration method of Tsai [92], objects with a known geometry are captured which contain two or three orthogonal planes. The knowledge of the orthogonal planes is used to retrieve the calibration parameters of the camera. This approach requires a high configuration effort what makes the method not to the preferred calibration procedure.

Another approach uses one plane with a known calibration pattern, instead of a 3D object. The pattern is then captured from several distances and angles. On the one hand, this method is flexible, but on the other hand, it has the disadvantage that it is computationally expensive with an accuracy of the results which is depending on many factors, such as plane positions captured, number of images, feature points extracted, etc.. Because of its wide applicability, this calibration is introduced in many variations. Important work is presented by Hartley [37], Triggs [91], and Zhang [103], with the latter one also being used as calibration approach in this work.

There are many more approaches for camera calibration that address specific applications, such as vanishing point analysis presented by Caprile and Torre [17], or calibration using camera rotation, as proposed in the work of Stein [85].

Based on the work of Azad *et al.* [1] the extrinsic parameters describe the relation between the world coordinate system $(X, Y, Z)^T$ and the camera coordinate system $(x_c, y_c, z_c)^T$ using a rotation matrix $R^{3\times3}$ and translation vector $t^{3\times1}$ with

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + t. \tag{3.4}$$

The intrinsic parameters describe the relation between the camera coordinate system $(x_c, y_c, z_c)^T$ and the image coordinate system $(u, v)^T$, using

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{z_c} \begin{pmatrix} f_x \cdot x_c \\ f_y \cdot y_c \end{pmatrix} + \begin{pmatrix} c_x \\ c_y \end{pmatrix}, \tag{3.5}$$

where (f_x, f_y) describes the focal length independently for x and y-direction in case pixels are not square and (c_x, c_y) denotes the coordinates of the optical center within the image plane. These parameters can be summarized as

$$K = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{3.6}$$

where K represents the camera matrix. Additionally, there is a factor s inside the camera matrix, which represents the skew factor of the pixels. This factor describes how trapezoidal the pixels are, where the value is zero for perfectly shaped rectangular pixels. Using the intrinsic parameters expressed by K and the extrinsic parameters R and t in the projection matrix P

$$P = (K|0) \left(\begin{array}{c|c} R & t \\ \hline 0 & 1 \end{array} \right), \tag{3.7}$$

a 3D point given in homogenous coordinates $(X, Y, Z, 1)^T$ is transformed into homogeneous image coordinates $(u, v, 1)^T$ by

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \tag{3.8}$$

Further to these parameters, the parameters for the lens distortion must be calculated, because only an ideal lens would not have any distortion. There are two different types of distortions present in a lens, which are depicted in Figure 3.5. The first distortion is the radial distortion, which increases as the distance from the optical center increases.

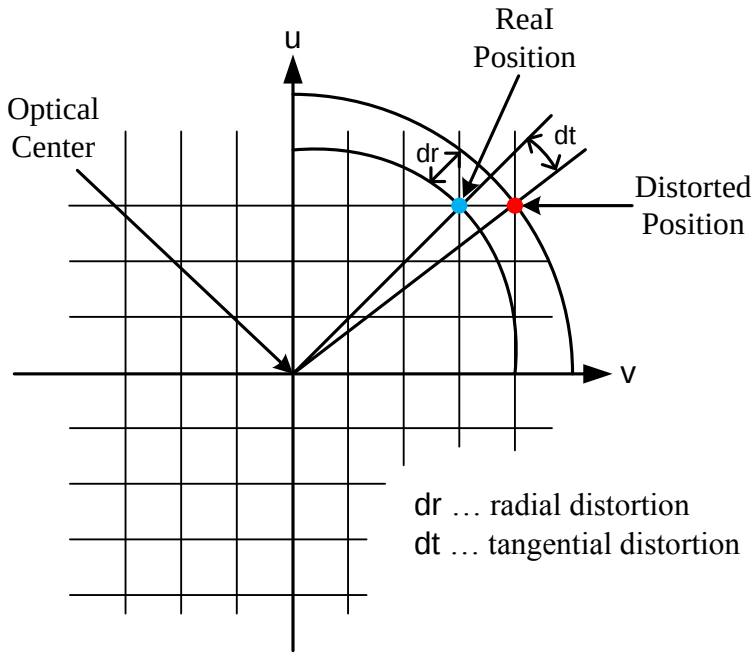


Figure 3.5: Explanation of the radial (d_r) and tangential (d_t) distortion caused by the lens.

A radial distortion can appear either as barrel distortion, or pin cushion distortion. The second distortion is the so-called tangential distortion, describing the displacement in tangential direction. To remove the distortion, the distortion coefficients are calculated describing the relation between the distorted camera coordinates $(x_{cd}, y_{cd}, z_{cd})^T$ and undistorted camera coordinates $(x_c, y_c, z_c)^T$, where $z_{cd}/z_c = 1$. For the approximation of the distortion, three coefficients k_{r1}, k_{r2}, k_{r3} are used for the radial model and two coefficients k_{t1}, k_{t2} for the tangential model. There is no closed form to calculate the undistorted coordinates and therefore, a so-called backward mapping takes place, where from the undistorted coordinates the distorted coordinates are calculated using

$$\begin{pmatrix} x_{cd} \\ y_{cd} \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \end{pmatrix} (1 + k_{r1}r^2 + k_{r2}r^4 + k_{r3}r^6) + \begin{pmatrix} k_{t1}(2x_c y_c) + k_{t2}(r^2 + 2x_c^2) \\ k_{t1}(r^2 + 2y_c^2) + k_{t2}(2x_c y_c) \end{pmatrix}, \tag{3.9}$$

where

$$r = \sqrt{x_c^2 + y_c^2} \tag{3.10}$$

is the radius between the optical center and the undistorted pixel. The calculated distorted coordinates $(x_{cd}, y_{cd})^T$ will be subpixel coordinates and a bilinear interpolation is applied to calculate the new pixel value for the undistorted coordinates.

In case of a stereo camera system, the stereo calibration is needed, which additionally estimates the relative translation and rotation between the left and the right camera, in

order to rectify the images as mentioned in Section 3.2 to limit the search space during the stereo matching procedure. There are two common approaches for calculating the rectification information. One is the method of Hartley [38], for which no calibration information is needed, and the other method is the calibrated case where the already known intrinsic and extrinsic parameters are used for the calculation of the rectification. The second method is used in the calibration toolbox of Bouguet [11] which is used in this work. In the work of Tsai [92] and Zhang [103] the calibrated stereo rectification approach is explained in detail. The method calculates two rotation matrices R_L and R_R, which describe the transformation of each image where the epipolar lines are parallel. After the rectification the camera matrices K_L and K_R are changed to rectified camera matrices K'_L and K'_R, where the parameters are adapted based on the rotation and shift during the rectification. As already for the removal of the lens distortion, the rectification step can be applied as a backward mapping and can be combined with the lens undistortion mapping. In this case a single mapping step is generated in which the subpixel values of the input image are taken and mapped to the final pixel positions which are integer coordinates.

3.4 Depth Reconstruction

Assuming that two corresponding points are found, the depth reconstruction of the scene point can take place. In Figure 3.6 the schematic of Figure 3.3 from a top view is shown, where the triangulation [34] of the scene point is illustrated. In this perspective

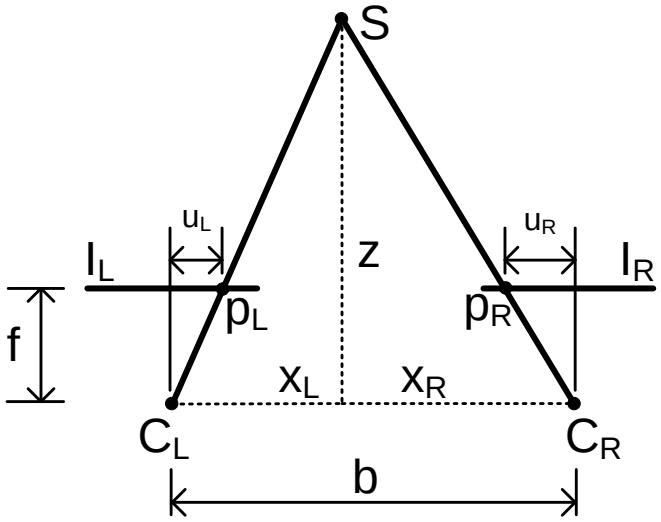


Figure 3.6: The reconstruction of a scene point using triangulation from two different views is illustrated. The difference $|u_L - u_R|$ represents the disparity d.

additionally the focal length f of the cameras and the distance z between the scene

point S and the cameras' optical centers C_L and C_R are marked. Considering x_L and x_R as helping variables splitting the baseline and u_L and u_R which describe the distance between the projected image points p_L and p_R and the center of the images planes I_L and I_R, the equation

$$\frac{u_L}{f} = \frac{x_L}{z} \quad \text{and} \quad \frac{-u_R}{f} = \frac{x_R}{z} \tag{3.11}$$

can be formulated using similar triangles and the intercept theorem. Substituting x_R with $b - x_L$ and writing x_L in an explicit form leads to

$$x_L = \frac{z \cdot u_L}{f} \quad \text{and} \quad x_L = \frac{z \cdot u_R}{f} + b. \tag{3.12}$$

Combining both equations in Equation (3.12) and writing z in explicit form gives

$$z = \frac{b \cdot f}{|u_L - u_R|} = \frac{b \cdot f}{d}, \tag{3.13}$$

where the difference $|u_L - u_R|$ represents the disparity d which is used to calculate the depth z of the scene point S.

$$z = \frac{f \cdot b}{d}, \tag{3.14}$$

which represents the law of calculating depth values z using the focal length f, baseline b and disparity d. Using this equation, all disparity values d calculated during the matching procedure are reconstructed into their depth values z representing the distance of the scene point S in relation to the optical centers.

3.5 Summary

This chapter summarizes the relevant fundamentals of stereo vision for this work. For a classical stereo vision camera system, two parallel-mounted cameras survey the same scene. Using a stereo matching algorithm that searches for pixel correspondence between the two views enables the reconstruction of depth information for each pixel. The horizontal displacement of the matched pixels is inversely proportional to the distance between scene point and camera and is called disparity. The relationship between disparity and the distance of the scene point to the camera is described by the epipolar geometry. If the epipolar geometry is known, the stereo image pair can be rectified, which reduces the search effort of the matching algorithm to one dimension. The cameras' calibration information which includes the specific parameters of the camera and lenses is necessary for calculating the epipolar geometry.

In the next chapter, we will present related work that considers stereo matching approaches in general and, in particular, stereo matching algorithms using data delivered by silicon retina cameras.

CHAPTER 4

Related Work

The correspondence problem, explained in Section 3.2, describes the fundamental task that must be solved by a stereo matching algorithm. Many diverse approaches have been developed over the last few years regarding the search for matches between the left and right camera images. In Section 4.1, an overview of stereo matching approaches that consider conventional stereo vision is given. Section 4.2 presents silicon retina-based stereo matching algorithms. In Section 4.3, techniques and methods for the evaluation of stereo matching results are presented.

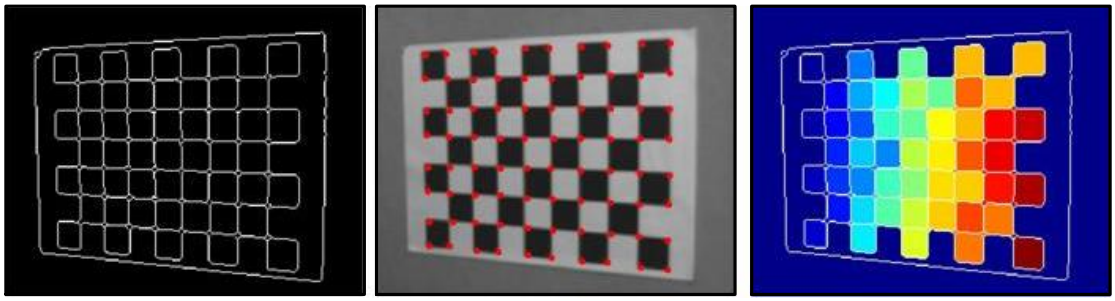
4.1 Conventional Stereo Matching

Stereo matching approaches can be divided into a variety of different categories, but a major distinction is traditionally made between area-based and feature-based algorithms. For both area- and feature-based algorithms, a calibration and rectification step is applied to the input data, as explained in Section 3.3. For further explanations, we assume that a calibrated and rectified image pair is available. Comprehensive summaries of area-based and feature-based stereo matching algorithms are presented in the work of Scharstein and Szeliski [75] and Brown *et al.* [14].

4.1.1 Difference Between Area-based and Feature-based Algorithms

Area-based algorithms attempt to match each pixel of an image pair by minimizing a cost function (probability of a correct match). This results in dense disparity maps even if uncertain pixels - such as occlusions - are present.

In contrast, feature-based algorithms do not attempt to find a match for each individual pixel. These algorithms first detect appropriate features, such as lines, corners or segments, as depicted in Figure 4.1. For the extracted features, matching costs are determined, which are then used to search for correspondences. This also means that the disparity map is sparse, thus, only representative values for the matched features



(a) (b) (c)

Figure 4.1: Examples of different image features which can be used for stereo matching. (a) Line features. (b) Corner/Point features. (c) Segment features.

are present in the disparity map. Shi and Tomasi [80] have analyzed different features and give an overview of which features are useful for tracking (matching).

4.1.2 Cost Calculation and Cost Aggregation

For both area-based and feature-based stereo matching, costs must be calculated. As previously mentioned, the calculated costs represent a measure for the probability of a correct match. For stereo matching algorithms, the calculated costs for each pixel and disparity level are usually stored in a so-called *Disparity Space Image* (DSI), or cost volume, shown in Figure 4.2. In the context of area-based algorithms, we are

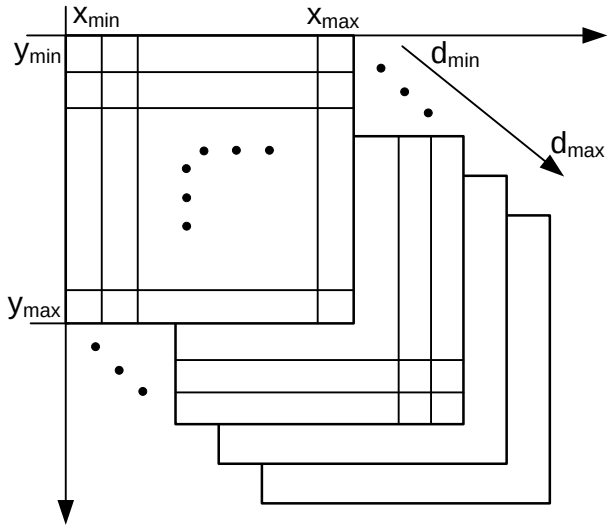


Figure 4.2: Disparity space image which stores for each pixel (area-based) or each feature (feature-based) the calculated costs within the defined disparity range.

usually dealing with DSIs that are filled densely. Contrary feature-based algorithms will produce sparsely populated DSIs corresponding to the sparsely computed costs of features.

The following constraints are often utilized to reduce mismatches and remove impossible matches from the DSI.

Uniqueness The assumption can be made that a pixel from the left image has only one unique corresponding match in the right image if the point is not occluded in one of the images. As described in Section 3.2, besides occlusions, other challenges in the correspondence search can lead to multiple matches for the same reference pixel. Due to the uniqueness assumption only one match is valid, which means that all other matching candidates must be excluded.

Ordering The ordering constraint demands that the pixel order of the left scan line remain unchanged in the corresponding scan line of the right image. This constraint is violated if the perspective view of the left and right camera is too different. It also fails on thin objects in the foreground.

Smoothness The smoothness constraint assumes that the disparity around a pixel only changes smoothly. The constraint is not valid at locations where depth discontinuities, such as objects edges of foreground objects, occur.

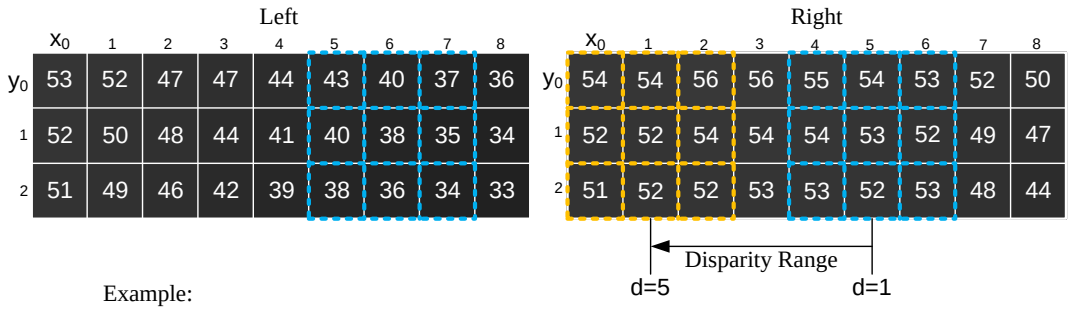
Consistency This constraint describes the fact that a corresponding match searched from the left to the right image must be also found if the search is done from the right to the left image. For the consistency check, the two disparity maps are analyzed, and a certain threshold can be defined which excludes matches that do have a higher disparity difference (e.g., a disparity difference of more than one pixel) between the left-right and right-left matching results.

The usage of single pixels to find corresponding matches will lead to many mismatches, and therefore the costs are often aggregated with the purpose to reduce the ambiguity of possible matches. The basic idea of so-called *cost aggregation techniques* is to process the cost entries of the DSI using windows of suitable shape and size based on the assumption that pixels within a certain neighborhood are likely to have the same disparity.

The following metric represents a common cost calculation metric used for stereo matching, where pixels of the left image $I_l(x, y)$ (reference image) are correlated with the pixels of the right image $I_r(x, y)$. The Sum of Absolute Differences (SAD) metric calculates for a pixel (x, y) and disparity d the cost values $C_{SAD}(x, y, d)$ with

$$C_{SAD}(x, y, d) = \sum_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} |I_l(x + i, y + j) - I_r(x - d + i, y + j)|, \tag{4.1}$$

where the cost aggregation is over an $m \times n$ window. In Figure 4.3 the principle of the matching using a 3×3 aggregation window is shown. The best match is represented by



Example:

$C_{SAD}(x_6,y_1,d_1)$ = |43-55|+|40-54|+|37-53|+|40-54|+|38-54|+|35-52|+|38-53|+|36-52|+|34-53| = 139

$C_{SAD}(x_6,y_1,d_2)$ = |43-56|+|40-55|+|37-54|+|40-54|+|38-54|+|35-53|+|38-53|+|36-53|+|34-52| = 143

$C_{SAD}(x_6,y_1,d_3)$ = |43-56|+|40-56|+|37-55|+|40-54|+|38-54|+|35-54|+|38-52|+|36-53|+|34-53| = 146

$C_{SAD}(x_6,y_1,d_4)$ = |43-54|+|40-56|+|37-56|+|40-52|+|38-54|+|35-54|+|38-52|+|36-52|+|34-53| = 142

$C_{SAD}(x_6,y_1,d_5)$ = |43-54|+|40-54|+|37-56|+|40-52|+|38-52|+|35-54|+|38-51|+|36-52|+|34-52| = 136

Figure 4.3: Grayvalues from a part of the left and right image, which are used for the area-based matching. As an example, the matching costs for a 3×3 window and a disparity range of 5 using the SAD correlation method are calculated.

the lowest costs. In this example, the lowest costs of the pixel at location (x_6, y_1) with the value 38 from the left image are found at disparity 5 (i.e., at location (x_1, y_1)) in the right image. Other related correlation metrics are used and explained in Section 5.2.2.2.

It is also possible to first transform the input images in order to enhance their properties for stereo matching. The Census and Rank transforms [100], for example, encode the intensity differences between a pixel p_c (center) and a pixel p_n (neighbor) with 0 and 1 according to

$$\xi(p_c, p_n) = \begin{cases} 1 & \text{if} \quad p_c > p_n \\ 0 & \text{if} \quad p_c \leq p_n \end{cases}. \tag{4.2}$$

The Census transform uses Equation (4.2) to generate a bit vector for each pixel which encodes its difference to all neighbors within an $m \times n$ window as shown in Equation (4.3).

$$I_C(x,y) = \bigotimes_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \bigotimes_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} \xi(I(x,y), I(x+i, y+j)) \tag{4.3}$$

Here, the Census image I_C is calculated, where $\bigotimes$ symbolizes the bit-wise concatenation of the $m \times n$ neighborhood. Image I_C has stored a bit vector at each position, which is then used for the further cost calculation. For calculating the costs C_{Census} given by

$$C_{Census}(x,y,d) = \sum_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} H_D(I_{C\_l}(x+i, y+j), I_{C\_r}(x-d+i, y+j)), \tag{4.4}$$

the Hamming distance [35] H_D between the bit vector of the left Census image I_{C_l} and the bit vector of the right Census image I_{C_r} at the disparity d is calculated and summed up over an aggregation window of size m × n. The Hamming distance H_D between two bit vectors (v_1, v_2) is calculated by

$$H_D(v_1, v_2) = \sum_{i=0}^{(m \times n)-1} v_1[i] \neq v_2[i], \tag{4.5}$$

where the number of all different elements within the two vectors is calculated.

These methods tend to be more robust against illumination differences between the left and right camera, variations in shutter times, and other factors that may influence the absolute intensity values. A detailed evaluation and comparison of the non-parametric local transforms is presented in the work of Hirschmüller and Scharstein [45].

The choice of a proper window size is an elementary decision for the aggregation, because the size influences the matching results. Choosing a larger window size increases, on one hand, the probability of finding uniquely identified correct matches. However, on the other hand, large windows fail at depth discontinuities and decrease the chances of properly detecting the disparity of small objects. Using smaller window sizes usually increases the matching quality at depth borders, but the detection of disparities in regions with repetitive and textureless patterns becomes less reliable. Therefore, assigning an appropriate aggregation window size is an essential task in stereo matching. In the work of Fusiello *et al.* [30], as well as in the work of Hirschmüller [41] and Hirschmüller *et al.* [44], multiple window aggregation techniques are presented as ways to overcome the problem of choosing a correct window size. These approaches use different window sizes and varying shapes of windows around the considered pixel to calculate the costs. Another approach to determine the window size follows the work of Kanade *et al.* [50], where the window size and shape are adaptively set after a local analysis of intensity and disparity variations. This means that an initial calculation of the disparities is used to form a statistical model, which helps, in conjunction with the intensity analysis, to adaptively choose the best matching window. Yoon *et al.* [99] present a process of analyzing the object boundaries with an initial disparity map, which is further used in choosing a proper window size and shape. The procedure of finding the best support window size and shape and applying the window to the cost computation is computationally expensive and increases with larger window sizes. Therefore, Veksler's study [94] demonstrated the use of integral images, which reduce the processing effort and are not dependent on the support window size. In order to better model object surfaces, which are mainly slanted and not fronto-parallel to the camera, an aggregation using slanted support windows can be applied per the work of Bleyer *et al.* [8] and Cho and Humenberger [19]. In addition to the size, shape, and orientation of the aggregation window, a weight can be determined for each pixel in the window to describe the impact of its contribution to the aggregation. Yoon and Kweon [97, 98] offer an adaptive weight approach, where the support-weight of each pixel within the window is calculated based on the color similarity and the distance to the center pixel. In this context, suitable techniques for cost aggregation, including

newer cost volume filtering techniques, have gained traction [22, 46]. A good comparison of different aggregation methods and techniques can be found in the work of Tombari *et al.* [90]. Hosni *et al.* [47] present an evaluation study in their work that considers different approaches to calculating adaptive support weights.

After the cost aggregation, the best matches are usually selected by a winner-takes-all (WTA) approach, where the lowest costs represent the disparity finally chosen. All of these aforementioned methods are local approaches because the final cost function is evaluated locally (within a small neighborhood). In contrast to local techniques, global methods minimize a global energy function (using multiple scanlines, or even the whole images). An example of such a global energy function E is

$$E(DM) = E_{Data}(DM) + E_{Smooth}(DM), \tag{4.6}$$

where E_{Data} describes the data term which measures the color dissimilarity for each pixel, E_{Smooth} represents the smoothness term measuring the disparity difference of neighbor pixels and DM which represents the disparity map.

There are different global optimization approaches that try to either optimize the energy of each scan line, as in Dynamic Programming [3, 6, 33], or optimize the energy of the whole image, as used in Graph Cuts [12, 52] and Belief Propagation (BP) [29, 89] approaches. Another approach that searches for an optimal global solution is the Simple Tree Method introduced in the work of Bleyer and Gelautz [7]. In the interest of completeness, we should also mention the work of Hirschmüller [42, 43], which describes the Semi-Global stereo matching approach that combines concepts of local and global stereo matching.

4.2 Silicon Retina-based Stereo Matching

For stereo matching using silicon retina cameras several matching algorithms have been developed, which differ from the stereo matching algorithms used for conventional stereo vision systems. We have divided the stereo matching approaches for silicon retina cameras into two different groups. One group proceeds by using event data from the silicon retina sensors directly without converting the event data into images. The other group follows the direction of using the event data of silicon retina sensors by converting the incoming events first into images (see Section 5.2.1) and then applying one of the stereo matching algorithms used for conventional stereo vision explained in the previous section. The first group is categorized under the term *event-based* algorithms, and the second group is called *event image-based* algorithms. Both groups have their benefits and drawbacks. On the one hand, taking an event-based approach enables exploiting the unique sensor characteristics, and on the other hand, aggregating single events to images allows directly using many well-known stereo matching algorithms.

Stereo matching for silicon retina sensors started in 1989 when Mahowald and Delbrück [57] presented an event-based stereo matching approach which operates on one-dimensional event input data, using static and dynamic image features. This approach is based on the cooperative algorithm presented in the work of Marr and

Poggio [60], where matching candidates within the same disparity receive a support and matching candidates violating the uniqueness constraint are inhibited. Motivated by the idea of a cooperative stereo matching algorithm, Hess [40] presented in his work two approaches. The first algorithm measures the time difference between events and calculates the costs for the matching with an inverse linear function. Additionally it is assumed that a single frontal object is observed, where all events within the same period have the same disparity. This limitation of considering a global disparity is handled in the second algorithm by processing each event individually. Here, the disparity of each event is calculated by assuming that disparity changes spatially and temporally smooth. Additionally the mean disparity of previous events restrict the disparity range, which leads to more representative stereo matching results. In the work of Schraml *et al.* [78] event image-based stereo matching algorithms are presented, where as mentioned events are converted into images to apply conventional stereo matching approaches. The work evaluates different correlation metrics and comes to the conclusion that the normalized sum of absolute difference (NSAD) metric works best.

This point represented the state-of-the-art in stereo matching using events from retina cameras at the beginning of this thesis. Part of our motivation was to develop a silicon retina stereo set-up which delivers a calibrated and rectified input for the algorithms. Since the approaches by Hess [40] had not been evaluated yet in a qualitative way, one of our goals was to implement an event-based time correlation algorithm and to test it with real-world data. We also wanted to evaluate the comparison of Schraml *et al.* [78] with calibrated data and extend the comparison with a feature-based stereo matching approach. Additionally we saw potential using global optimization methods from the conventional stereo matching area and decided to apply them to stereo matching algorithms using events. Our approaches are presented in detail in Section 5.2. After this starting point in parallel some new work based on silicon retina-based stereo matching was published, which we present in the following.

Shimonomura *et al.* [81] proposed in 2008 a neural network which emulates the stereo matching process in the visual cortex (V1), using a disparity energy model. This approach exploits the biological characteristics of the silicon retina sensor and tries to emulate depth perception characteristics of the visual cortex. In the event-based stereo matching approach of Benosman *et al.* [4] a temporal-spatial activation of pixels forms coactivation sets, which are used for the determination of the epipolar geometry. Without an explicit calibration and rectification step, the method can be applied on any stereo camera set-up independent on its geometrical structure. Dominguez-Morales *et al.* [23] present in their work an event-based fuzzy stereo matching technique which counts the events received at each pixel. All counts are written in a table and the matching takes place in the table. The counts within this table represent matching costs used for the matching process. The fuzzy approach is that entries with different counts are considered as matching candidates. In the work of Rogister *et al.* [73] an event-based stereo matching algorithm is proposed which uses the spatial distance to the epipolar line and appearance in time of events as matching criteria. Additional constraints such as po-

larity of events, ordering, uniqueness and similar temporal activity of pixels are taken into account for the correspondence search. The work of Carneiro *et al.* [18] proposes an event-based stereo matching approach similar to the work of Benosman *et al.* [4] and Rogister *et al.* [73], which uses for the matching spatio-temporal information. The main difference is that this approach allows to use more than two cameras for the stereo matching and subsequent 3D reconstruction. This algorithm benefits from the N-ocular usage to reduce occluded areas and ambiguities of the scene. Piatkowska *et al.* [65, 66] proposed in their work an event-based adaptive cooperative stereo matching approach based on the work of Marr and Poggio [60] and Hess [40]. The extension is to apply the algorithm on two-dimensional areas such as event data from two silicon retina cameras. Here, the spatial neighborhood within the same disparity level contributes excitatory and the neighbors across all disparity levels contribute inhibitory to find the correct matches of a considered pixel. In the work of Eibensteiner *et al.* [26] an event-based algorithm is presented using time differences as correlation metric. Additionally a segmentation of the sensor field takes place which lets the matching process focus only on areas with a certain event activity to reduce the computational effort.

Some of the algorithms mentioned have also been tested on embedded platforms to evaluate their real-time performance. In order to complete the related work section, we outline the work realizing silicon retina-based stereo matching on hardware platforms such as a field programmable gate array (FPGA) or a digital signal processor (DSP). The approach of Schraml *et al.* [78] is implemented on a DSP, and an FPGA version of the algorithm is presented in a study by Eibensteiner *et al.* [25]. Shimonomura *et al.* [81] and Eibensteiner *et al.* [26] implemented their approach on a FPGA platform. A time correlation algorithm [51] is implemented on a DSP by Sulzbachner *et al.* [88] as well as on the FPGA processing unit by Eibensteiner *et al.* [28].

4.3 Evaluation of Stereo Matching Algorithms

Testing is an important part of developing stereo matching algorithms. The evaluation of the stereo matching results provides an overview about how accurately and robustly a stereo matching approach performs. Different platforms for the evaluation of stereo matching algorithms producing dense disparity maps are available. The most popular evaluation platform for dense stereo matching algorithms is the Middlebury[1] stereo database developed by Scharstein and Szeliski [75]. This online evaluation platform provides many stereo image data sets consisting of the stereo image pair and the corresponding ground truth data. The data sets represent static scenes and have been created with a structured light approach [76]. For the evaluation of the algorithm results, the disparity maps must be generated and uploaded to the website. The evaluation engine calculates the performance of the matching algorithm, within a certain disparity error threshold, by pixel-wise comparison with the reference disparity values. Many stereo algorithm developers, contributing to approximately 160 entries to date, have used

[1] `http://vision.middlebury.edu/stereo`

this platform for evaluation. This offers a significant indication of how a developed algorithm performs in comparison to other algorithms. The platform is up-to-date and constantly growing.

However, a certain disadvantage of the Middlebury platform is that some data sets do not realistically represent real-world scenarios and the processing speed as well is not considered within the Middlebury ranking. The usage of stereo vision as 3D sensor technology has been growing over the last couple of years, especially in driver assistance systems, for autonomous robotics as well as consumer vehicles. To provide a more suitable evaluation platform for this kind of application in particular, the KITTI[1] (Karlsruhe Institute of Technology and Toyota Technological Institute) benchmark was introduced by Geiger *et al.* [32].

Similar to Middlebury, this platform provides data sets to evaluate stereo vision algorithms (as well as optical flow, tracking, visual odometry, and object detection) online. Unlike Middlebury, these data sets have been recorded from the roof of a car driving on regular roads, with a front pointing stereo camera. The reference 3D data has been determined with a laser scanner calibrated onto the stereo camera. Another remarkable difference to the Middlebury database is that, at KITTI, the processing time of the algorithm is also a part of the evaluation. This platform is rather new in the stereo vision community, thus, fewer algorithms are available than at Middlebury.

The Auckland analysis test site (EISATS[2] - Environment perception and driver assistance Image Sequence Analysis Test Site) provides several data sets of, for example, dangerous situations in traffic scenes. These data sets also include challenging scenes for the camera hardware, such as direct sunlight, shadows, and fluctuating light. Unfortunately, no reference data is available, which makes no direct statistical evaluation and comparison of different algorithms. To overcome this limitation, EISATS also provides synthetic sequences of automotive scenes with ground truth [93].

A further stereo vision evaluation method was presented by Meister *et al.* [62]. The provided data sets[3] show a huge variety of different weather conditions, motion, and depth layers. City as well as countryside situations were acquired at night and at day.

4.4 Summary

This chapter presented the relevant related work on stereo matching especially on silicon retina-based stereo matching. Both groups of silicon retina-based stereo processing introduced, event-based and event image-based, aim to use the uniqueness of silicon retina sensors to retrieve depth information. Using this specific type of data is the obvious first step for silicon retina stereo matching. However, advances in global optimization of the cost volume and disparity post-processing for conventional stereo matching showed the importance of these algorithmic steps, which are currently not

[1] `http://www.cvlibs.net/datasets/kitti/index.php`
[2] `http://www.mi.auckland.ac.nz/EISATS`
[3] `http://hci.iwr.uni-heidelberg.de/Benchmarks`

addressed in the related work regarding silicon retina stereo. Thus, we tackle this challenge and details are given in the following sections.

All the presented evaluation methods and platforms within this chapter well contributed to making progress in dense stereo vision for scientific as well as industrial purposes. However, none of them can be used for evaluating silicon retina stereo systems because of two important reasons. Firstly, many of the available data sets are static, so no events can be created and, thus, no silicon retina sensor output is available. Secondly, and this is the main reason why it does not work for us, all data sets acquired with conventional video cameras do not represent the asynchronous, time-continuous, event-driven spiking output of the silicon retina sensor chip properly. Another fact is that challenging lighting conditions and fast moving objects can be well handled by a silicon retina sensor and are a less significant limitation than for conventional video cameras. This is the reason why we developed evaluation methods especially for silicon retina-based stereo matching, presented in Section 6.1.

CHAPTER 5

Silicon Retina-based Stereo Matching

In this section, we describe our new stereo matching approaches for silicon retina cameras. Before the stereo matching algorithms are described in Section 5.2, Section 5.1 introduces calibration and rectification procedures which were specifically designed for silicon retina-based stereo camera systems. Section 5.2.1 describes the special properties of silicon retina-based data and their preparation for usage with the stereo matching algorithms. We start in Section 5.2.2 with the event image-based algorithms, which include area-based and feature-based techniques. We take a closer look at event-based stereo matching algorithms in Section 5.2.3. In Section 5.3 we propose two new techniques for improving silicon retina-based stereo matching results. The first method (Section 5.3.1) uses a global optimization scheme specifically adapted to deal with sparse data in order to minimize the matching costs. The second method (Section 5.3.2.2) seeks to efficiently eliminate outliers in a post-processing step.

5.1 Calibration and Rectification

Before the stereo matching approaches are explained, the necessary preparatory tasks for stereo matching, such as calibration and rectification, are described as they differ to stereo vision systems based on conventional grayscale or color cameras. For conventional cameras, the calibration task is well-known as summarized in Section 3.3. Before the calibration can begin, the lenses of the cameras must be adjusted. For the silicon retina camera, we developed a hardware, which enables a straightforward adjustment of the camera lenses. After the adjustment, the calibration takes place. The standard checkerboard calibration pattern (shown in Figure 5.1 on the left side) cannot be used for silicon retina cameras in the same way as for conventional cameras. Observing the same static calibration pattern with the silicon retina camera will lead to an image as

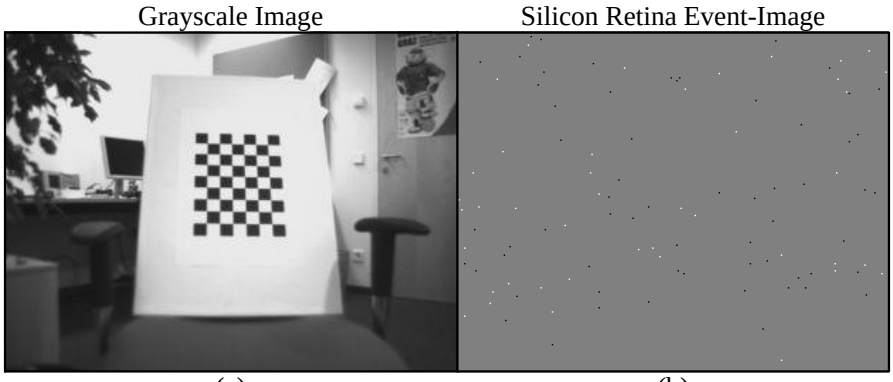


Figure 5.1: (a) Shows the checkerboard calibration pattern captured and used for the calibration procedure of a conventional monochrome sensor and (b) shows the silicon retina event-image generated by observing the same calibration pattern. No information except that of noise is present in the image captured of the static pattern.

shown on the right side of Figure 5.1, because it contains no changes in brightness. Hence, other procedures or modified techniques must be established for the calibration of silicon retina cameras. Since the calibration and rectification of silicon retina stereo sensors is a relatively new topic, for which practically no literature is available, we have developed two novel techniques expressly tailored to deal with the characteristics of silicon retina cameras.

To add motion to the scene, we use the static pattern alongside a white paper moved up and down in front of the checkerboard pattern in our first approach, as shown in Figure 5.2(1). This movement generates events at the locations of the black squares of the checkerboard pattern. The events are converted into a binary event-image in stage (2), as explained in more detail in Section 5.2.1. In stage (3), morphological operators are applied to the binary event-image to extract separate areas (*blobs*). All blobs with certain properties, such as a pixel count between 10 and 75 pixels and an aspect ratio bigger than 0.5, are considered to be valid and are used in the search for blob center features, as shown in Figure 5.2(3). The ego motion of the person moving the white sheet generates additional events that are visible in the binary event-image and influence the blob detection. Therefore, the relevant blob center features are annotated and selected manually by the user, who assigns the correct order of specific centers used in the calibration step. The list with the blob center feature points is used as input for the calibration toolbox of Bouguet [11], as indicated in step (5). The results showed that the usage of a calibration step improves the results of the stereo matching, but this method requires a great deal of manual interaction and has a low overall accuracy regarding the blob center extraction.

For this reason, we developed a second approach to improve the quality of the silicon retina-based calibration. In our second approach, not only the checkerboard pattern

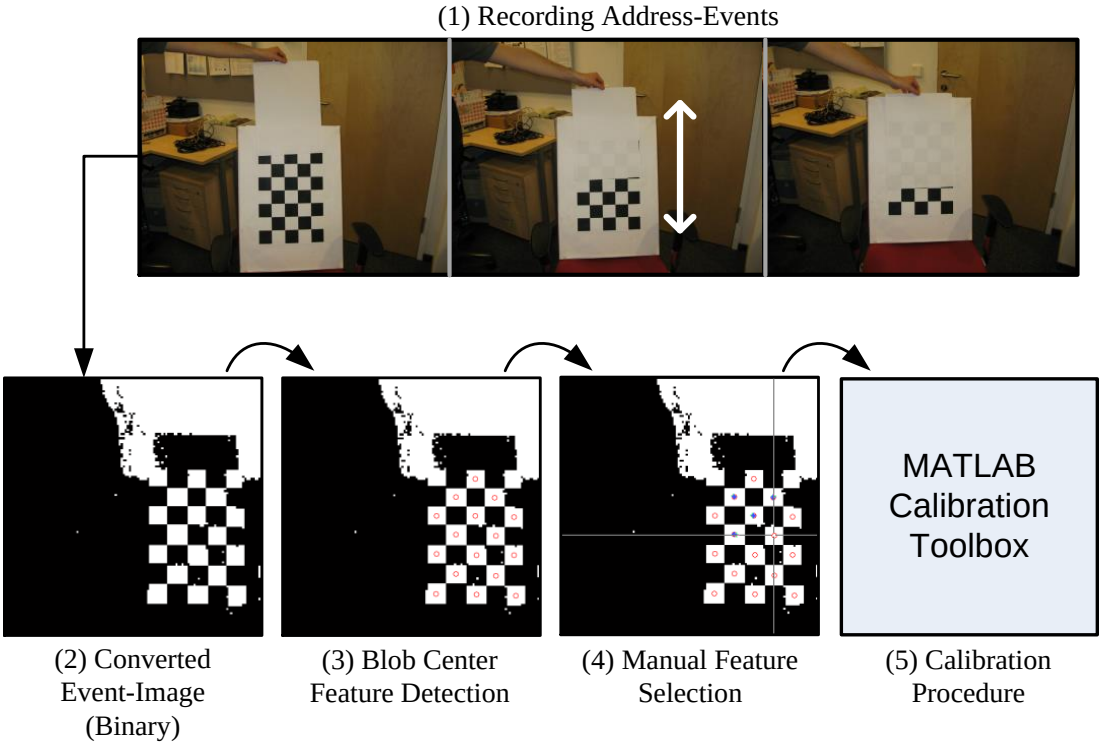


Figure 5.2: (1) shows the generation of events by moving a white paper up and down in front of a static checkerboard calibration pattern. (2) shows the converted binary event-image generated from the received events. (3) presents the results of the blob center feature detection. (4) illustrates the manual selection of detected feature points (centers) involved in the calibration process. (5) represents the calibration toolbox which takes the extracted points as input and delivers as output the calibration and rectification parameters.

was used, but also a square pattern and a circle pattern, as illustrated in Figure 5.3(1)I-III. Additionally, the generation of events was, instead of moving a white paper up and down, done by a computer screen, which flashes with the chosen calibration pattern. This flashing screen generates events that are collected over time and converted into a binary event image, as shown in Figure 5.3(2). Automatic filtering delivers a cleaner binary event image, shown in step (3), which is then subjected to morphological operations in step (4). The main difference to the first approach, aside from the flashing calibration pattern, is the extraction of the blob center features. The subsequent blob center feature detection in step (5) works fully autonomously and is also robust to changes in viewing perspectives by using the three circles with holes in the middle (see Figure 5.3(1)) as reference. The approach is described in detail in the work of Schörghuber [77] and evaluated in our work published by Eibensteiner *et al.* [27]. Using this approach accelerates the calibration procedure of silicon retina cameras. Mueggler *et al.* [64] presented as

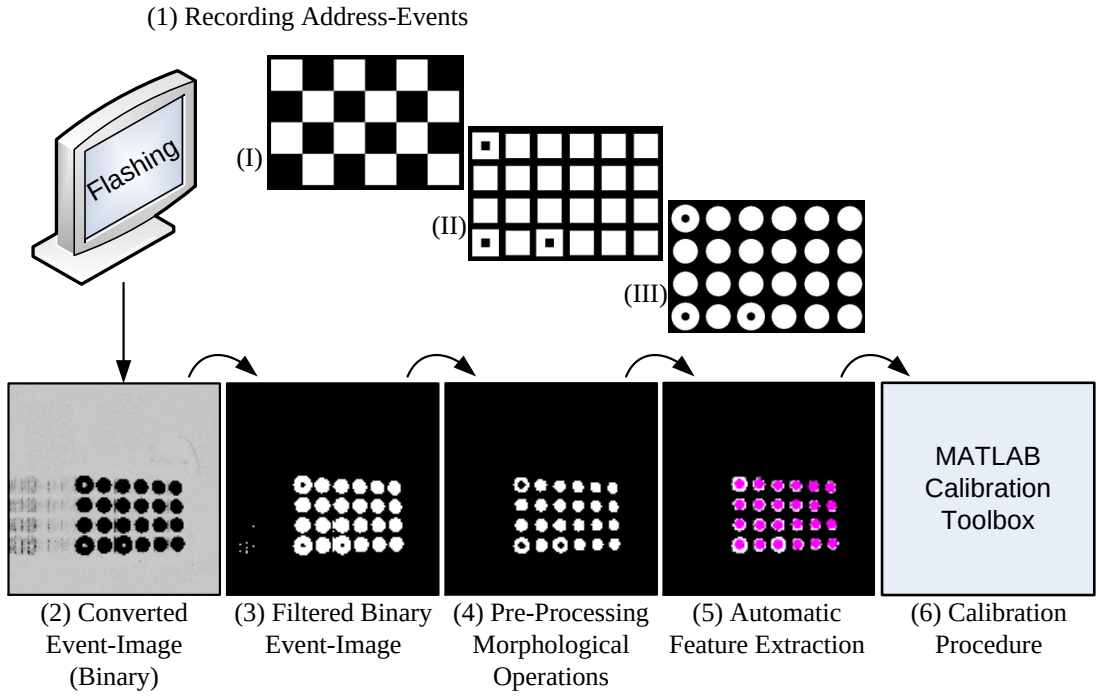


Figure 5.3: (1) shows the generation of the events by using a computer screen where one pattern is used and flashes continuously. (2) shows the converted binary event-image from the received events. In (3) filters were applied to clean the binary event-image before the feature extraction takes place. (4) presents the result of the pre-processing using morphological operations. In (5) the automatic feature points extraction detects the centers (purple dots) of the shown circle pattern, which are further used in (6) as input for the calibration toolbox.

well a method including a toolbox for the calibration of silicon retina sensors.

Additionally, as we are dealing with a stereo camera setup, the rectification parameters are calculated during the calibration procedure using Bouguet's calibration toolbox [11]. The rectification is the transformation of the images in such a way that the epipolar lines are parallel and correspond to the image lines. As mentioned in Section 3.3, the rectification and undistortion can be combined in a backward mapping [79], which is illustrated in Figure 5.4. Here, the backward mapping starts to transform the rectified and undistorted image (destination image) back to the distorted and unrectified image, where the coordinates will be subpixel coordinates (source image). In the case of grayscale images, an interpolation takes place to calculate the best representing grayscale value for the subpixel, which is used as the coordinates in the destination image.

Using backward mapping for a silicon retina camera fails because the interpolation step in the source image cannot be carried out in the same way for sparse address-

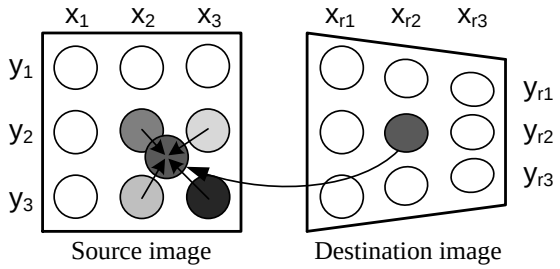


Figure 5.4: Shows the backward mapping from the destination image to the source image in case of monochrome cameras.

event data. For this reason, a forward mapping procedure has been implemented, as shown in Figure 5.5. The forward mapping does not exist in a closed form because the

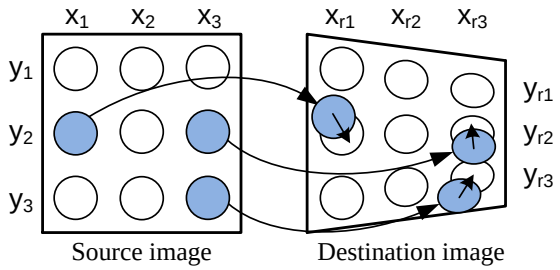


Figure 5.5: Shows the rectification based on forward mapping where each event from the source event-image is mapped to the destination event-image.

undistorted coordinates can only be calculated with an iterative approach. Thus, we use the approach suggested in a study by Heikkila and Silven [39], where the coordinate mapping is approximated. Forward mapping has the advantage that each received event of the source is considered and is assigned a destination coordinate. That means that no input event is lost or ignored, but on the destination event-image one coordinate can be assigned to two different source events.

5.2 Stereo Matching Approaches

This section explains the stereo matching approaches that we developed and implemented for sparse silicon retina event-based data. In the following, we distinguish between event-based and event image-based algorithms. Before the stereo matching methods are described, the handling of event streams received from the silicon retina camera and their conversion are presented.

5.2.1 Event to Event-Image Converter

In order to process the silicon retina sensor's raw data, we have implemented a converter interface, which transforms the data received from the silicon retina sensor into event lists or event-images to be processed by our developed algorithms.

5.2.1.1 Significance of Time History

The *time history* is an important parameter driving the amount of events considered for the conversion of events into a list or an image. A silicon retina sensor sends events asynchronously and time continuously based on the activity in front of the camera. A timestamp is attached to the data, which identifies the time of occurrence. This time stamp information is used for collecting event data based on the time history chosen. In general, static parts (e.g. objects without movement) of the scene are completely suppressed and will not be recognized from a stationary-mounted silicon retina sensor. Therefore, the time history must be set based on the scene dynamics to collect or consider enough events to obtain complete contours and distinguish data from noise. However, as the time history becomes longer, the contours of objects tend to blur. In Figure 5.6, the influence of the time history on the conversion process, which is explained in the next section in more detail, is illustrated. Figure 5.6(a) shows the grayscale image of a non-moving person captured by a monochrome stationary camera. Figure 5.6(b)

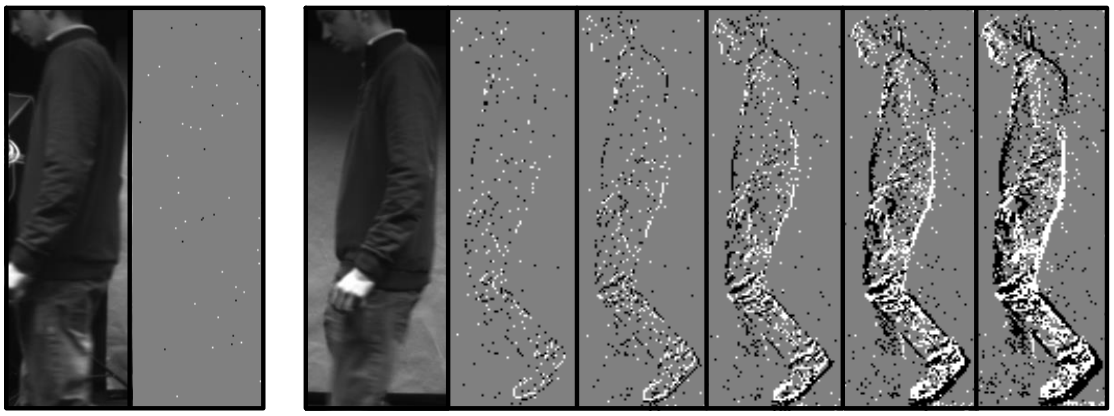


Figure 5.6: Silicon retina sensor in comparison to a conventional monochrome sensor. White pixels (on-events), black pixels (off-events), gray pixels (no events). (a) Person without movement in front of monochrome sensor, (b) silicon retina sensor output without movement, (c) person walking in front of monochrome sensor, (d)-(h) silicon retina sensor data from the walking person with events collected over a time period of 5ms, 10ms, 20ms, 40ms and 60ms.

shows the same scenario, but as an event event-image derived from the silicon retina camera, where *on-events* are marked white and *off-events* are shown in black. The pixels

where no event information (e.g. background without moving objects) from the silicon retina camera was received are marked in gray. Because the person is not moving, the silhouette of the person is not visible, and only very few events are present (largely representing noise). In contrast, Figure 5.6(c) shows an intensity image of a walking person. The same walking person observed with a silicon retina camera induces the event-generation behavior of this sensor. The sensor's high temporal resolution of 10ns was reduced for our experiments to 100μs. In this example, a temporal resolution of 100μs in conjunction with a collection period of 50 timestamps (i.e., 5ms) results in relatively incomplete contours (Figure 5.6(d)). The object's shape becomes more complete when events corresponding to more timestamps are collected. Figure 5.6(e-h) shows the events collected within a time history of 100 (10ms), 200 (20ms), 400 (40ms), and 600 (60ms) timestamps. The time history should be chosen to optimize complete object edges, as shown in Figure 5.6(f), without causing blurred object contours as shown in Figure 5.6(h). This illustrates that the time history is not a fixed parameter and plays an important role when input data for stereo matching algorithms are generated.

5.2.1.2 Conversion Process

The silicon retina camera sends an event stream based on the AER protocol as Section 2.3.2 stated. Based on the time history described in the previous section, we will now present the conversion process in more detail, along with four different output formats of the converter. In Figure 5.7 the data flow of the conversion process and the different types of output formats are presented. Figure 5.7(a) shows the events represented as list which is used in event-based stereo matching algorithms and (b-d) shows the image outputs used for event image-based algorithms. The events are transmitted continuously and all events $e(x, y, t)$ received between the time $t_C - h$, which represents the current time t_C minus the time history h, and the current time t_C are considered for the conversion process. All coordinates and timestamps within this time history form the event index set e_I given by

$$\begin{aligned} e_I := \{(x, y, t), ...\} \quad &, \text{with} \\ x \in \{1, ..., s_W\} \quad \wedge \quad y \in \{1, ..., s_H\} \quad &\wedge \quad t \in \{t_C - h, ..., t_C\}\,, \end{aligned} \tag{5.1}$$

where s_W and s_H describe the maximum value of the camera resolution in horizontal (width) and vertical (height) direction. In output (a), all events of the described event index set e_I are concatenated to an event list EL according to

$$EL = \bigotimes_{(x,y,t)\in e_I} (x, y, t, e(x, y, t)), \tag{5.2}$$

where $\bigotimes$ symbolizes the concatenation of the information from all events.

For the generation of event-images three different types are distinguished. Figure 5.7(b) shows an *Event* event-image $EI_e(x, y)$, Figure 5.7(c) shows a *Binary* event-image $EI_b(x, y)$ and Figure 5.7(d) shows a *Grayscale* event-image $EI_g(x, y)$. All three

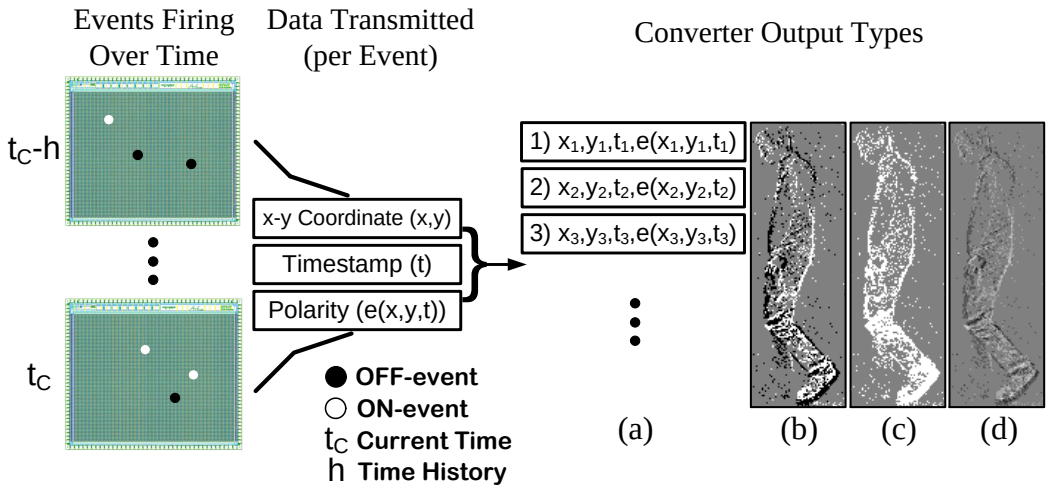


Figure 5.7: Conversion of address-events received from the silicon retina sensor to different output types. (a) Events encoded in an event list. (b) *Event* event-image with on-events (white), off-events (black) and background without information (gray). (c) *Binary* event-image with events in white and background in gray. (d) *Grayscale* event-image where events collected over time are represented by different gray values.

images are stored as 8-bit grayscale image and are initialized with a middle grayscale value g_m of the grayscale range 0-255 (grayscale range 0-255 $\hat{=}$ middle grayvalue 128). After this generation, the specific conversion procedure for one of these three individual event-images starts.

For the *Event* event-image $EI_e(x, y)$ shown in (b), the polarity of the events described by the event index set e_I is directly mapped into the event event-image according to

$$EI_e(x, y) = f_e(e(x, y, t)) \quad \forall (x, y, t) \in e_I \quad \text{, with} \tag{5.3}$$

$$f_e(\varepsilon) = \begin{cases} 0 & \text{if} \quad \varepsilon = +1 \quad \triangleq \quad \text{on-event} \\ 255 & \text{if} \quad \varepsilon = -1 \quad \triangleq \quad \text{off-event} \end{cases}, \tag{5.4}$$

where function f_e distinguishes the assignment of the grayscale value for an on- and off-event. This conversion was used for the converted images in Figure 5.6(b) and Figure 5.6(d-h), to illustrate the importance of the time history.

For the *Binary* event-image $EI_b(x, y)$ shown in (c) the polarity of the events within the event set e_I are directly mapped into the binary event-image according to

$$EI_b(x, y) = f_b(e(x, y, t)) \quad \forall (x, y, t) \in e_I \quad \text{, with} \tag{5.5}$$

$$f_b(\varepsilon) = \begin{cases} 255 & \text{if} \quad \varepsilon = +1 \quad \triangleq \quad \text{on-event} \\ 255 & \text{if} \quad \varepsilon = -1 \quad \triangleq \quad \text{off-event} \end{cases}, \tag{5.6}$$

where function f_b assigns for each event the grayscale value 255, which forms together with the grayscale value g_m the binary event-image.

The event event-image and binary event-image have one of three grayscale values representing on-events, off-events or background information g_m. In this case events occurring within the time period at the same spatial location override the previous information, which means a loss of information useful in correspondence search. Therefore, we use all events to form the *Grayscale* event-image $EI_g(x, y)$ shown in (d). Using all events described by the event index set e_I, the polarity of the event adds or subtracts a grayscale value gs to the grayscale event-image according to

$$EI_g(x, y) = EI_g(x, y) + f_g(e(x, y, t)) \quad \forall (x, y, t) \in e_I \quad \text{, with} \tag{5.7}$$

$$f_g(\varepsilon) = \begin{cases} +gs & \text{if} \quad \varepsilon = +1 \quad \triangleq \quad \text{on-event} \\ -gs & \text{if} \quad \varepsilon = -1 \quad \triangleq \quad \text{off-event} \end{cases}, \tag{5.8}$$

where function f_g determines a grayscale value gs which is added or subtracted depending on the occurrence of an on- or off-event. Events with different polarities occurring at the same spatial location during the time history, can cancel each other.

5.2.2 Event Image-based Stereo Matching

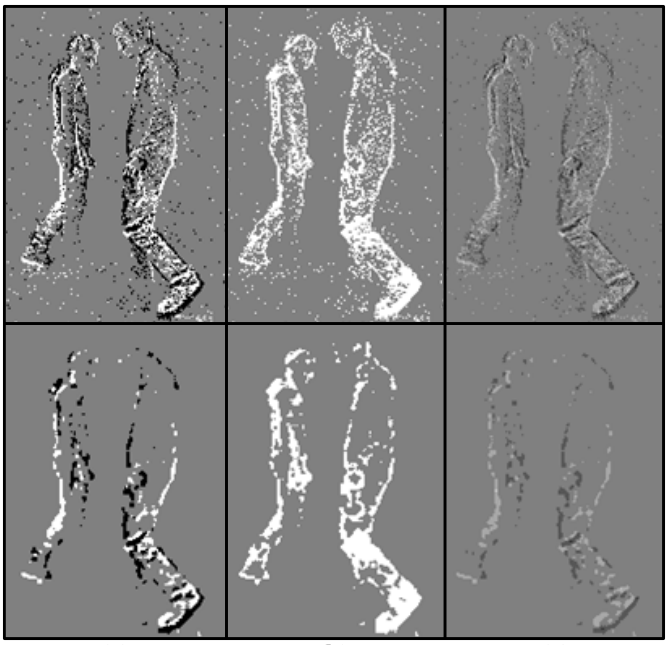
In the category of event image-based algorithms, we can distinguish between area-based and feature-based approaches. Before the two algorithm categories are described, two filtering options for the input event-images are discussed. This filtering reduces noise and outliers in the input event-image and increases the performance of the subsequently applied stereo matching algorithms.

5.2.2.1 Filtering Input Event-Images

For the filtering of the event-images a median filter and a connected component filter were implemented.

Median Filter The first filter, which is used for the filtering of the input event-images, is a median filter [15] with a kernel size of 3×3. In Figure 5.8, the first row shows the input images with the deactivated median filter and the row below shows the results when the median filter is applied. The filter removes the noise but we found that - depending on the time history - important parts of the camera data might be suppressed as well. Additionally, the median filter also changes the value based on the neighbor values, which is a data manipulation.

Connected Component Filter The second filter which we tested for the filtering of the input event-images is a connected component filter. Here, all three event-image types $EI_e(x, y)$, $EI_b(x, y)$ and $EI_g(x, y)$ are possible inputs for the filter. In the following, $EI_\#(x, y)$ represents all three event-image types. For each pixel (x, y) within the event-image $EI_\#(x, y)$, where the value is unequal to the value g_m, the filter operation is applied.



(a) (b) (c)

Figure 5.8: The effect of using a 3×3 median filter (second row) for filtering the input event-images (first row). (a) Event event-image, (b) Binary event-image and (c) Grayscale event-image.

Then, at the pixel location (x, y) the sum $s(x, y)$ of all values n of the neighborhood $N(x, y)$ (8-connected neighborhood) is calculated according to

$$s(x, y) = \sum_{n \in N(x,y)} EI_{\#}(n). \tag{5.9}$$

Based on the sum $s(x, y)$ can be determined with

$$EI_{\#}(x, y) = \begin{cases} EI_{\#}(x, y) & \text{if} \quad s(x, y) \neq 8 \cdot g_m \\ g_m & \text{if} \quad s(x, y) = 8 \cdot g_m \end{cases}, \tag{5.10}$$

if the pixel (x, y) has a neighbor unequal to g_m. A sum which is equal eight times the value g_m describes a pixel that has no direct neighbor (connected component) and is removed from the event-image $EI_{\#}(x, y)$ by overwriting the pixel (x, y) with g_m. In Figure 5.9, the first row again shows the input event-images without an applied filter, and the second row provides the results when the connected component filter is applied. The effects of the filtering are shown for event event-images in Figure 5.9(a), binary event-images in (b), and grayscale event-images in (c). In comparing the connected component filter with the median filter, we find that the connected component filter removes noise more carefully than the median filter. This can be explained by the fact that the connected

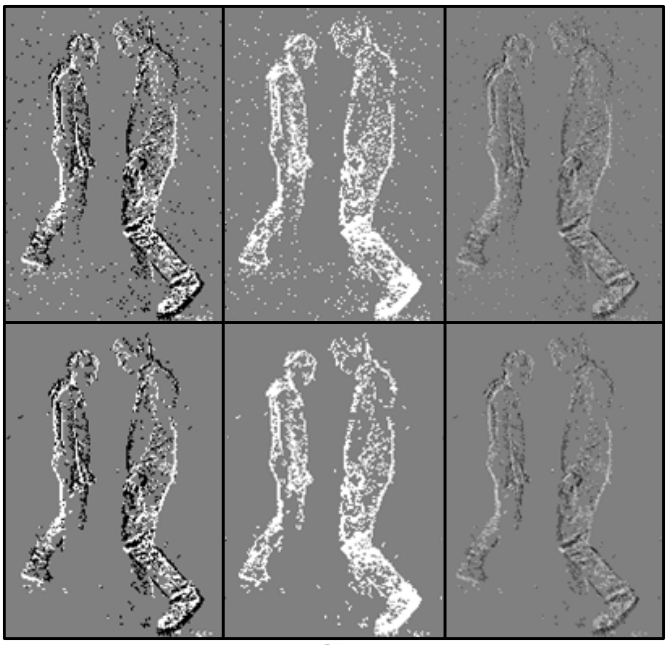


Figure 5.9: The effect of using a connected component filter (second row) for filtering the input event-images (first row). (a) Event event-image, (b) Binary event-image and (c) Grayscale event-image.

component filter only removes pixels if they have no direct neighbors different to g_m. In contrast the median filter removes or the same pixel. Even so, there are neighbors unequal to g_m and, in some cases, applying the filter on grayscale event-images changes the pixel value. This difference makes the connected component filter the preferred filter to use with event-images.

5.2.2.2 Area-Based Approaches

The first category of event image-based algorithms we consider is comprised of the area-based approaches. There exists a variety of cost calculation metrics, as introduced in Section 4.1.2. For our experiments, we have chosen seven metrics that are applied to event-images.

The following six metrics are applied to a grayscale event-image $EI_g(x, y)$. First correlation metric is the previously introduced SAD metric (Section 4.1.2), which calculates the cost values $C_{SAD}(x, y, d)$ for a given left EI_{g_l} and right EI_{g_r} grayscale event-image with

$$C_{SAD}(x, y, d) = \sum_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} |EI_{g\_l}(x + i, y + j) - EI_{g\_r}(x - d + i, y + j)|, \tag{5.11}$$

where m and n define the window size in horizontal and vertical direction and d the disparity.

The next correlation measure is the Zero-mean Sum of Absolute Differences (ZSAD) which calculates the costs C_{ZSAD} using

$$C_{ZSAD}(x,y,d) = \sum_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} |(EI_{g\_l}(x+i,y+j) - \overline{EI_{g\_l}}) - (EI_{g\_r}(x-d+i,y+j) - \overline{EI_{g\_r}})|. \quad (5.12)$$

Here, in comparison to the SAD metric, the average value of the left $\overline{EI_{g\_l}}$ and right $\overline{EI_{g\_r}}$ grayscale event-image is subtracted before the costs are calculated. In case of grayscale event-images, the average value is almost always the middle grayscale value g_m.

Another correlation metric is the Locally Scaled Sum of Absolute Differences (LSAD), which calculates the costs C_{LSAD} using

$$C_{LSAD}(x,y,d) = \sum_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} |EI_{g\_l}(x+i,y+j) - \frac{\overline{EI_{g\_l}}}{\overline{EI_{g\_r}}} \cdot EI_{g\_r}(x-d+i,y+j)|. \quad (5.13)$$

The next three correlation metrics are similar to the previous three except that the squared differences of the pixels in the window are summed up, according to Equation (5.14) for the Sum of Squared Differences (SSD), Equation (5.15) for the Zero-mean Sum of Squared Differences (ZSSD) and Equation (5.16) for the Locally Scaled Sum of Squared Differences (LSSD).

$$C_{SSD}(x,y,d) = \sum_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} (EI_{g\_l}(x+i,y+j) - EI_{g\_r}(x-d+i,y+j))^2 \quad (5.14)$$

$$C_{ZSSD}(x,y,d) = \sum_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} ((EI_{g\_l}(x+i,y+j) - \overline{EI_{g\_l}}) - (EI_{g\_r}(x-d+i,y+j) - \overline{EI_{g\_r}}))^2 \quad (5.15)$$

$$C_{LSSD}(x,y,d) = \sum_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} (EI_{g\_l}(x+i,y+j) - \frac{\overline{EI_{g\_l}}}{\overline{EI_{g\_r}}} \cdot EI_{g\_r}(x-d+i,y+j))^2 \quad (5.16)$$

The seventh correlation metric differs from the previous six metrics and is based on the definition in Section 4.1.2 that described non-parametric local transforms introduced by Zabih and Woodfill [100]. This algorithm, based on the Census transform, is called in the following *event transform* (ET) and is applied on event event-images. In Figure 5.10 we describe the principle of the event transform and the differences to the census transform. Figure 5.10(a) shows the left and right input event event-images with the windows considered. The correlation using the dual-state logic is illustrated in Figure 5.10(b) and using the tri-state logic in (c). Before the costs can be calculated, vectors are created, which encode the pixels within the considered window. Using the

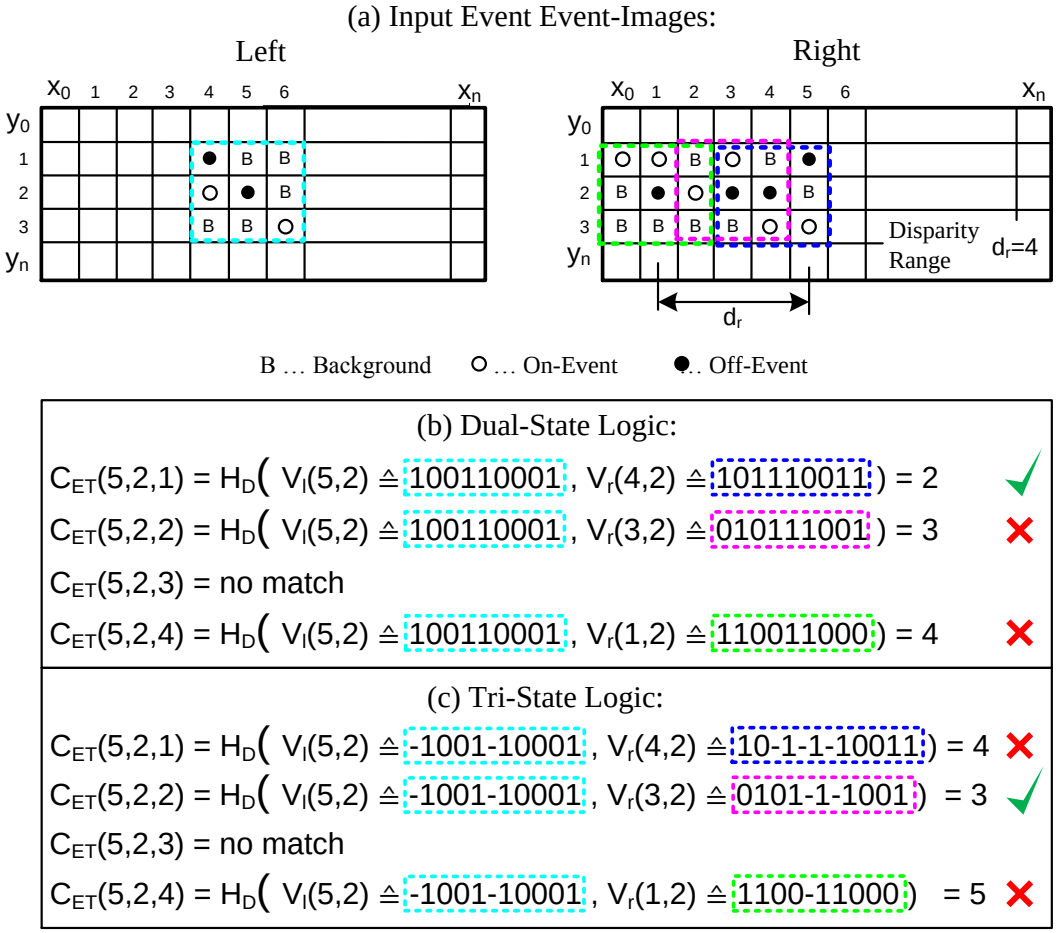


Figure 5.10: Matching of the input event event-images (a) using the event transform: The neighborhood of the matching candidate is encoded in a bit vector ((b) dual-state logic or (c) tri-state logic) which is used for calculating the matching costs.

dual-state logic the vector image $V(x, y)$ for a considered pixel is constructed according to

$$V(x,y) = \bigotimes_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \bigotimes_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} f_d(EI_e(x+i, y+j)), \tag{5.17}$$

where $\bigotimes$ symbolizes value-wise concatenation of the $m \times n$ neighborhood using the

function f_d according to

$$f_d(\varepsilon) = \begin{cases} 1 & \text{if} \quad \varepsilon = 0 \quad \triangleq \quad \text{on-event} \\ 1 & \text{if} \quad \varepsilon = 255 \quad \triangleq \quad \text{off-event} \\ 0 & \text{if} \quad \varepsilon = g_m \quad \triangleq \quad \text{background} \end{cases} . \tag{5.18}$$

If the neighbor is a background pixel g_m then a 0 is concatenated to the vector, and if the neighbor is an on- or off-event the added value to the vector is 1. To use the tri-state logic, the function f_d is replaced by the function f_t which calculates the values for the vector according to

$$f_t(\varepsilon) = \begin{cases} 1 & \text{if} \quad \varepsilon = 0 \quad \triangleq \quad \text{on-event} \\ -1 & \text{if} \quad \varepsilon = 255 \quad \triangleq \quad \text{off-event} \\ 0 & \text{if} \quad \varepsilon = g_m \quad \triangleq \quad \text{background} \end{cases} . \tag{5.19}$$

The difference with the tri-state logic is that the on-event is encoded with 1 and the off-event with -1. A neighbor which represents a background pixel g_m is encoded as before with 0.

After the vectors of the window patches have been generated and stored into the vector image, the costs for each pixel $C_{ET}(x, y, d)$ at disparity d within the disparity range d_r are calculated with

$$C_{ET}(x, y, d) = H_D(V_l(x, y), V_r(x - d, y)) \tag{5.20}$$

$$H_D(v_1, v_2) = \sum_{i=0}^{(m \times n)-1} v_1[i] \neq v_2[i]. \tag{5.21}$$

The function H_D computes the Hamming distance [35] between a vector v_1 from the left vector image V_l and a vector v_2 from the right vector image V_r. The Hamming distance describes the number of different elements (values) between the two vectors, and less dissimilar values correspond to lower cost and indicate a better match.

5.2.2.3 Feature-Based Approaches

Feature-based matching approaches represent the second category of event image-based algorithms we use for our experiments with silicon retina stereo data. Before the matching takes place, the event-image is processed and analyzed to extract features that can be used for the correspondence search. Various properties of an image can be chosen as features for further processing. Examples of such features are lines, points, segments, and many more. For our silicon retina stereo matching, the following two feature matching approaches are chosen.

The first feature-based approach is a *Center of Gravity* (COG) matcher, which extracts objects in the binary event-image and calculates the geometric center of the extracted segments. These segment centers are involved in the correspondence search to retrieve

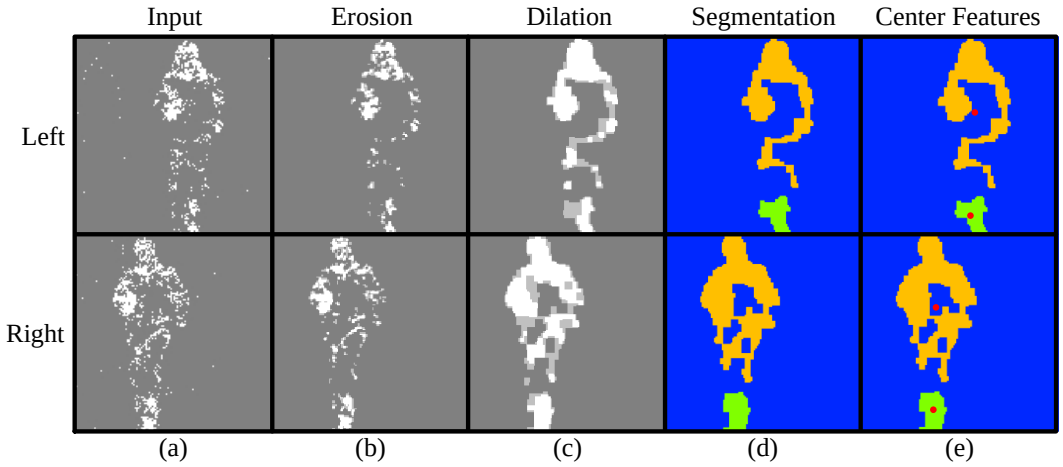


Figure 5.11: Work flow of the center of gravity stereo matching algorithm. The images show in (a) the binary event-images, (b) the results after the erosion step, (c) the output after the dilation step, (d) the segmented and labeled images and (e) the extracted center of gravity features used for matching (red dots).

the depth values. In Figure 5.11 the work flow of the COG stereo matching algorithm is shown. In Figure 5.11(a) the binary event-images of the left and right camera are depicted before erosion (5.11(b)) and dilation (5.11(c)) [34] take place. For both morphological operations, which represent an additional filtering of the binary event-images, a circular structure element is used. After the morphological pre-filtering, we apply a segmentation step using a flood-filling algorithm [15]. This algorithm groups and labels connected pixels within blobs, which is illustrated in Figure 5.11(d). In the next step, the labeled segments are analyzed and the geometrical center (center of gravity) of the segment is calculated. The center represents the whole segment within the matching process. In Figure 5.11(e) the red dots symbolize the center of gravity features matched in the next step. The whole segment is represented by a point, which is used for the matching and calculating of the depth for all pixels within the segment. This, as well as the fact that the center of gravity from the same segment in the left and right image can have different y-coordinates, make this approach faulty. Additionally, the approach is specialized for single objects or objects without overlapping edges, which reduces the flexibility and range of usage. The possible displacement in the y-coordinate is considered during the matching process to achieve comparable results. In the experimental result section, we will evaluate the COG algorithm and compare the results with the second implemented feature-based stereo approach.

The second feature-based matching approach we evaluate with silicon retina stereo data is a *corner feature* (CF) matching algorithm. The CF algorithm is based on features described in the work of Shi and Tomasi [80], which is implemented in the function

goodFeaturesToTrack of the OpenCV[1] [13] library. In Figure 5.12 the work flow of the corner feature matcher is presented. In Figure 5.12(a) the extracted corner features of

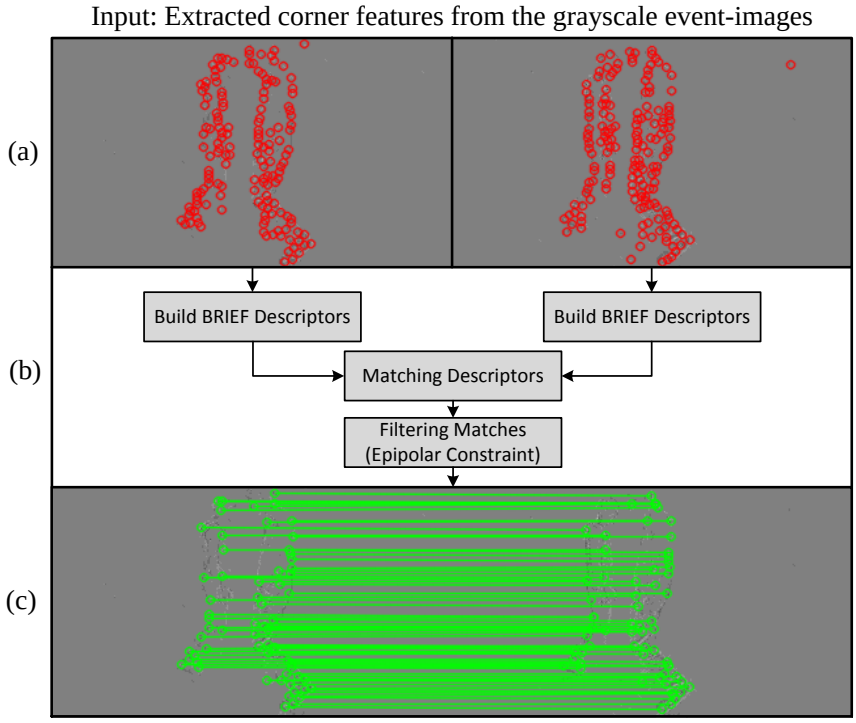


Figure 5.12: (a) Detected corner features (red circles) of the left and right grayscale event-image. (b) Workflow of the corner feature matcher. (c) Corresponding corner features (green circles) are connected with green lines.

the left and right grayscale event-image are shown. After the feature detection, the work flow for finding corresponding matches is shown in Figure 5.12(b). The workflow starts with the extraction of image descriptors. Here, the *Binary Robust Independent Elementary Features* (BRIEF) descriptors presented in the work of Calonder *et al.* [16] are used. These descriptors are very competitive to the often used *Speeded Up Robust Features* (SURF) descriptors and calculated in a fraction of time. A brute force matcher of OpenCV is used to match the constructed descriptors of the left and right image. Finally, filtering checks if the found matches fulfil the epipolar constraint. In a rectified setup, only features which are on the same y-coordinate fulfil the epipolar constraint, but the possible localization of the features on slightly different y-positions requires a consideration of matches on different y-lines. In Figure 5.12(c) the final matched features (connected by solid lines) are shown. Besides the search space of the line position, we

[1]Open Source Computer Vision Library - `http://opencv.org/`

also considered other parameters during the evaluation of the corner feature matcher, which are presented in the experimental results section.

5.2.3 Event-based Stereo Matching

The second category of matching algorithms this study used is the event-based stereo matching algorithms which do not require the conversion of events to event-images prior to matching. Instead of an event-image, we use an event list as described in Figure 5.7.

The proposed algorithm for the event-based category is a time correlation (TC) stereo matching approach. In this approach, we use the time of occurrence of the events and the spatial location as correlation information. Here, the costs $C_{TC}(x, y, d)$ at disparity d within the disparity range d_r are calculated according to

$$C_{TC}(x, y, d) = f_m(EL_l[i_l], EL_r[i_r], d) \ \ldots$$
$$\forall i_l = [0 \ldots n_l - 1] \wedge i_r = [0 \ldots n_r - 1] \quad \text{, with} \tag{5.22}$$

$$f_m(s_l, s_r, d) = \begin{cases} f_c(t_{s_l}, t_{s_r}) & \text{if} \quad y_{s_l} = y_{s_r} \wedge (x_{s_l} - d) \geq (x_{s_r} - d_r) \wedge e_{s_l} = e_{s_r} \\ c_M & \text{if} \quad \text{otherwise} \end{cases}, \tag{5.23}$$

where n_l and n_r describe the number of entries in the left EL_l and right EL_r event list. The matching function f_m calculates for a pair of event data (s_l, s_r) the costs using the function f_c (see Equation (5.24)) if the y-coordinate of the left y_{s_l} and right y_{s_r} event data are the same, the x-coordinate between the left x_{s_l} and right x_{s_r} event data is within the disparity range d_r and the polarity of the left e_{s_l} and right e_{s_r} event is the same. Otherwise the cost of the position (x, y, d) of the DSI is set to the value c_M which describes a predefined cost value stored in the DSI needed to search for matching candidates. The maximum costs have to be set greater than the time history h, because the best match has a cost 0 which would lead to problems during the search for matching candidates.

For the cost calculating function f_c two different timestamps (t_l, t_r) are used to calculate the cost, based on the chosen method m according to

$$f_c(t_l, t_r) = \begin{cases} t_l - t_r & \text{if} \quad m = 0 \\ 0 & \text{if} \quad m = 1 \wedge (t_l - t_r) \leq 1 \\ log(t_l - t_r) \cdot s & \text{if} \quad m = 1 \wedge (t_l - t_r) > 1 \end{cases}, \tag{5.24}$$

where the costs for the events are calculated with an inverse linear method ($m = 0$) or a logarithmic method ($m = 1$). If the time difference is equal or smaller than 1, then the costs for the logarithmic function are set to 0. For a time difference greater than 1 and method where $m = 1$, the result of the logarithmic function is multiplied by scaling factor s, which is adapted based on the time history.

In Figure 5.13, the matching process using the time as correlation criterion is shown in more detail. For the demonstrated example of cost calculation, the inverse linear method is used. The timestamp of the considered event of the left event list t_{e_1} is 10,

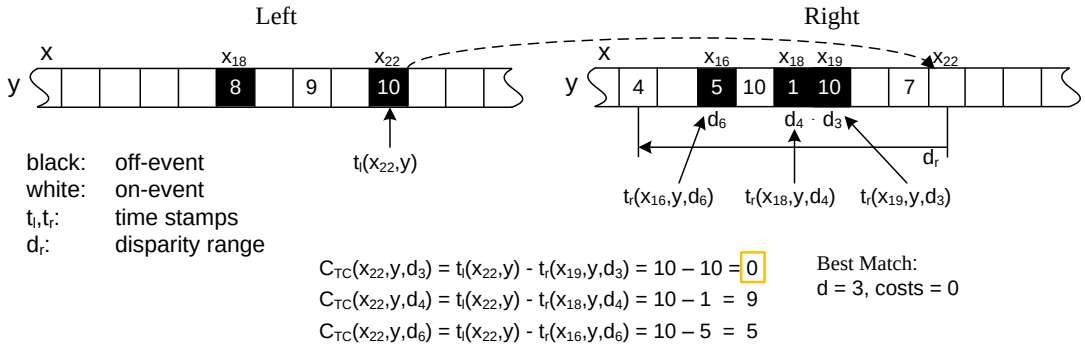


Figure 5.13: Time-based matching: The time differences between events are used as matching costs.

which represents the current time. In the right event list, all timestamps t_{e_2} of events with the same polarity and within the disparity range d are compared with the timestamp of the considered event on the left side. The events with the lowest time difference between their timestamps likely represent the best match.

5.3 Improvement Techniques for Silicon Retina-based Stereo Matching Algorithms

Before the following improvement techniques were applied to the stereo matching algorithms, all introduced stereo matching approaches were tested and evaluated independently. We used these improvement techniques to increase the accuracy of the overall stereo matching performance. There are two main approaches that we suggested. First, we developed a modified version of the global optimization *Belief Propagation* (BP) method, which was adapted to work with silicon retina data; secondly, we developed different post-processing methods such as average filter, median filter, and a novel approach called *Two-Stage Postfilter* (further called also as *Two-Stage Filter* (2SF)).

5.3.1 Sparse Belief Propagation Improvement Method

The first improvement technique we suggest is a *Belief Propagation* (BP) [89] approach. BP is also known as a sum-product message passing algorithm, and is applied on initial stereo matching results to improve the matching quality. In the literature, different versions can be found that focus not only on the stereo matching improvement, but also try to optimize the BP approach itself in terms of processing time and memory consumption [104]. Even though BP mainly benefits from the smoothness assumption between neighbors in dense cost volumes, we have chosen BP to be used with sparse stereo input. The reason is its potential for adapting the global smoothness term in a way that operates locally with connected groups of disparities. We expected especially

sparse data with little local information to benefit from this information gain. In the following, we present our adaptations of BP to improve the stereo matching results.

Before going into detail about the adaptations for sparse data, we will first explain the BP method itself. First, the initial matching costs

$$C_I(x,y,d) = \sum_{i=-\lfloor \frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{j=-\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} |I_l(x+i,y+j) - I_r(x-d+i,y+j)| \tag{5.25}$$

are calculated. I_l and I_r denote the left and right grayscale event-images, and d represents the disparity. Here, a sum of absolute difference function (SAD) with an aggregation window size of $m \times n$ is used. Second, the global energy function [104] defined according to

$$E = \sum_{p \in P} \left\{ E_d(p_x, p_y, d_p) + \sum_{q \in N_p} E_s(d_p, d_q) \right\}, \tag{5.26}$$

where P describes the set of all pixels in the image and N_p denotes the 4-connected neighborhood of pixel p, is minimized. $E_d(p_x, p_y, d_p)$ represents the cost of assigning disparity d_p to pixel p, which is called data costs (data term) and is extracted from the DSI calculated in Equation (5.25). The second term $E_s(d_p, d_q)$ of the equation is the smoothness term, which represents the smoothness costs between neighboring pixels p and q if p is assigned to disparity d_p and q to disparity d_q. The calculation of the smoothness cost is presented in Equation (5.30). BP is an iterative procedure that seeks to minimize the mentioned energy costs. In each iteration j, the pixel p updates its neighbors' costs using a D-dimensional message M_{pq}^j, as shown in Figure 5.14(a). This message sends the belief (costs) in what pixel q's disparity could be, given by

$$M_{pq}^j(d_q) = \min_{d_p \in D} \left\{ E_d(d_p) + E_s(d_p, d_q) + \sum_{q' \in N_p \setminus q} M_{q'p}^{j-1}(d_p) \right\}, \tag{5.27}$$

where D is the set of all possible disparities. N_p represents the set of all neighbors of p and $q' \in N_p \setminus q$ are all neighbors of p except q. Figure 5.14(a) shows the generated message at iteration j for one neighbor using the beliefs of the other neighbors at $j-1$. In Figure 5.14(b) the graph with the message update for all neighbors at iteration j is shown. Each pixel has sent its beliefs to all four neighbors, who use these beliefs in the next iteration as information for generating new messages as described in Equation (5.27). After a certain number of iterations J, the final step of BP is to sum (also called *aggregation*) the costs shown in Figure 5.14(c) and search for the best disparity of each pixel, which can be expressed as

$$d_p = \arg\min_{d \in D} \left\{ E_d(d) + \sum_{q \in N} M_{qp}^J(d) \right\}. \tag{5.28}$$

$E_d(d)$ are representing the costs of the data term, calculated in Equation (5.25), where the summed costs of the neighbors N are added. The lowest costs are associated with the best match d_p which is chosen for the disparity map.

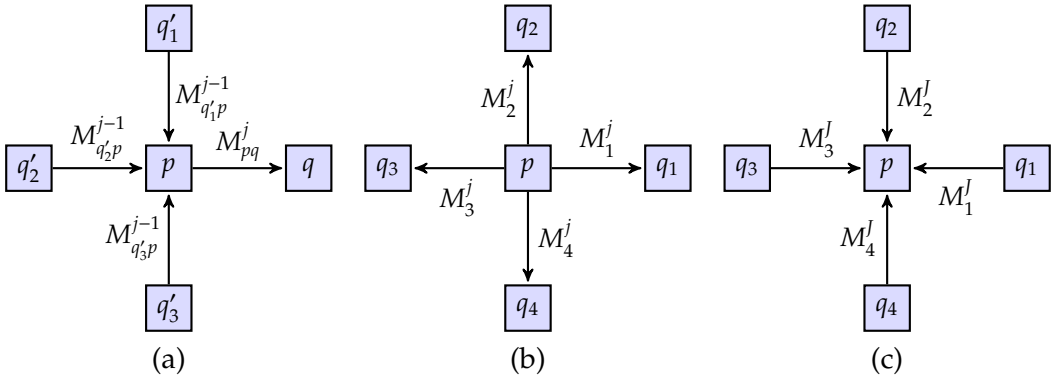


Figure 5.14: BP message flow: (a) Shows the messages sent to one neighbor (p to q) at iteration j depending on the other three neighbors' belief at $(j-1)$. (b) Shows the messages sent from one pixel at iteration j to its 4-connected neighborhood. (c) Indicates the final message passing after J iterations where all neighbors' beliefs contribute for the evaluation of pixel p.

One problem with this method is that the BP algorithm in its original version works in conjunction with neighbors in the 2D image grid, and expects cost values from the 4-connected neighborhood to optimize the overall energy $E(d)$ of Equation (5.26). In the beginning, the initial matching costs C_I are calculated by using Equation (5.25) with $I_l = EI_l$ and $I_r = EI_r$, where EI_l and EI_r describe the left and right grayscale event-images from silicon retina sensors. The DSI filled with matching costs C_I exists due to use of grayscale event-images that are not densely filled, which means that many locations of the DSI have a value of zero. This causes failure of the propagation steps. Therefore, a maximum cost value evaluation within the DSI takes place before starting the update of the neighbors' costs, using a D-dimensional message M^j_{pq}. The maximum costs are calculated according to

$$C_{Imax}(x, y) = \max_{d \in D} \Big\{ C_I(x, y, d) \Big\}, \tag{5.29}$$

where $C_{Imax}(x, y)$ is the maximum of a pixel's matching costs within the defined disparity range D. If $C_{Imax}(x, y)$ is not equal to zero, the processing of the current considered pixel continues, otherwise it is skipped because there is no possible match for pixel p. Now, if the neighbor pixel q has a value unequal to the background value g_m of the grayscale event-image, the calculation of the belief and its corresponding message occurs. If the cost values $E_d(d_p)$ within the disparity range of the considered pixel are zero, a replacement cost value must be assigned to calculate the beliefs of the neighbors. For this reason, we use the calculated cost value $C_{Imax}(x, y)$ of Equation (5.29) to represent the values which are zero within the DSI. If there is only one possible match, which has the cost value $C_{Imax}(x, y)$, then the belief calculation would negatively influence the message generation process. To avoid this influence over the message, the final replacement value for values equal to zero within the DSI is two times the calculated

cost value $C_{Imax}(x, y)$. The calculation of the smoothness costs $E_s(d_p, d_q)$ is done with assignment of a penalty according to

$$E_s(d_p, d_q) = \begin{cases} 0, d_p = d_q \\ \frac{P_e}{8}, |d_p - d_q| = 1 \\ \frac{P_e}{2}, |d_p - d_q| = 2 \\ P_e, |d_p - d_q| \geq 3 \end{cases}, \tag{5.30}$$

where the penalty depends on the neighbors' disparity difference and a constant value P_e. Another important implementation step is the division of the third term of Equation (5.27) by the number of neighbors N_{msg} who have sent costs unequal to zero because of the sparse DSI. In case all neighbors have sent costs equal to zero, the sum of all neighbors' beliefs is set to two times the previously introduced $C_{Imax}(x, y)$. The result of the final aggregation step given in Equation (5.28) also needs to be divided by the number of neighbors N_{msg} who are unequal to zero. In case the result of the final aggregation is equal to zero, the costs are set to $2 \cdot C_{Imax}(x, y)$. These calculation steps are necessary to make the BP work with the sparse data within the DSI, where many cost values equal to zero are present. The proposed adaptations enable the usage of BP-based algorithms with sparse grayscale event-images.

5.3.2 Post-processing Improvement Methods

As input for the post-processing improvement techniques, we use the disparity maps from the stereo matching algorithms to get more robust and accurate depth information. We employ three different filters for the post-processing which include two standard filters (average filter and median filter), which are presented in Section 5.3.2.1. The third filter introduced in Section 5.3.2.2 is more advanced and optimized for the operation on sparse disparity maps.

5.3.2.1 Simple Filters

The average filter and median filter [15] used for the post-processing of the sparse disparity maps are applied with window sizes of 3×3 and 5×5. Both filters operate on the disparity map calculated from the stereo matching algorithm. In Figure 5.15 the impact of the average and median filter is shown. Image (a) shows the original disparity image delivered by the stereo matching algorithm. The average filter shown with a window size of 3×3 in (b) and 5×5 in (c) changes the values but does not remove disparity values during the filtering process. In contrast, the median filter, based on the window size of the filter, removes a disparity value if more than half of the values within the considered window are zero. Image (d) shows the results of the 3×3 median filter and image (e) the results of the median filter using a 5×5 window. Using the 5×5 window many of the disparity values are completely removed and the outcome changes significantly, which shows that the post-processing filter must be chosen carefully where there is sparse data.

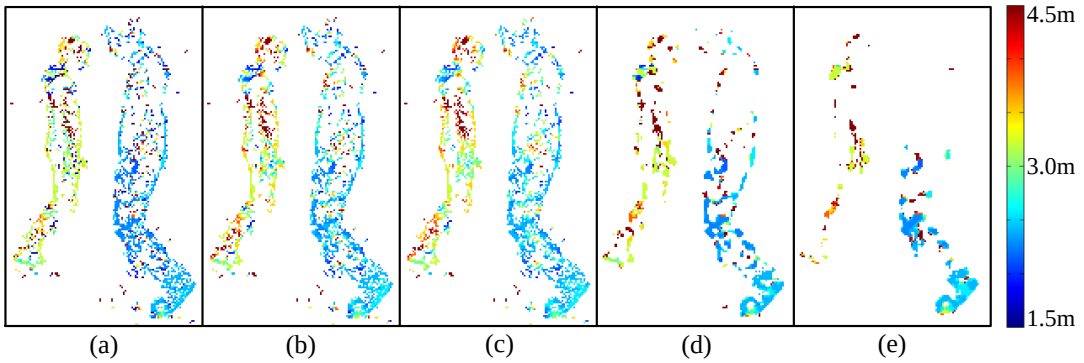


Figure 5.15: Showing the impact of the average filter 3×3 in (b) and 5×5 in (c). The results of the median filter using a 3×3 window is shown in (d) and a 5×5 window in (e). For the comparison of the filter outcome, the original stereo matching result is shown in (a).

5.3.2.2 Two-Stage Filter

The third post-processing filter was specifically adjusted to the operation on sparse disparity maps. This filter works in two steps, and therefore, we call it the *Two-Stage Filter* (2SF).

In comparison to standard filters, such as the average filter or median filter, which were designed for dense image processing, the proposed filter is especially designed for sparse input data and considers an 8-connected neighborhood with a certain radius as shown in Figure 5.16. First for each disparity value d, an array $L1_d^j \in \mathbb{R}^{r_m \times 8}$ at iteration j is calculated with

$$L1_d^j[r,i] = DM_{N_i}^j(x+r, y+r) \qquad \forall r \in \{1 \dots r_m\} \wedge i \in \{0 \dots 7\}, \tag{5.31}$$

where for each direction N_i (8-connected neighborhood) the values of the disparity map DM within the radius r_m are collected and stored in $L1_d^j$. Because of the sparse character of the disparity map the array contains disparity values which are equal to zero. Now a median filter f_m is applied to all disparity values not equal to zero within the array $L1_d^j$. For each direction N_i the median value is calculated and stored in an array $L2_d^j \in \mathbb{R}^{1\times 8}$ given by

$$L2_d^j[i] = f_m(L1_d^j[1,i], L1_d^j[2,i], \dots, L1_d^j[r_m,i]) \qquad \forall i \in \{0 \dots 7\}. \tag{5.32}$$

$L2_d^j$ stores now all median values of all directions. In a second step, a median filter f_m is once again applied to all values of $L2_d^j$ unequal to zero, resulting in the final disparity value d.

$$SDM_d^j = f_m(L2_d^j[0], L2_d^j[1], \dots, L2_d^j[7]) \tag{5.33}$$

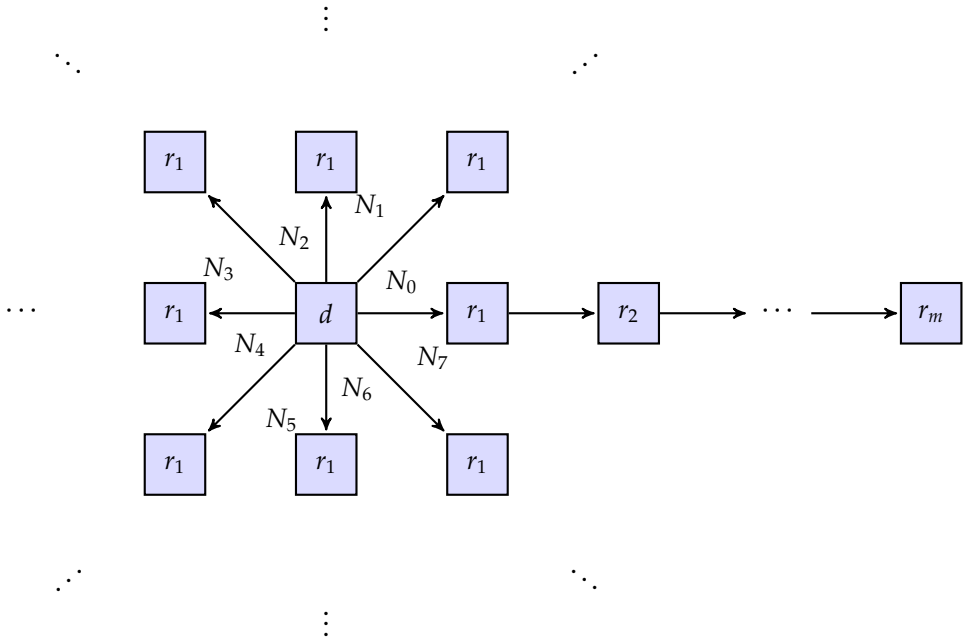


Figure 5.16: 8-connected neighborhood of disparity value d with radius r_m used for the 2SF.

defines the filtered and smoothed disparity map SDM_d^j for further processing. Now, depending on the number of iterations J, the disparity map is considered as final $DM_d^J = SDM_d^J$ or used for the next iteration cycle.

5.4 Summary

In this chapter, we have introduced stereo matching algorithms for silicon retina cameras. Because the specific sparse data format consists of only two states (on- and off-events), the calibration and rectification tasks needed to be adapted. Additionally, we described an event-to-event-list and an event-to-event-image conversion process that prepares the received data according to the demands of the stereo matching algorithm applied. On one hand, we converted the events to event-images in order to apply stereo matching algorithms that are also suitable for conventional stereo camera systems. On the other hand, we developed stereo matching approaches using the event data as event list without event-image generation to benefit from the specific data of the silicon retina cameras. Both approaches have advantages and draw backs and, therefore, we also addressed improvement techniques that seek to increase the quality of the results delivered by the stereo matching algorithms. The overall improvement of silicon retina-based stereo matching results that can be achieved by these algorithms will be evaluated and discussed in Chapter 6.

CHAPTER 6

Experimental Results

In this chapter, the newly developed stereo matching algorithms are evaluated. Before the results of the stereo matching algorithms are presented, the evaluation methods are explained. The tests are divided into two test series. In the first test series (TS1) the silicon retina sensor with a resolution of 128×128 is used. The second test series (TS2) uses the silicon retina sensor with a resolution of 304×240.

6.1 Evaluation Methods

As mentioned in Section 4.3, standard evaluation platforms and methods do not suit silicon retina-based stereo matching. Therefore, we developed an evaluation platform using synthetic silicon retina sensor data, which is described in Section 6.1.1. This evaluation is more suitable for laboratory tests, and does not represent real-world scenes. For testing the algorithms with real-world scenes an evaluation method was implemented that considered real objects first as planar surfaces (described in Section 6.1.2), and second as complex curved surfaces (presented in Section 6.1.3). Thus, it was necessary to develop an approach which uses real silicon retina sensor output data for evaluation. For TS1 in Section 6.2, the evaluation described in Section 6.1.2 was used, where the distance of objects was measured and used as reference for the evaluation. This method works accurately for planar objects located in parallel to the image plane of the camera. However, to improve the evaluation of silicon retina-based data, a more accurate method was developed. Instead of assuming a planar object representation, this method evaluates all pixels separately. That means that the evaluation of curved shapes and different objects at different positions is possible. This approach is presented in Section 6.1.3 and is used for evaluation of all stereo matching approaches presented within TS2 in Section 6.3.

6.1.1 Testing with Synthetic Data

A first evaluation platform for silicon retina-based stereo matching algorithms generates both synthetic event data streams [87] for the stereo matching algorithms and ground truth data for the subsequent evaluation step. To test the functional behavior and correct operation of the silicon retina stereo matching algorithms, we have developed the tool called *Event Editor* [87]. This tool allows the generation of synthetic silicon retina stereo data, as well as the evaluation and analysis of the results. The event editor interface, presented in Figure 6.1, shows, for example, a vertical line moving from top left to bottom right. The tool consists of a graphical user interface for visualizing the event

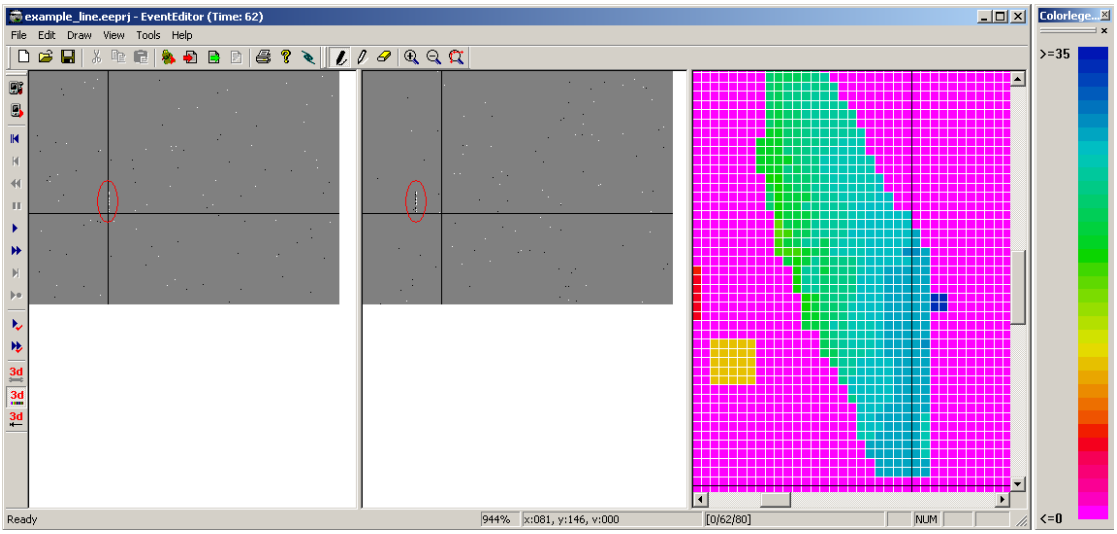


Figure 6.1: Software tool *Event Editor* used for generating synthetic silicon retina sensor data. Scenario shows a line moving from the upper left to the bottom right corner. Left window: left camera view; Middle window: right camera view; Right window: disparity map generated by the stereo matching algorithm.

input data and the stereo matching results in the form of a disparity map. Additionally, the event editor offers a Python[1] interface for textual input of commands controlling the input data and evaluation process. To generate the synthetic input data, Python scripts are used and all objects are defined by size, shape and moving direction. The synthetic data points generated do not use an exact model of a silicon retina camera and, therefore, the asynchronous characteristics are not considered and algorithms tailored to the silicon retina sensor cannot be tested under real-world conditions. Real-world application thus requires an additional evaluation with real-world data.

[1] `http://www.python.org`

6.1.2 Testing with Assuming Planar Objects at Fixed Distances

For the evaluation of stereo matching algorithms under real-world conditions, test data of real scenarios are used. In case of the real-world data the ground truth has to be defined as well. To facilitate the ground truth generation, objects in front of the camera are considered as ideal planar objects at a certain distance, which is used as ground truth distance for the evaluation of the TS1 presented in Section 6.2. Figure 6.2 shows some examples, assuming planar objects. The ground truth distance between the object

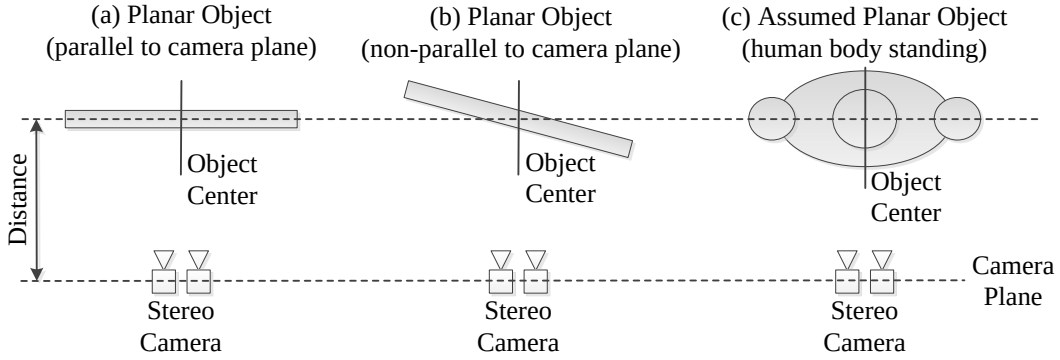


Figure 6.2: Assumption of planar objects for evaluation stereo matching algorithms. (a) Planar object fronto-parallel to the image plane, (b) planar object slanted to the image plane, and (c) a human body assumed as planar object fronto-parallel to the image plane.

centers and the camera plane was measured with a laser distance meter. This evaluation method works well for objects that are planar and parallel to the image plane, as shown in Figure 6.2(a), because all object pixels then have the same distance. If planar objects (Figure 6.2(b)) are slanted with respect to the image plane, some object points are closer and other ones are farther from the camera. This results in inaccurate evaluation because the distance at the object's center is used as ground truth. Here, the obvious defect of this approach is evident since real objects are not ideally planar, as illustrated in Figure 6.2(c). The person's shape with different distances along its silhouette cannot be evaluated accurately with this approach. Therefore, we developed another method described in Section 6.1.3 and used the method for TS2.

6.1.3 Testing Pixel-wise with Complex and Curved Objects

The evaluation method described in this subsection generates ground truth data by capturing the same scene as the silicon retina stereo sensor with a second conventional stereo camera system. This enables a pixel-wise comparison of calculated silicon retina stereo matching results with the ground truth depth values offered by the second camera system. Though the depth data calculated by the conventional stereo sensor exhibits a limited accuracy, this accuracy is considered sufficient for our experiments and therefore,

is considered to be ground truth. This method is used for the evaluation of the TS2 presented in Section 6.3.

6.1.3.1 Ground Truth System Setup

Enabling the pixel-wise evaluation of the silicon retina stereo matching results requires the measurement of the ground truth depth values from the test scenes shown in Figure 6.9. We used conventional monochrome cameras in a stereo set-up that was designed to achieve at least twice of the accuracy of the silicon retina stereo system, in order to obtain valid reference data. In Figure 6.3, the stereo system in the white box represents the silicon retina stereo system, which is under evaluation. This system has a rigid connection to the monochrome reference stereo vision system above, which is marked with a dashed bounding box. The reference system shown in Figure 6.3 consists

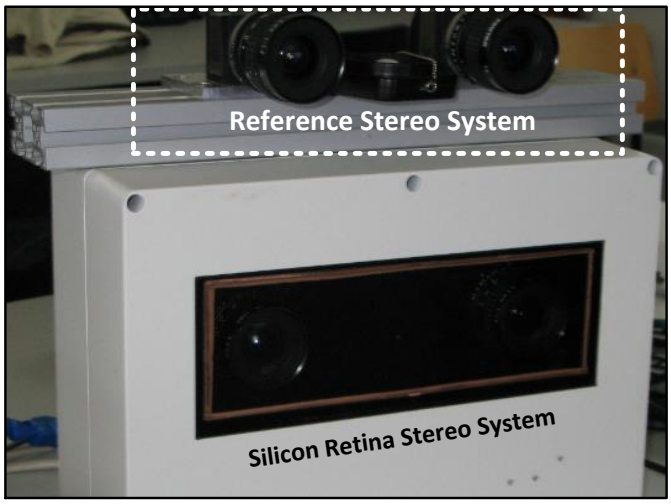


Figure 6.3: Camera set-up for ground truth generation. The white box holds the embedded silicon retina stereo system, the stereo system in the dashed bounding box above acts as the reference stereo camera system.

of two *Imaging Development Systems*[1] (IDS) cameras (model UI-1220SE-M-GL Rev.2) which are mounted on a rigid baseline of 0.12m. The cameras transmit their images to the PC, where further processing takes place. The rectified images are processed with a sufficiently accurate and reliable Census-based stereo matching algorithm. Details about the stereo matching engine can be found in the work of Humenberger *et al.* [49].

For the measurement of the reference system's accuracy, objects were placed at different distances. All distances were measured with a laser distance meter device and compared with the depth output of the stereo algorithm. The accuracy was evaluated in the range where the tests took place. The average distance error in the range of interest is shown in Table 6.1. The results show that the reference system has at close distances

[1] `http://en.ids-imaging.com/`

distance	avg err	error
1.0m	0.012m	1.20%
1.5m	0.017m	1.13%
2.0m	0.027m	1.35%
2.5m	0.040m	1.60%
3.0m	0.081m	2.69%
3.5m	0.117m	3.35%
4.0m	0.220m	5.48%

Table 6.1: Evaluation of the distance accuracy of the reference stereo vision system. In the testing range, the depth algorithm output is compared with the real distance measured by a laser distance meter.

till 2.5m an error of less than 1.6%, and for distances between 2.5m and 4.0m of less than 5.48%.

6.1.3.2 Calibration of Ground Truth Setup

Before the ground truth-based testing can be done, both stereo camera systems need to be calibrated and registered onto each other in a way that they have a pixel congruent representation of the scene in front of them. In the calibration step, both stereo heads are calibrated separately. The reference system, as well as the silicon retina stereo system, use the same calibration procedure as described in the work of Zhang [103] and adapted in our work [27]. The only difference is the pattern used. For the reference system, the classic checkerboard calibration pattern is captured in different positions to provide the necessary feature points. In contrast, the silicon retina system uses a circle pattern flashing on a computer display to generate stimuli for the retina sensors, and later to extract the feature points for the calibration step. In this case, either the computer display or the silicon retina stereo camera can be moved to capture the necessary different views. After the calibration step for both stereo heads, all calibration and rectification parameters are available and are further used in the registration process.

6.1.3.3 Registration of Ground Truth Setup

For the registration of both cameras onto one another, in order to achieve a common understanding of the scene, the left image is used as reference for the depth map. Therefore, the registration took place between the left views of the stereo systems.

In a first experimental setup, we tried to register both camera systems to a common world coordinate system. The results have shown that the accuracy depends on the exact calculation of the origin point and orientation of the world coordinate system. The overall problem with this approach was that the transform had to be done in 3D space, where we additionally (to the estimation of R and t) had to deal with reconstruction

uncertainty. This lack of accuracy led us to another approach with more promising results. The second approach is based on homographies, which represent the projective transformation between two planar spaces. Thus, we use a homography H according to

$$p_{ref} = H \cdot p_{sr}, \tag{6.1}$$

with p_{sr}, p_{ref} in homogeneous coordinates which is written element by element as

$$\begin{pmatrix} x_{ref} \\ y_{ref} \\ z_{ref} \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \cdot \begin{pmatrix} x_{sr} \\ y_{sr} \\ z_{sr} \end{pmatrix}, \tag{6.2}$$

where a point $p_{sr} = (x_{sr}, y_{sr}, z_{sr})^T$ from the silicon retina stereo camera image (left) is transformed to a point $p_{ref} = (x_{ref}, y_{ref}, z_{ref})^T$ in the reference stereo camera image (left). The homography H is determined to match one certain plane for a set of given feature points, as shown in Figure 6.4. All feature points represented by the green circles are

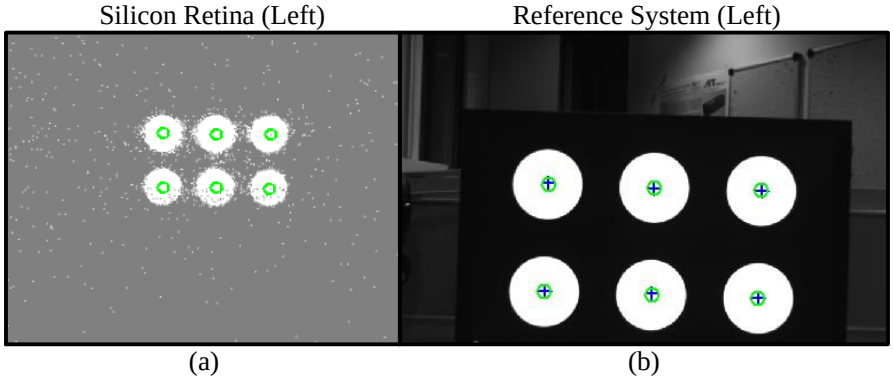


Figure 6.4: (a) Silicon retina camera image. (b) Reference image. The green circles in the left and right image represent the extracted feature points, and crosses in the right image show the feature points after transformation using the homography.

the extracted feature points. Image (a) shows the silicon retina image with the extracted feature points p_{sr} from the left sensor, and the image (b) shows the extracted feature points p_{ref} from the reference image of the left sensor. The blue crosses in image (b), mark the transformed feature points, estimated with the calculated homography at a distance of 1m.

Using only this homography will lead to errors when applying it to other distances. Therefore, the homography H was calculated at different distances d (in meters), where $d \in M := \{1, 1.5, 2, 2.5, 3, 3.5, 4\}$. For each of these distances the corresponding homography $H(d)$ was calculated using the singular value decomposition (SVD) [86].

Before the SVD can be used the Equation (6.2) is transformed into the Euclidean form by dividing with z_{ref} and setting $z_{sr} = 1$, which leads to

$$\frac{x_{ref}}{z_{ref}} = \frac{h_{11}x_{sr} + h_{12}y_{sr} + h_{13}}{h_{31}x_{sr} + h_{32}y_{sr} + h_{33}} \tag{6.3}$$

$$\frac{y_{ref}}{z_{ref}} = \frac{h_{21}x_{sr} + h_{22}y_{sr} + h_{23}}{h_{31}x_{sr} + h_{32}y_{sr} + h_{33}}. \tag{6.4}$$

Now, the equations are multiplied by the denominator and rearranged according to

$$\frac{x_{ref}}{z_{ref}} \cdot (h_{31}x_{sr} + h_{32}y_{sr} + h_{33}) - h_{11}x_{sr} - h_{12}y_{sr} - h_{13} = 0 \tag{6.5}$$

$$\frac{y_{ref}}{z_{ref}} \cdot (h_{31}x_{sr} + h_{32}y_{sr} + h_{33}) - h_{21}x_{sr} - h_{22}y_{sr} - h_{23} = 0, \tag{6.6}$$

which can be written as system of linear equations according to

$$Dh = \begin{bmatrix} -x_{sr1}, -y_{sr1}, -1, 0, 0, 0, \frac{x_{ref1} \cdot x_{sr1}}{z_{ref1}}, \frac{x_{ref1} \cdot y_{sr1}}{z_{ref1}}, \frac{x_{ref1}}{z_{ref1}} \\ 0, 0, 0, -x_{sr1}, -y_{sr1}, -1, \frac{y_{ref1} \cdot x_{sr1}}{z_{ref1}}, \frac{y_{ref1} \cdot y_{sr1}}{z_{ref1}}, \frac{y_{ref1}}{z_{ref1}} \\ \vdots \\ -x_{srN}, -y_{srN}, -1, 0, 0, 0, \frac{x_{refN} \cdot x_{srN}}{z_{refN}}, \frac{x_{refN} \cdot y_{srN}}{z_{refN}}, \frac{x_{refN}}{z_{refN}} \\ 0, 0, 0, -x_{srN}, -y_{srN}, -1, \frac{y_{refN} \cdot x_{srN}}{z_{refN}}, \frac{y_{refN} \cdot y_{srN}}{z_{refN}}, \frac{y_{refN}}{z_{refN}} \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0, \tag{6.7}$$

where N describes the number of feature points and fills the matrix D with $2N$ rows. Together with the vector

$$h = [h_{11}h_{12}h_{13}h_{21}h_{22}h_{23}h_{31}h_{32}h_{33}]^T, \tag{6.8}$$

which describes all elements of the homography and forms a system of linear equations, which has to be solved under the constraint $h \neq 0$. For a system with more than four point correspondences the solution of

$$arg \min_{||h||=1} ||Dh|| = arg \min_{||h||=1} h^T D^T Dh = \lambda_{min} \tag{6.9}$$

is wanted, where λ_{min} represents the smallest eigenvalue of $D^T D$. To find the eigenvector which corresponds to the smallest eigenvalue the SVD is applied given by

$$[USV] = SVD(D(p_{sr}(d), p_{ref}(d))), \tag{6.10}$$

where the columns of U contain the eigenvectors of DD^T and the columns of V the eigenvectors of $D^T D$ corresponding to the singular values of the diagonal in matrix S.

Since we searched for the vector h an eigenvector with the eigenvalue closest to zero, we used the last column of matrix V, which corresponds to the smallest eigenvalue of $D^T D$. This last column of V represents the solution for vector $h = V(:, 9) \in \mathbb{R}^{9x1}$ in Equation (6.8), which gives the coefficients of our searched homography $H(d)$ in the form of

$$H(d) = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}. \tag{6.11}$$

After the *SVD*, an optimization step f_r

$$H_r(d) \quad = \quad f_r(H(d)), \tag{6.12}$$

using the Levenberg-Marquardt [63] algorithm, takes place to obtain the refined homography $H_r(d)$.

After this step, the homographies for the seven defined distances are available, but the distances in between are still missing. For this reason, an interpolation step was done to determine a polynomial function of degree 4 to approximate the homography of each position in the range between 1m and 4m. The polynomial of degree 4 was chosen because it achieved the best results in our experiments, in which we tested polynomials of degree 3, 4 and 6. All of the homographies calculated in Equation (6.12) for the distances $d \in M$ are used to calculate a coefficient vector $C \in \mathbb{R}^{5x1}$, which represents the coefficients for a polynomial of degree 4. The polynomial curve fitting function f_p is used to calculate the vector C for each element of the homography $H = (h_{i,j})_{i,j=1..3}$ with

$$C(h(i, j)) = f_p(H_r(d, i, j)) \ \ \forall \ \ d \in M. \tag{6.13}$$

Now, for a certain distance d_n all elements of the vector C are used to calculate with

$$\begin{aligned} H_n(d_n, i, j) = C(h(i, j)_1) \cdot d_n^4 + C(h(i, j)_2) \cdot d_n^3 + \ldots \\ C(h(i, j)_3) \cdot d_n^2 + C(h(i, j)_4) \cdot d_n + C(h(i, j)_5) \end{aligned} \tag{6.14}$$

the elements of a new homography H_n.

To check the accuracy of the homographies at the distances $d_n \in M_n := \{1.25, 1.75, 2.25, 2.75, 3.25, 3.75\}$ (in meters), the coefficient vectors C described in Equation (6.14) are used. In Table 6.2 the displacement of the calculated pixel positions in relation to the real measured pixel positions in x- and y-direction are shown. The average pixel error in x- and y-direction is less than 2 pixels, which is an acceptable accuracy for our evaluations, because we assume that, largely, a similar depth value is present within a 2 pixel neighborhood. This means that in the evaluation, we accept the fact that sometimes a value is evaluated wrongly; but this has, as shown in the experiments, only minor effects on the total results.

6.2 Test Series 1

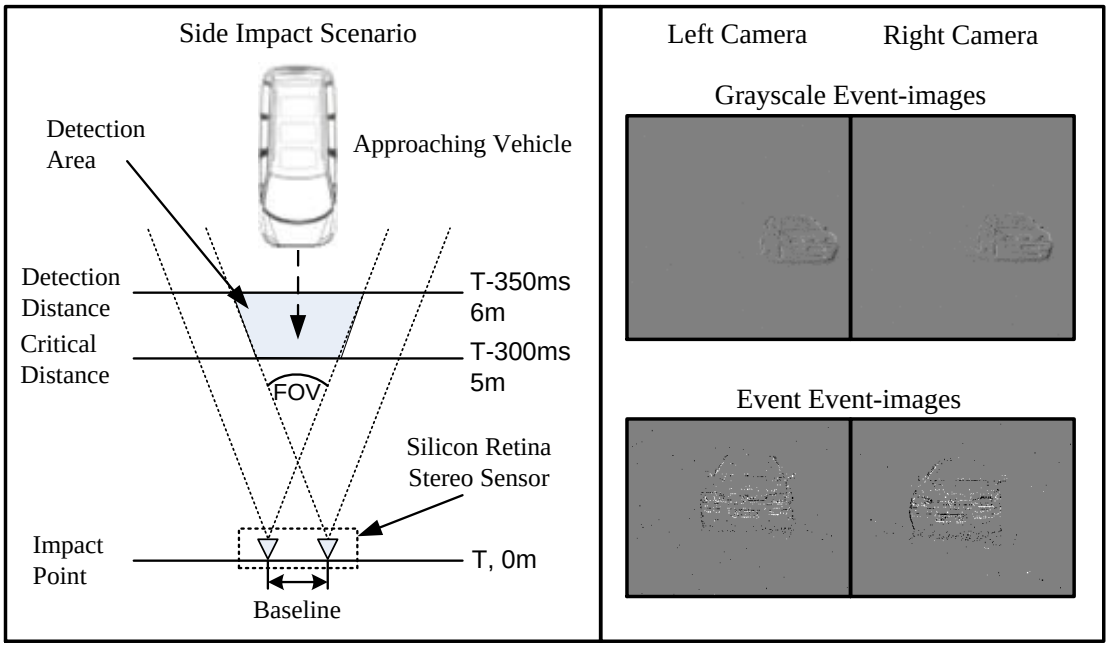
In the first test series (TS1) a silicon retina sensor with 128×128 and a temporal resolution of 1ms was used. Two cameras of this type were rigidly mounted to form a stereo sensor,

distance [m]	avg pix err x [px]	avg pix err y [px]
1.25	0.83	1.50
1.75	0.67	0.67
2.25	0.67	1.67
2.75	0.17	0.67
3.25	0.67	0.83
3.75	1.67	1.33

Table 6.2: Accuracy and displacement of the calculated pixel positions in relation to the real measured pixel positions in x- and y-direction.

which had been designed for an application of the silicon retina sensor in an automotive application. The goal of this application was to use a silicon retina stereo sensor for pre-crash side impact detection, where a depth resolution of 0.3m at a distance of 5 to 6 meters was needed. Based on these requirements, the two sensors were mounted with a baseline of 0.45m and lenses with a focal length of 8.5mm were chosen. In Figure 6.5(a) a scenario is shown where the stereo sensor was mounted on the side of the car and should have detected approaching vehicles with a maximum speed of 60 km/h, that may have caused a side impact. The detection must be made before the 5 meter limit is reached so that the system then has the time necessary to activate the pre-crash safety features, such as pretensioner or side airbag preparation. Figure 6.5(b) shows an image pair of an oncoming car, once converted and displayed as grayscale images (first pair) and once as on- and off-events (second pair). Another assumption we made for the first tests, was that the ego motion of the stereo sensor is zero. This means that the side impact detection is carried out when the stereo sensor is not moving. An additional focus within the project was to test the ability of the silicon retina stereo sensor in different traffic environments, where with different visibility and lighting condition had to be dealt with. The functional behavior of the algorithms used in TS1 was tested with synthetic data, as described in Section 6.1.1, and the evaluation with real-world data was done with the method explained in Section 6.1.2.

The algorithms evaluated and compared within TS1 were an area-based sum of absolute difference (SAD) algorithm and a feature-based center of gravity (COG) stereo matching approach. Due to the lack of ground truth data, indoor test data with persons at different distances were used for the evaluation of the algorithms instead of automotive data. The persons were considered to be planar objects at a certain distance. In Figure 6.6 the test data in 2m (a), 4m (b) and 6m (c) distance are shown. The top row represents input data for the SAD algorithm and the bottom row the input data for the COG stereo matching algorithm. The event data received were converted into images, and 500 images were used to calculate an average error rate. The average error was determined with the method presented in Section 6.1.2, where, for the SAD algorithm, each calculated depth value was compared with the ground truth depth value mea-



(a) (b)

Figure 6.5: (a) Silicon retina stereo sensor used in a scenario for side impact pre-crash detection. (b) Grayscale event-images from the left and right camera in the first row and a pair of event event-images in the second row.

sured. In the case of the feature-based stereo matching, the depth of the feature centers was compared with the depth of the measured ground truth distance.

Based on the time resolution of 1ms and the speed of the observed objects, time histories of 10 and 20 timestamps were chosen.

6.2.1 Evaluation of SAD Matching Approach

The SAD approach was tested with four different window sizes, which were set to 3×3, 5×5, 7×7, and 9×9. The results of the SAD stereo matching algorithm are presented in Figure 6.7. The results show that for farther objects a longer time history of 20ms is more appropriate because more grayscale values needed the matching are available in this timeframe. Considering the window sizes no major difference is visible, but a slight trend towards better results with bigger window sizes is evident.

6.2.2 Evaluation of COG Matching Approach

In the tests with the COG approach, morphological operations [34] were applied to the binary event-images. First an erosion to remove noise and outliers, and second a dilation to close holes and support edges. The shape of the kernel element is a circle with

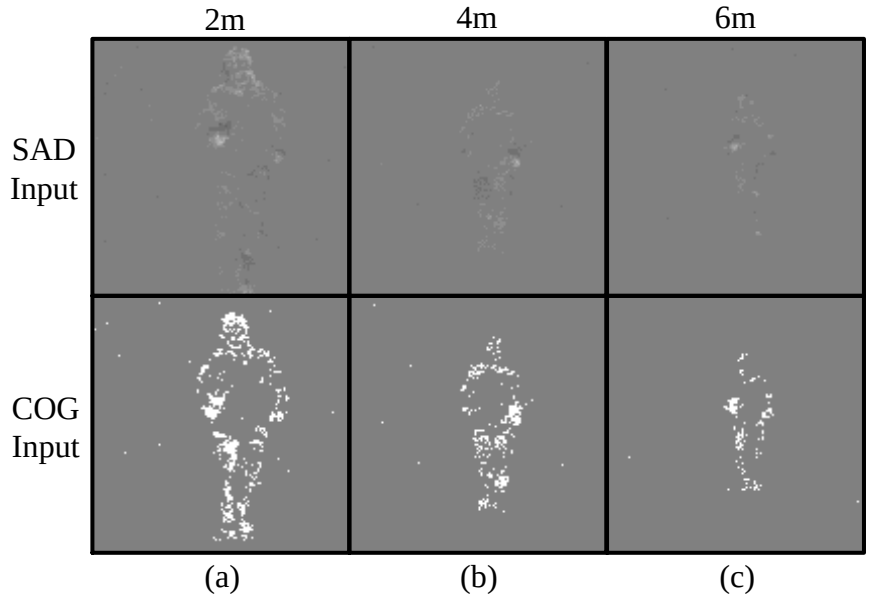


Figure 6.6: Test input data in distances of (a) 2m, (b) 4m and (c) 6m for the SAD algorithm shown in the first row and for the COG algorithm in the second.

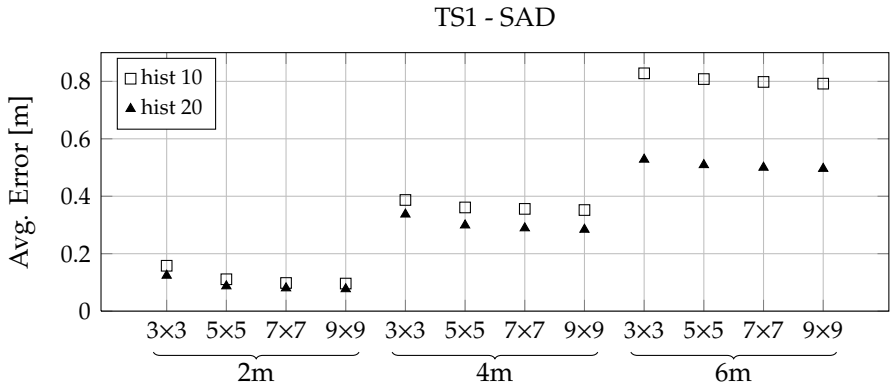


Figure 6.7: Evaluation of the SAD algorithm with a history of 10 and 20 timestamps and different window sizes.

varying radii which are set during the experiments to erosion/dilation combinations of 3/5, 5/9 and 3/11 pixels. The results of the COG stereo matching approach are illustrated in Figure 6.8. In case of the COG matching, only the center of an object is used for calculating the distance, which means that this single center match is used to represent and describe the depth of the whole object. Various erosion and dilation sizes generate different results, but the combination 5/9 generates optimal matching results for all investigated distances. Regarding the time history, for distances of 2m and 4m, no

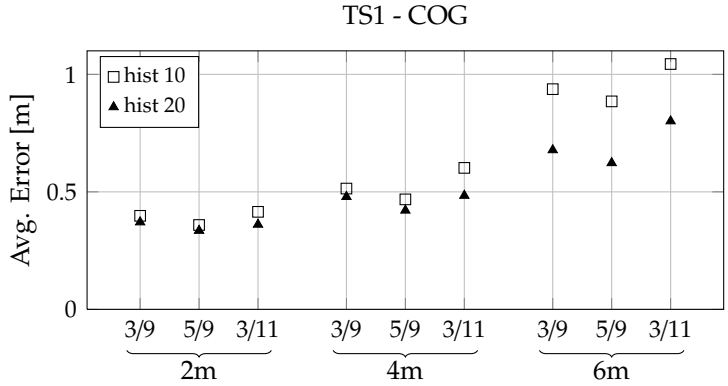


Figure 6.8: Evaluation of the COG feature-based algorithm with a time history of 10 and 20 timestamps and different kernel sizes of the morphological operators.

significant difference is visible between 10 and 20 timestamps. A time history of 20 performs better for the 6m distance. Overall, it remains a challenging task to set the parameters of the erosion/dilation kernels and time history using the COG approach.

To summarize the evaluation of both algorithms in TS1, the SAD algorithm performs better given the average distance error of the three evaluated distances.

6.3 Test Series 2

The second test series (TS2) is based on a silicon retina stereo sensor consisting of two retina cameras with a spatial resolution of 304×240 and a temporal resolution of 100μs. This sensor has a higher resolution than the one used for TS1 and is used for all further experiments and evaluations of the different stereo matching algorithms in this thesis. The sensor was designed for a variety of applications, and the baseline was chosen to be 0.15m. Furthermore, for the evaluations in this section we used lenses with a focal length of 8.5mm. All test data used for the evaluation of the different algorithms were recorded indoors between 1m and 4m. For evaluation we used the pixel-wise ground truth method presented in Section 6.1.3. In TS2, we used four test data sets (called A, B, C, and D in the following), which are shown in the first row of Figure 6.9. Test data set A shows one person walking in parallel to the camera plane at a distance of 2.5m and another one at 3.5m distance. In test data set B, two persons are walking at distances of 2.5m and 3.5m, respectively, while partly overlapping. The third test data set C consists of a human torso sitting in front of the camera and moving at a distance of 1.5m. Test data set D shows a striped rotating disc at a distance of 1.5m for stimulation of events. The second row in Figure 6.9 shows the ground truth depth maps used for the evaluation of the stereo matching results.

The silicon retina sensor is set to a temporal resolution of 100μs, which is an important setting for the time history used for our test data sets. Based on the temporal resolution and the test data sets used, the time history was chosen in the range of 20-

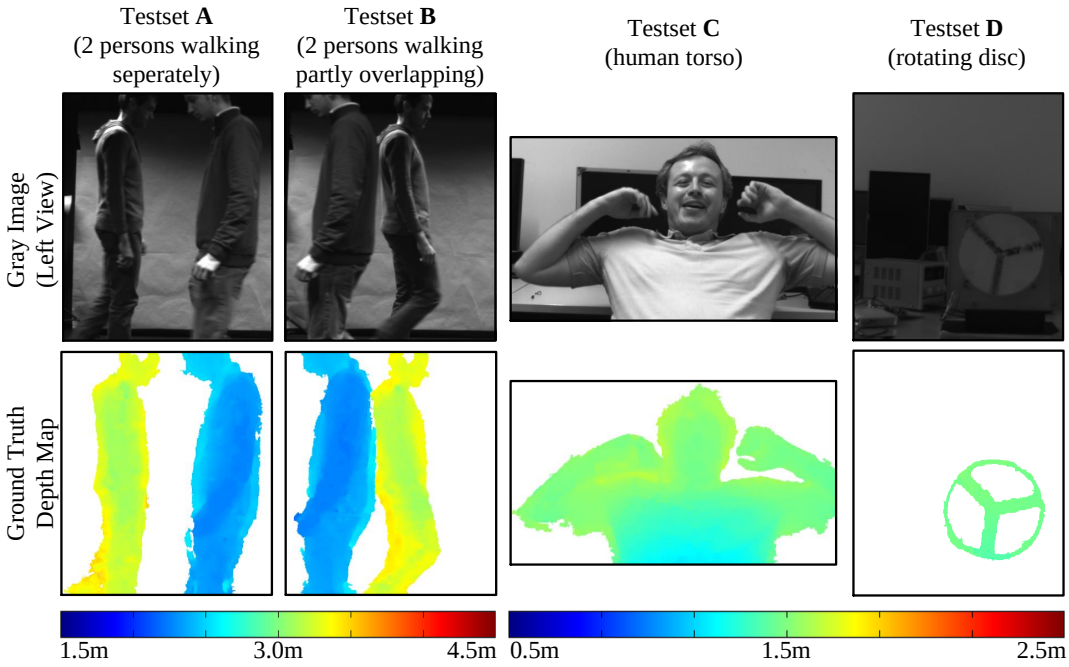


Figure 6.9: The four test data sets A-D used in TS2, first row from left to right: A) two persons walking at 2.5m and 3.5m, B) two persons walking partly overlapping at 2.5m and 3.5m, C) human torso moving at 1.5m, and D) a rotating disc at 1.5m. In the second row, the ground truth data corresponding to each test data set is shown.

600 timestamps, or 2ms-60ms. This range was chosen in order to maximize complete objects contours without blurring effects. Figure 6.10 shows the examples of generated grayscale event-images for all four data sets. These input images are examples of the gray scale images used for the stereo matching algorithms evaluated. During the evaluation, time histories are used to generate different input images from the shown examples in Figure 6.10.

In addition to the average distance error, we calculate two ratios, which are used as confidence value as well as providing us with information on how reliable the calculated average distance error is. A low average error indicates accurate results, but does not note how many values are contributing to the calculation of this average error. Therefore, two event ratios are introduced which describe the relation between input and output. The first value is the ratio R_D calculated by

$$R_D = \frac{Disparities\ Calculated}{All\ Input\ Events}, \tag{6.15}$$

which describes the relation between the amount of disparity values calculated and the input events considered for the calculation. This is a measure of the density of the output in comparison to the input.

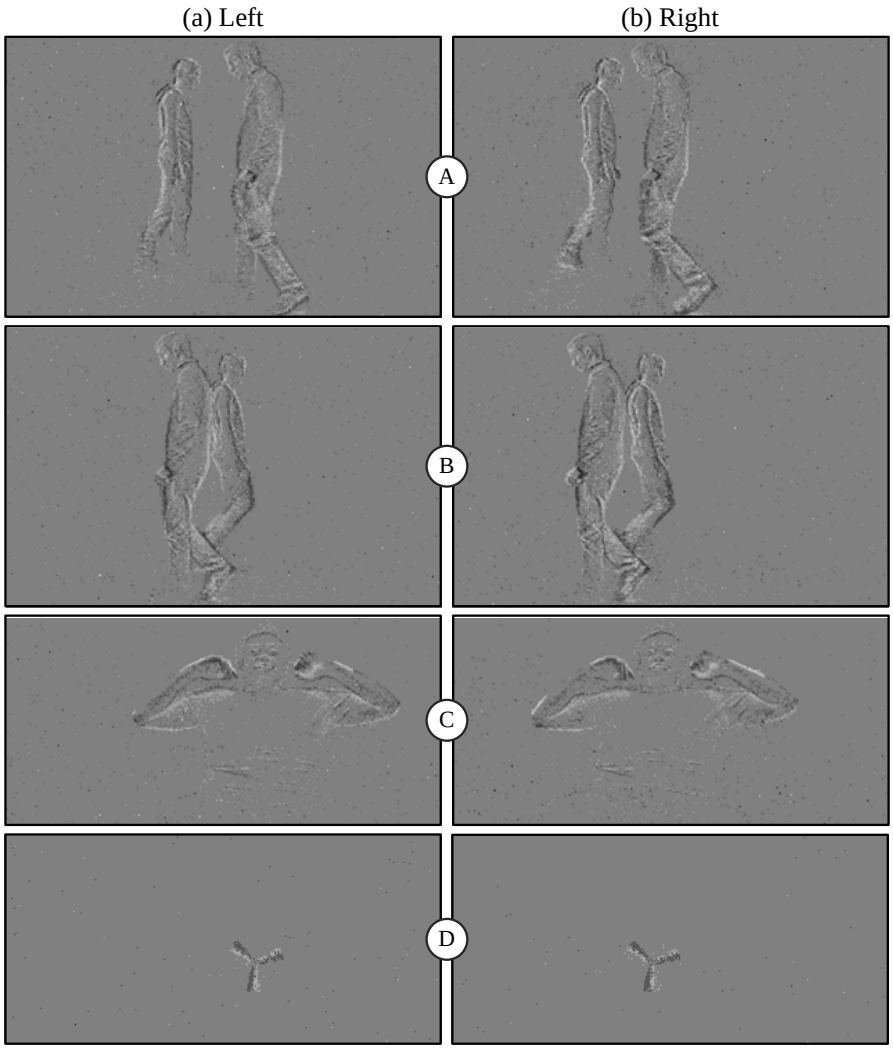


Figure 6.10: Grayscale event-images from all four test data sets A, B, C and D, generated from the collected events of the silicon retina sensor. The images shown represent the grayscale event-images with a time history of 600 (A-C) and 100 (D) timestamps. (Contrast-enhanced images for better visualization.)

The second ratio R_E is calculated by

$$R_E = \frac{Disparities\ Evaluated}{Disparities\ Calculated'} \tag{6.16}$$

which measures the relation between the evaluated disparity values and the disparity values delivered by the algorithm. This ratio measures the amount of values which are evaluated using the sparse ground truth and finally contribute to calculate the average distance error.

In Figure 6.11 the meaning of the ratios introduced in Equation (6.15) and (6.16) is illustrated. Image (a) shows all input events (on- and off-events) within a certain time

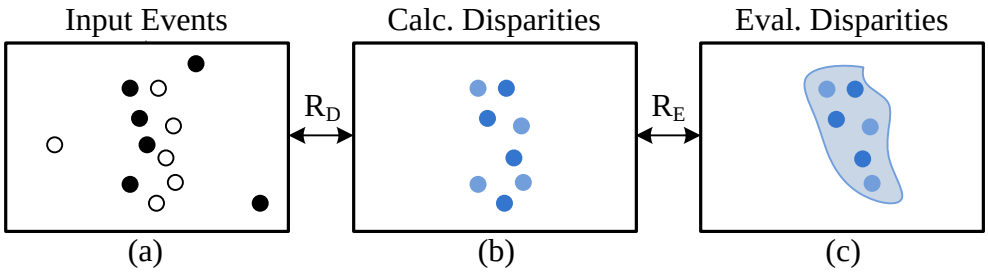


Figure 6.11: The three subsets for calculating the ratios R_D and R_E represented by images: (a) shows the input event event-image with the on- and off events, (b) the calculated disparities, and (c) the disparities which are evaluated.

and image (b) represents the disparity values calculated based on this input. These two values in relation lead to the ratio R_D. In image (c), the background color indicates the area for which ground truth values are available, and only disparities located inside this area are evaluated. The disparities inside this area divided by all calculated disparities provide the ratio R_E.

For the evaluation of the stereo matching algorithms, the R_D and R_E values in conjunction with the average distance error are important metrics for the interpretation of the results. A good stereo matching approach would not only achieve a low average distance error, but high R_D and R_E ratios as well.

6.3.1 Evaluation of Area-based Algorithms

Within TS2 we evaluate the area-based correlation algorithms described in Section 5.2.2.2. Before the tests of the different area-based stereo matching algorithms are conducted, the effect of filtering the input event-images generated from the silicon retina camera is evaluated.

6.3.1.1 Evaluation of Input Event-image Filtering

The input data is filtered before the stereo matching takes place, to remove noise and outliers. For this reason, we tested and compared a median filter and a connected component filter. Both filters are tested on grayscale event-images generated from test data sets A and D. After filtering, we employed the SAD and SSD approach with three different window sizes (3×3, 9×9, 15×15) and three different time histories (200, 400, 600) to calculate the results.

The first tested filter is the median filter, described in Section 5.2.2.1, with a kernel size of 3×3. Figure 6.12 shows the effect of the median filter when applied to test set A. The results are improved by the median filter because the filter removes values which

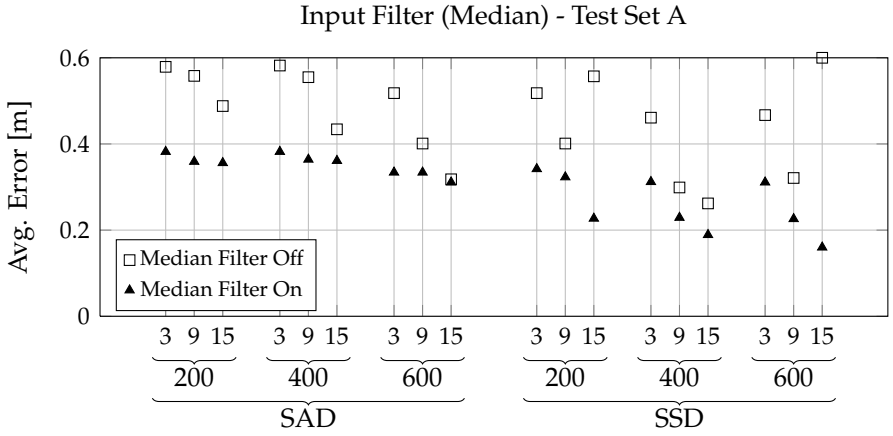


Figure 6.12: Evaluation of the median filter applied on the grayscale event-images generated from test data set A.

would contribute negatively to the results from test set A. This reduces the number of available and contributing matches, which decreases the ratio R_D as presented in Table 6.3.

The median filter was also applied on test set D, as shown in Figure 6.13. The median

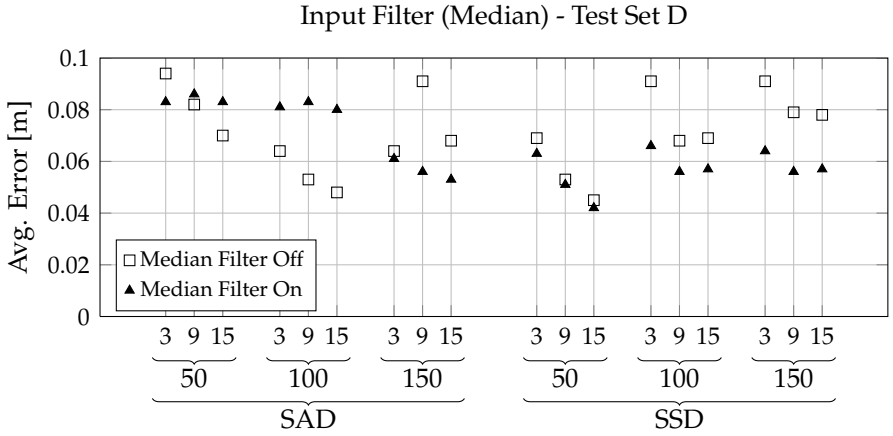


Figure 6.13: Evaluation of the median filter applied on the grayscale event-images generated from test data set D.

filter applied on test set D has a different result compared to Figure 6.12, because of the shorter time history less information was available for the generation of grayscale event-images. This means the median filter removes more grayscale values from the input images and lowers the amount of matching candidates.

Table 6.3 shows the event ratios R_D and R_E achieved with the median input filter. The overall results show that the usage of the median filter improves the quality of the

Median Filter	Time History	A R_D/R_E	Time History	D R_D/R_E
Off	h200	91.3/63.8	h50	86.3/51.6
On	h200	12.8/41.2	h50	18.3/76.2
Off	h400	84.0/61.0	h100	78.3/48.3
On	h400	29.9/76.0	h100	51.2/64.8
Off	h600	81.6/59.4	h150	68.8/41.1
On	h600	42.3/78.3	h150	45.8/55.5

Table 6.3: Average ratios R_D and R_E with and without applying a 3×3 median filter on the grayscale event-images of test data sets A and D.

calculated depth values but decreases the number of events evaluated. Therefore, we decided not to incorporate the median filter into our evaluation because the amount of removed input values is excessive.

Another filter applied to input grayscale event-images is the connected component filter introduced in Section 5.2.2.1. The filter is applied to each event of the grayscale event-image which is unequal to the background value and eliminates this event if no connected gray value unequal to the background value is present. In Figure 6.14 the results of applying the connected component filter to test set A are shown, and Figure 6.15 shows the results for test set D. If the time history is short, the number of

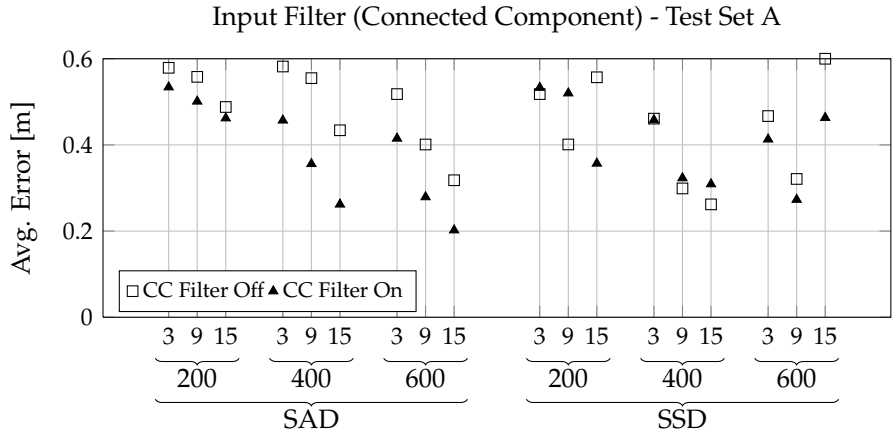


Figure 6.14: Evaluation of the connected component filter applied to grayscale event-images generated from test data set A.

events along edges is low and, consequently, edges appear sparse and become filtered out. In contrast to the median filter, such sparse data is only removed if no direct neighbor is present. This means the connected component filter deletes more safely input data, because, for example, a 3×3 median filter will remove the pixel even if there

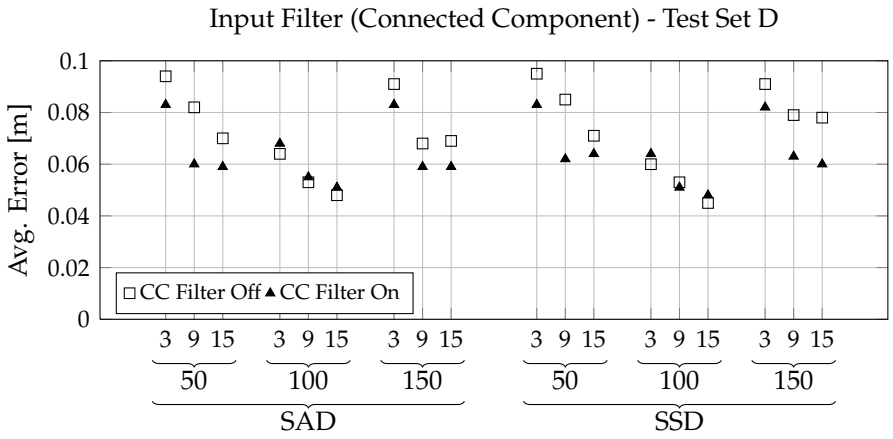


Figure 6.15: Evaluation of the connected component filter applied to grayscale event-images generated from test data set D.

are three direct neighbors. Therefore, the connected component filter achieves more stable filtering results over different time histories.

Table 6.4 shows the event ratios R_D and R_E using the connected component filter. The table shows the mentioned impact of the connected component filter in comparison

Conn. Comp. Filter	Time History	A R_D/R_E	Time History	D R_D/R_E
Off	h200	91.3/63.8	h50	86.3/51.6
On	h200	62.5/68.9	h50	62.1/56.8
Off	h400	84.0/61.0	h100	78.3/48.3
On	h400	67.2/65.8	h100	65.2/57.1
Off	h600	81.6/59.4	h150	68.8/41.1
On	h600	68.1/65.3	h150	56.2/49.0

Table 6.4: Average ratios R_D and R_E with and without application of the connected component filter.

to the median filter. Considering the ratio R_D more carefully, the noise pixels are removed, and the drop of the ratio R_D is lower than that of the median filter. Longer time histories increase the ratios R_D and R_E computed from the filtered results because the filter eliminates noise and additionally performs better than the median filter under the same test conditions.

From our tests, we conclude that the connected component filter is a good choice for the reduction of noise in our input event-images, and will therefore be used for the further tests.

6.3.1.2 Evaluation of Area-based Correlation Algorithms

In this section, we evaluate and compare various area-based correlation stereo matching approaches. We used the six correlation metrics SAD, ZSAD, LSAD, SSD, ZSSD and LSSD, which were introduced in Section 5.2.2.2. In Table 6.5, the different settings for the test of the correlation functions are summarized.

Correlation Functions	Time History for A,B,C	Time History for D	Window Sizes
SAD, ZSAD, LSAD SSD, ZSSD, LSSD	50, 200, 400, 600	20, 50, 100, 150	3, 9, 15

Table 6.5: Settings used for the evaluation of the area-based correlation functions.

Figure 6.16 shows the average distance errors for test set A. In general, the results

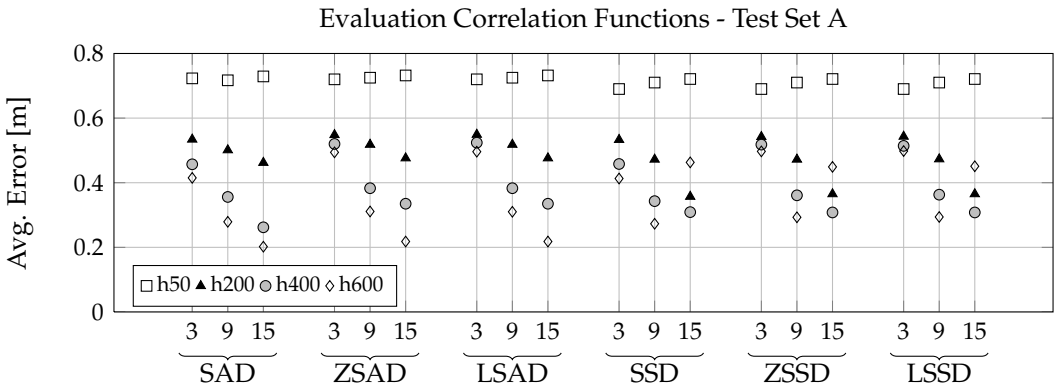


Figure 6.16: Evaluation of the different correlation functions using test data set A.

show that larger window sizes tend to achieve more accurate results, but have to be carefully chosen because of the foreground fattening effect which blurs the depth values at the objects edges. The results show also the expected increase of the accuracy with increasing time histories because of the higher number of events available for generating grayscale event-images. Considering the xSAD (SAD, ZSAD, LSAD) algorithms and the xSSD (SSD, ZSSD, LSSD) approaches, similar results are achieved. In Figure 6.17 the results are shown, where all correlation functions were evaluated with test set B. Here, the main difference in comparison to test set A is the better performance obtained from a time history of 50 timestamps. This can possibly be explained by the partial overlap of the walking persons and the amount of events in the overlapping area which are not affected by noise filter for the input event-images. This means despite the short history more events are available for matching.

The results of applying the different correlation methods on the human torso test set C are presented in Figure 6.18. The results show that the changing time history has

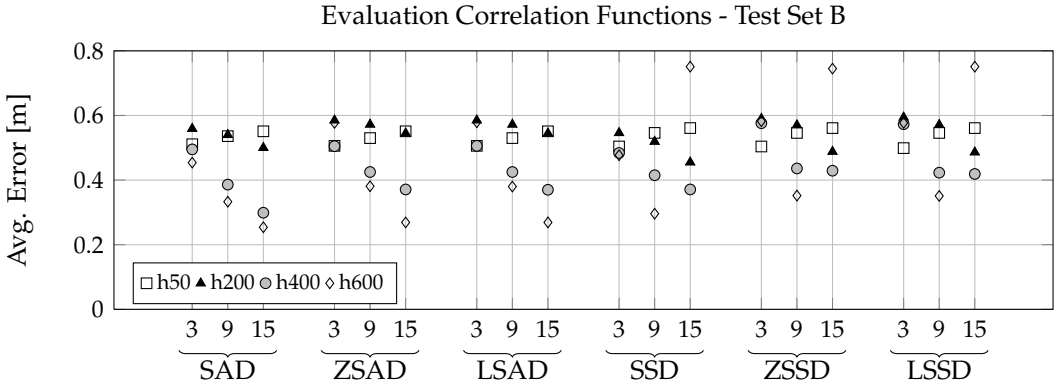


Figure 6.17: Evaluation of the different correlation functions using test data set B.

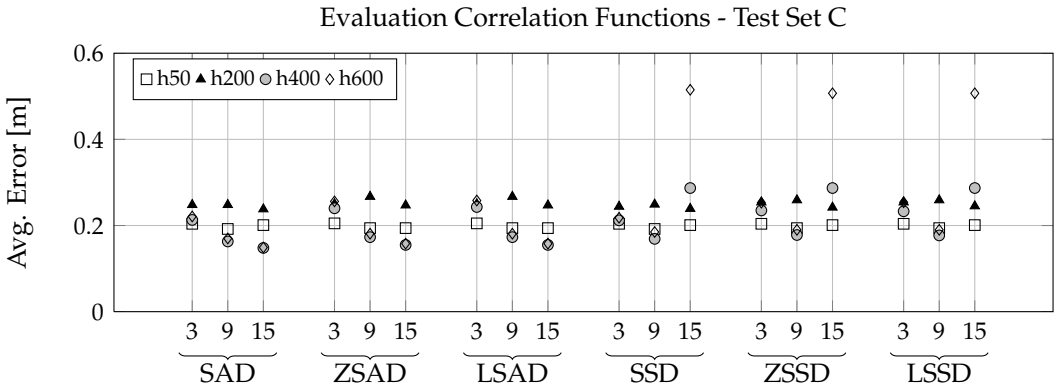


Figure 6.18: Evaluation of the different correlation functions using test data set C.

a rather small influence on the distance error. This can be explained by the fact that the human torso is moving at a relatively close distance of 1.5m, where short histories collect already enough events for good matching results.

The average distance error derived from testing the rotating disc (test set D) is shown in Figure 6.19. The results show that longer time histories with a 3×3 window show a higher error rate, which may be attributed to the fact that the higher number of active events causes some motion blur, which degrades the matching results for smaller window sizes. As the window size increases, the accuracy of the matching results also increases and the results tend to be stable with respect to different time histories.

In addition to the average distance error, the ratios R_D and R_E were calculated. In Table 6.6 the event ratios of the correlation function results are shown. The values show, as expected, that with short time histories, many events are filtered and the ratio R_D clearly drops in most cases. By contrast, the ratio R_E remains constant considering the different correlation functions.

In conclusion we can say that aside from the expected behavior with respect to

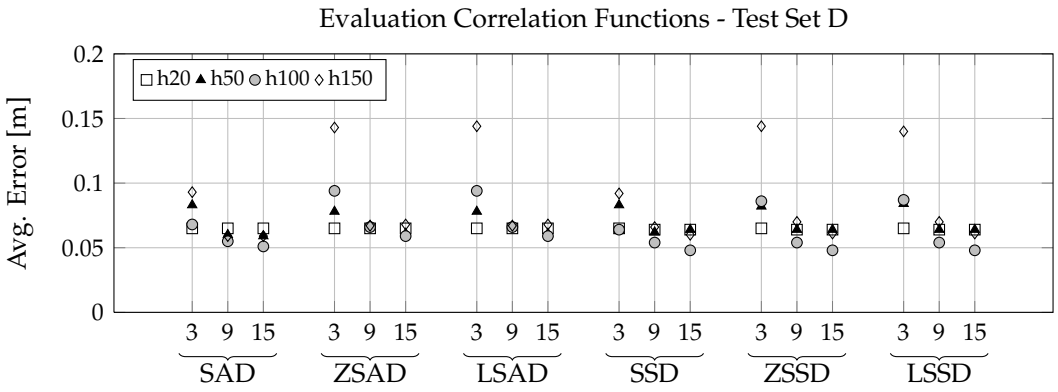


Figure 6.19: Evaluation of the different correlation functions using test data set D.

	SAD R_D/R_E	ZSAD R_D/R_E	LSAD R_D/R_E	SSD R_D/R_E	ZSSD R_D/R_E	LSSD R_D/R_E
A-h50	41.2/32.5	41.2/32.5	41.2/32.5	41.2/32.5	41.2/32.5	41.2/32.5
A-h200	61.9/74.8	61.6/75.8	61.6/75.8	63.1/63.0	63.1/63.5	63.0/63.5
A-h400	66.6/77.8	66.3/79.1	66.4/79.2	67.8/53.8	67.8/53.6	67.8/53.6
A-h600	67.7/79.7	67.2/83.1	67.2/83.0	68.5/51.0	68.5/50.9	68.4/50.9
B-h50	44.5/32.9	44.5/32.9	44.5/32.9	44.5/32.7	44.5/32.7	44.5/32.7
B-h200	67.4/79.9	67.4/79.6	67.4/79.6	67.9/65.3	68.1/65.2	67.9/65.6
B-h400	71.5/82.4	71.4/84.4	71.4/84.2	72.4/53.2	72.3/52.8	72.2/52.9
B-h600	73.0/78.4	72.5/84.1	72.5/84.0	73.8/49.4	73.7/49.1	73.7/49.4
C-h50	28.6/27.8	29.0/27.3	29.0/27.3	29.0/27.3	29.0/27.3	29.0/27.3
C-h200	67.1/75.7	67.1/76.4	67.1/76.4	67.1/73.5	67.1/73.7	67.1/73.7
C-h400	75.9/83.4	75.9/83.0	75.9/83.1	75.8/66.8	75.8/66.5	75.8/66.5
C-h600	76.7/85.1	76.7/85.7	76.7/85.6	76.5/64.8	76.5/65.2	76.5/65.2
D-h20	58.1/50.0	58.1/50.0	58.1/50.0	58.1/51.9	58.1/51.9	58.1/51.9
D-h50	62.1/57.2	62.1/55.4	62.1/55.4	62.1/56.5	62.1/56.8	62.1/56.5
D-h100	65.9/58.2	65.9/57.9	65.9/57.9	65.9/55.0	65.9/54.1	65.9/54.1
D-h150	56.7/50.1	56.7/52.4	56.7/52.4	56.7/47.2	56.7/49.2	56.7/49.1

Table 6.6: Average ratios R_D and R_E of all area-based correlation functions for all test data sets and time histories.

different window sizes and time histories, no major difference was observed among correlation methods. Since the correlation methods deliver nearly the same results, and the SAD is the fastest metric to calculate, we choose the SAD correlation method for the evaluation of the improvement techniques in Section 6.3.4.1.

6.3.1.3 Evaluation of Area-based Event Transform Algorithm

Another stereo matching approach we evaluate is the area-based event transform (ET) described in Section 5.2.2.2. In Table 6.7, all settings used for the evaluation of the event transform algorithm are summarized.

Input Noise Filter	Tri-State Dual-State	Time History for A,B,C	Time History for D	Window Sizes
NOn, NOff	TOn, DOn	50, 100, 200, 400, 600	20, 50, 100, 150, 200	3, 9, 15

Table 6.7: Settings used for the evaluation of the area-based event transform algorithm.

All four test sets are used for the event transform evaluation, where in Figure 6.20 the results of test data set A, in Figure 6.21 the results of test data set B, in Figure 6.22 the results of test data set C, and in Figure 6.23 the results of test data set D are presented.

Regarding different time histories and window sizes, the results show a behavior

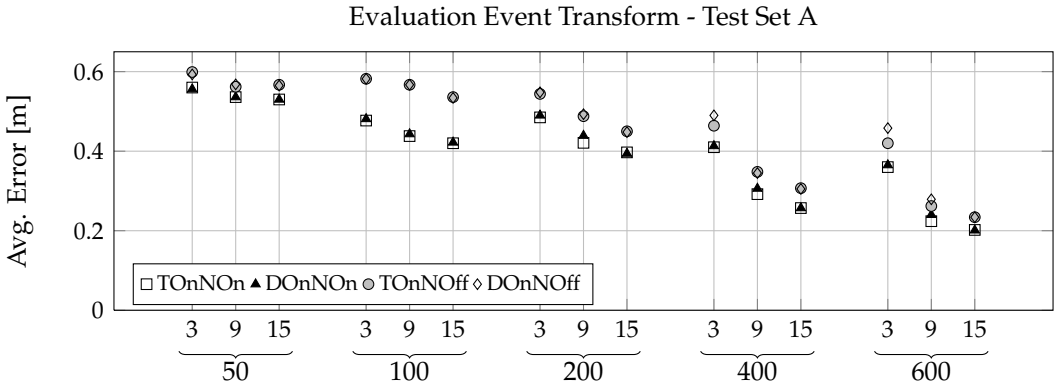


Figure 6.20: Evaluation of the event transform matching algorithm using test data set A.

similar to the area-based correlation functions. It is interesting that the results do not change significantly if the tri-state logic (TOn) is used instead of the dual-state logic (DOn). That means that the additional information extracted from the neighborhood using the tri-state logic does not provide additional insight to improve the matching results. In fact, the dual-state logic extracts enough information for the correspondence search by comparing the difference of bit vectors. The dual-state logic is the preferred setting for further tests because the calculation effort is less computationally expensive than tri-state logic. In terms of the input filter, the results show the expected behavior that the switched on input noise filter leads to better matching results.

Table 6.8 presents the event ratios of the event transform matching approach to analyze the average distance errors in more detail. Here, the usage of dual-state logic

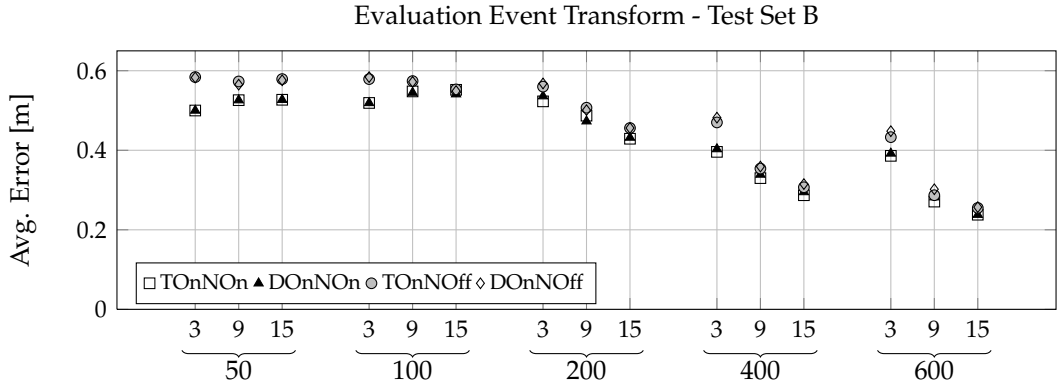


Figure 6.21: Evaluation of the event transform matching algorithm using test data set B.

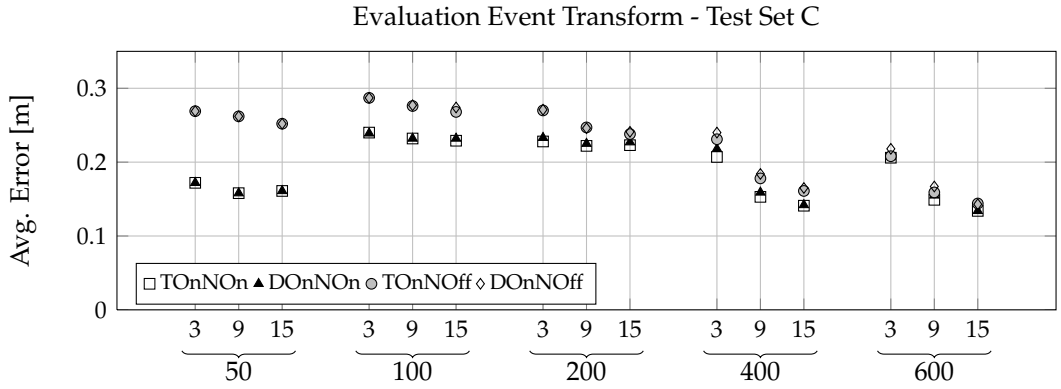


Figure 6.22: Evaluation of the event transform matching algorithm using test data set C.

and tri-sate logic respectively, makes no recognizable difference, and therefore, the event ratios are compared with activated and deactivated noise canceling. Comparing the event ratios of the event transform to the correlation functions (Table 6.6), shows that the ratio R_D is generally higher for all data sets and time histories, whereas the ratio R_E is almost identical in both cases. That means the event transform is performing better considering the matching in comparison to the before presented results of the different correlation methods.

6.3.2 Evaluation of Feature-based Corner Matching Algorithm

In this section we evaluate the feature-based corner feature matching approach (CF) which uses the feature detector from Shi and Tomasi [80], as described in Section 5.2.2.3. In Table 6.9 all parameters used during the evaluation of the feature-based corner

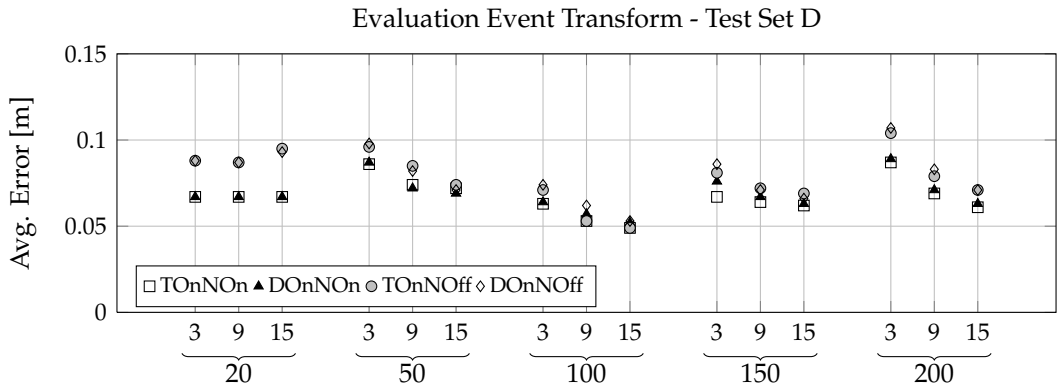


Figure 6.23: Evaluation of the event transform matching algorithm using test data set D.

	A R_D/R_E	B R_D/R_E	C R_D/R_E	D R_D/R_E
h20-NOn	—	—	—	27.4/76.5
h20-NOff	—	—	—	41.9/71.2
h50-NOn	10.2/88.3	12.7/90.7	8.1/74.5	61.4/60.3
h50-NOff	36.2/82.2	38.3/74.7	41.4/84.3	69.3/58.8
h100-NOn	34.3/83.6	42.5/80.3	26.4/84.7	68.2/58.5
h100-NOff	61.3/77.7	64.6/77.7	58.0/85.8	71.7/55.6
h150-NOn	—	—	—	70.0/54.2
h150-NOff	—	—	—	72.8/52.5
h200-NOn	63.3/83.0	68.8/85.4	58.1/88.3	72.3/45.6
h200-NOff	76.9/80.5	81.5/83.1	73.2/87.3	75.2/44.4
h400-NOn	77.8/84.1	81.7/85.7	72.4/88.5	—
h400-NOff	83.9/83.1	87.1/83.4	81.8/88.6	—
h600-NOn	81.2/87.1	84.8/87.1	77.2/89.6	—
h600-NOff	86.1/85.3	88.4/85.3	85.2/88.5	—

Table 6.8: Average ratios R_D and R_E of the event transform algorithm for all test data sets and time histories (lines represent that no results were calculated for the certain time history of the considered test set).

matching algorithm are presented.

In Figure 6.24 the results of the corner feature matching algorithm applied on all four test data sets are shown. The results show that the amount of allowed corners and the block size of the covariance matrix do not influence significantly the outcome of the algorithm. As expected, longer time histories deliver in general better results because of more information gathered in the grayscale event-images used by the feature extractor.

Maximal Corners	Block Size Covar. Matrix	y-Shift Tolerance	Time History for A,B,C	Time History for D
500, 1000	5, 9	1,3	200, 400, 600	50, 100, 150

Table 6.9: Settings used for the evaluation of the feature-based corner matching algorithm.

Regarding the influence of the y-shift allowed for finding corresponding matches, we found that a y-shift of three, which allows a difference of three pixels between left and right feature, does not always lead to superior results.

The ratios R_D and R_E are presented in Table 6.10. As expected, the ratio R_D is very

	A R_D/R_E	B R_D/R_E	C R_D/R_E		D R_D/R_E
h200-Y1	1.1/85.7	1.2/100.0	1.9/93.8	h50-Y1	37.5/100.0
h400-Y1	0.9/95.0	1.0/95.5	2.6/90.5	h100-Y1	45.2/100.0
h600-Y1	1.1/90.6	1.1/100.0	1.6/91.4	h150-Y1	66.3/98.4
h200-Y3	3.8/93.7	4.5/96.4	7.6/86.2	h50-Y3	37.5/100.0
h400-Y3	3.4/97.8	2.6/99.0	5.1/93.6	h100-Y3	45.2/100.0
h600-Y3	2.6/97.6	2.7/97.2	3.8/95.4	h150-Y3	66.3/98.4

Table 6.10: Average ratios R_D and R_E of the feature-based corner matching algorithm for all test data sets and time histories.

low because only a few features from all input events are matched. However, nearly all of the extracted features are evaluated, which explains the high ratio R_E. Regarding the y-shift, the ratio R_D confirms the assumption derived from the average distance errors that with a higher allowed y-shift more features are contributing to the results. But this contribution of values triggers a minor decrease in the accuracy of the average distance errors because added values with an inaccurate disparity are involved in the evaluation.

From the above experiments, we conclude that the feature-based corner matching algorithm is an alternative matching algorithm which can be used as a fast initial matching algorithm for key points before the matching of all available pixels takes place. For further testing, a y-shift of three pixels is suggested, because it provides the best results considering the amount of pixels matched and the average distance error.

6.3.3 Evaluation of Event-based Time Correlation Algorithm

This section evaluates the event-based time correlation algorithm (TC) using two different methods calculating the matching costs. In the case of time correlation, no window sizes are used because the algorithm is based on a line-wise search that considers each

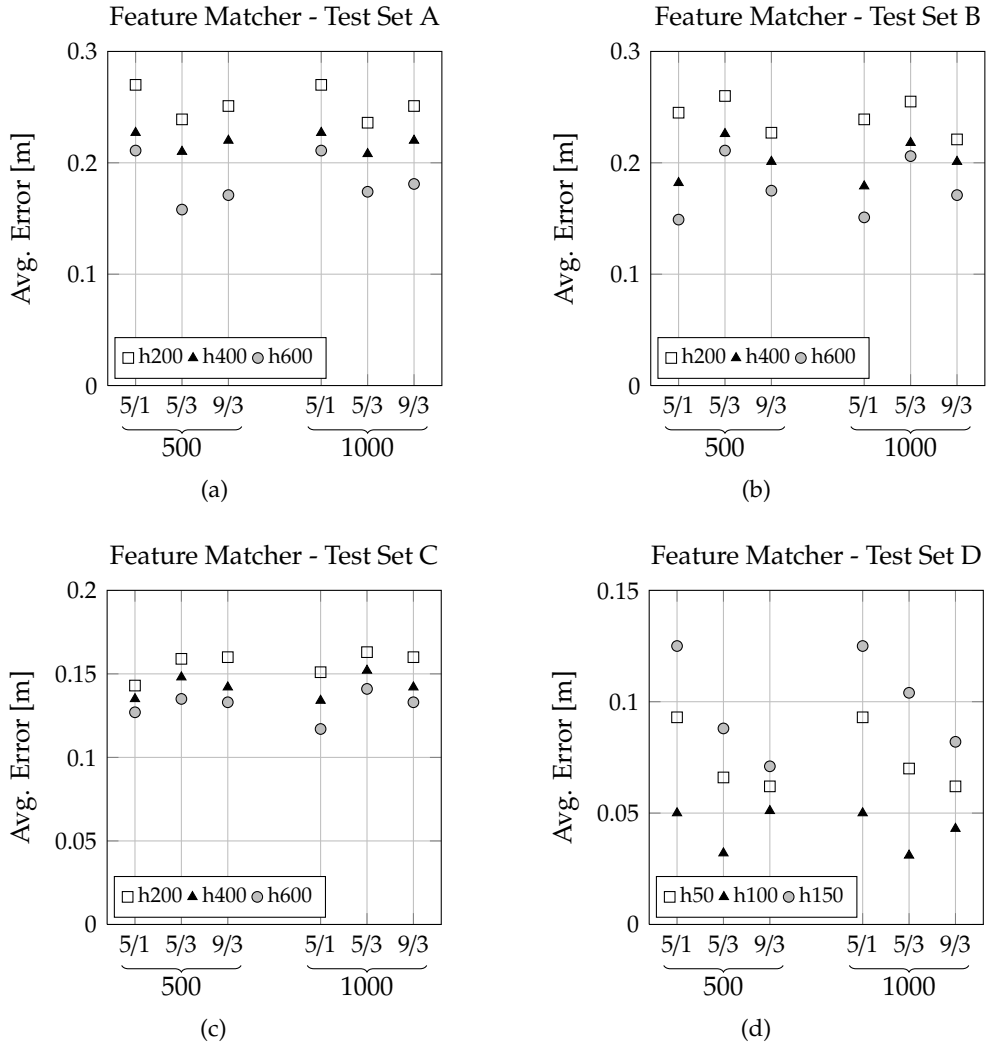


Figure 6.24: Evaluation of the corner feature matching algorithm using (a) test data set A, (b) test data set B, (c) test data set C and (d) test data set D. Here, 500 and 1000 on the x-axis represent the maximal corners used for the evaluation, and the value tuple above the number of maximum corners are the block size of the covariance matrix and the y-shift (block size/y-shift).

pixel individually. This also means that the input noise filter is switched off during the tests.

In Table 6.11, all settings used for the event-based time correlation algorithm are illustrated.

In Figure 6.25 the results of the event-based time correlation algorithm applied on all four test data sets are shown. The first observation which can be derived from the

Costs Calc. Method	Time History for A,B,C	Time History for D
m0 (inv. lin.), m1 (log.)	50, 100, 200, 300, 400, 500, 600	10, 20, 30, 40, 50, 100, 150, 200, 250, 300

Table 6.11: Settings used for the evaluation of the event-based time correlation algorithm.

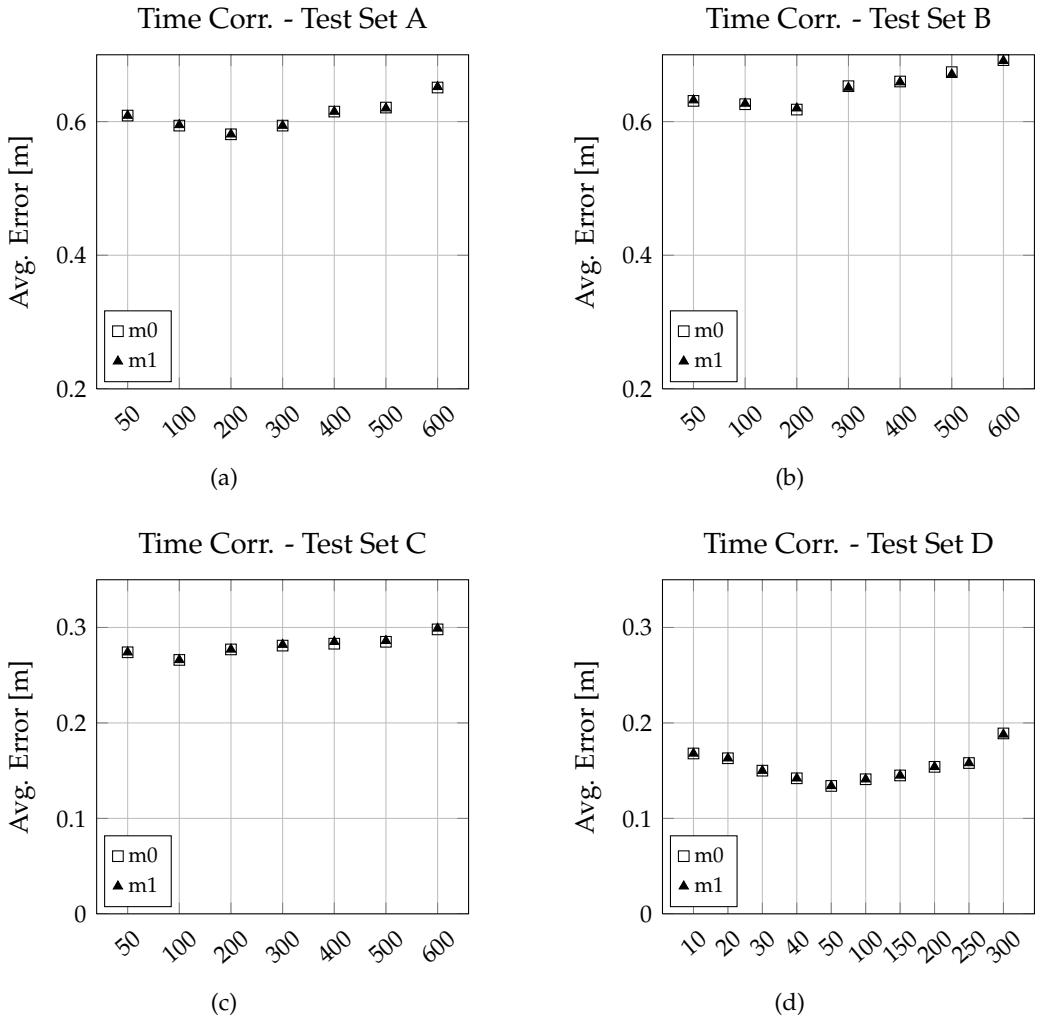


Figure 6.25: Evaluation of the time correlation algorithm with the two different cost calculation methods using (a) test data set A, (b) test data set B, (c) test data set C and (d) test data set D.

results are the nearly identical error values from the cost calculation method 0 (m0 - inverse linear) and method 1 (m1 - logarithmic). This means that the method does not influence the matching of the events, which leads to the decision of using the inverse linear method for a time-based matching algorithm, because it is computationally less expensive. In contrast, the time history has - as expected - a major influence on the distance errors, because the event-based time correlation approach matches in contrast to the evaluated algorithms in Figure 6.3.1.2 and Figure 6.3.1.3 only line-wise without considering the pixel neighborhood. This search is more sensitive to time histories because, firstly the history is needed as correlation metric and, secondly, too long time histories may lead to mismatches. Figure 6.25(d) illustrates this influence very clearly, where the best results are achieved with a time history of 50 timestamps.

Additionally to the average distance error, the event ratios have to be considered for a more detailed statement on the time history influence. Therefore, in Table 6.12 the ratios R_D and R_E calculated for all time histories are shown. The results show that

	A R_D/R_E	B R_D/R_E	C R_D/R_E	D R_D/R_E
h10	—	—	—	37.5/100.0
h20	—	—	—	45.2/100.0
h30	—	—	—	66.3/98.4
h40	—	—	—	68.0/98.8
h50	42.6/81.1	40.7/77.5	45.7/85.4	67.3/98.1
h100	65.3/79.3	66.5/78.2	62.5/86.6	69.9/97.1
h150	—	—	—	72.3/97.9
h200	78.9/79.9	82.9/79.3	73.3/86.3	73.4/98.4
h250	—	—	—	73.3/98.4
h300	84.0/78.3	86.6/79.3	81.3/85.7	74.1/97.5
h400	86.1/78.1	88.0/78.2	83.3/85.0	—
h500	86.7/78.0	88.3/78.0	84.2/85.2	—
h600	88.0/79.0	88.9/78.8	84.3/84.7	—

Table 6.12: Average ratios R_D and R_E of the event-based time correlation algorithm for all test data sets and time histories (lines represent that no results were calculated for the certain time history of the considered test set).

all test sets have in common an increasing ratio R_D in conjunction with longer time histories. Considering the ratio R_E all time histories generate a high ratio, which means concerning Figure 6.41 that a higher amount of matches with the wrong disparity are contributing to the average distance error.

Summarizing we have found that the event-based time correlation works reasonable due to the fact that the events are directly used as event list without converting event-images, and without using a neighborhood information for the matching. But the

algorithm is easily influenced by the time history which has to be set in accordance to the object speed in the scene. This renders the algorithm highly sensitive when operating on scenes with variable content.

6.3.4 Evaluation of Different Improvement Techniques

In this section we evaluate the improvement techniques presented in Section 5.3. Here, the goal is to reveal which improvement technique works best for the different stereo matching approaches tested in TS2.

In Table 6.13 all parameters used during the evaluation of the improvement techniques are presented. In all diagrams of Section 6.3.4.1, 6.3.4.2, 6.3.4.3 and 6.3.4.4 the

Avg. Filter	Med. Filter	BP Penalty	2SF Radius	BP - 2SF Iter.	Time Hist. for A,B,C	Time Hist. for D
3, 5	3, 5	10, 20, 40, 80, 160, 240	2, 4, 8, 12	2, 4	200, 400, 600	50, 100, 150

Table 6.13: Settings used for the evaluation of the improvement techniques.

results without improvement technique are represented by unfilled symbols and the results with an applied improvement technique are represented by filled symbols. In addition, the abbreviation *im* indicates in the legend of the diagrams that the symbols filled in black represent the improved results. The median filter and average filter we have combined under the category *diverse filters* (Div/Filt.). Next to the diverse filter category the *Belief Propagation* (BP) and *Two-Stage Filter* (2SF) improvement method results are presented on the x-axis of the diagrams.

6.3.4.1 Impact of Improvement Techniques Applied on Area-based Correlation Algorithm

In this section we evaluate the improvement techniques applied on an area-based correlation stereo matching approach, which we have introduced in Section 5.2.2.2. The representative correlation function chosen for this test was the SAD algorithm with a 9×9 window. In Figure 6.26, 6.27, 6.28 and 6.29, the impact of the improvement techniques applied on the SAD stereo matching algorithm using the test data sets A, B, C and D, respectively, is shown.

Table 6.14 shows the event ratios achieved by applying the tested improvement techniques to the area-based SAD stereo matching algorithm. All results show that the median filter (Med.) increases the accuracy compared to the SAD algorithm without the improvement technique, but at the same time reduces the amount of data in the disparity image, which is represented by the low event ratio R_D in the corresponding rows of Table 6.14. In contrast, the average filter (Avg.) has a minor or negative impact on the accuracy compared to the SAD stereo matching results, but it does not eliminate

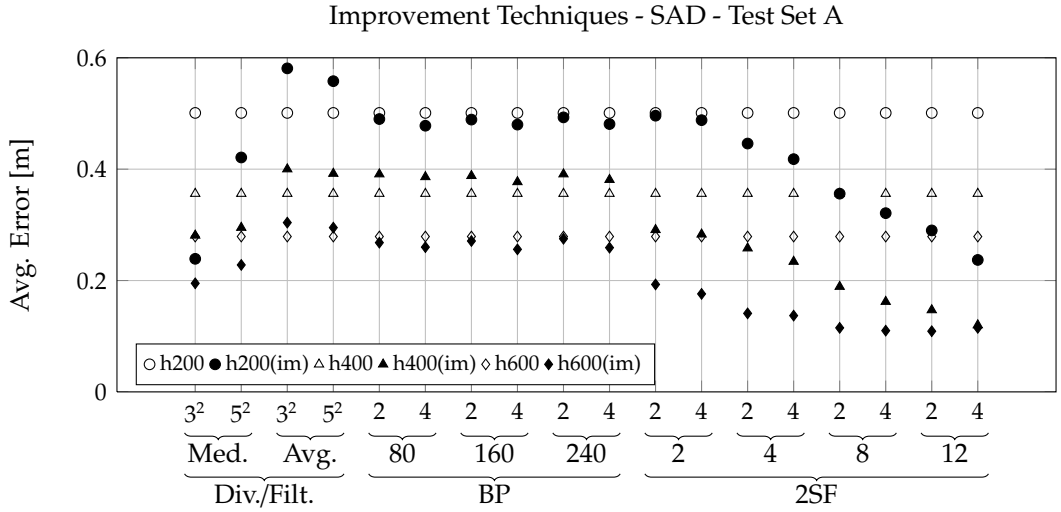


Figure 6.26: Evaluation of the impact of different improvement techniques applied to the area-based SAD algorithm using test data set A.

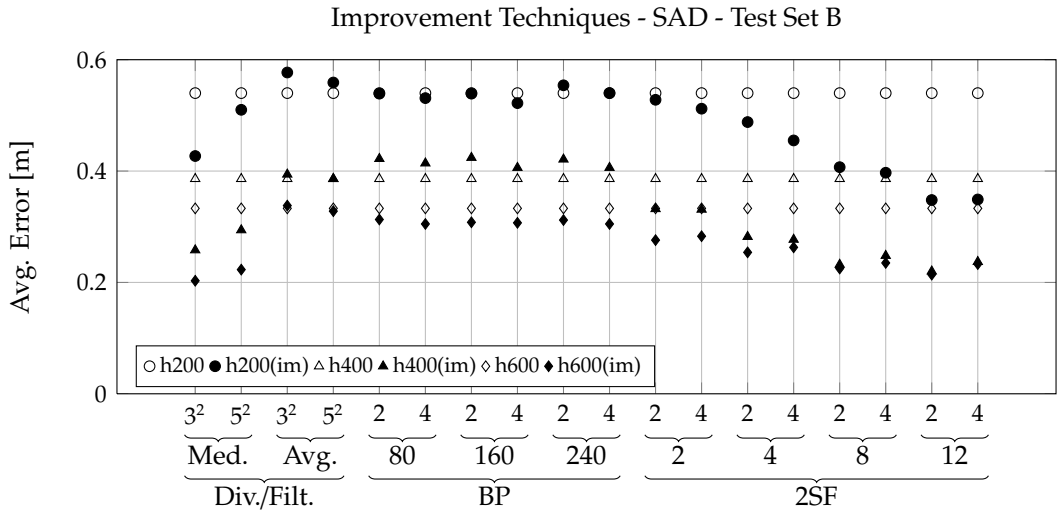


Figure 6.27: Evaluation of the impact of different improvement techniques applied to the area-based SAD algorithm using test data set B.

the same amount of disparity values as the median filter and, therefore, more values are involved in the evaluation process.

Regarding the belief propagation (BP) method, which mainly improves the SAD matching results minimal for all four test data sets in some cases (Figure 6.26 and Figure 6.27) also has a negative influence on the matching results. This may be explained by the dependency of the belief propagation on the neighborhood to perform the message

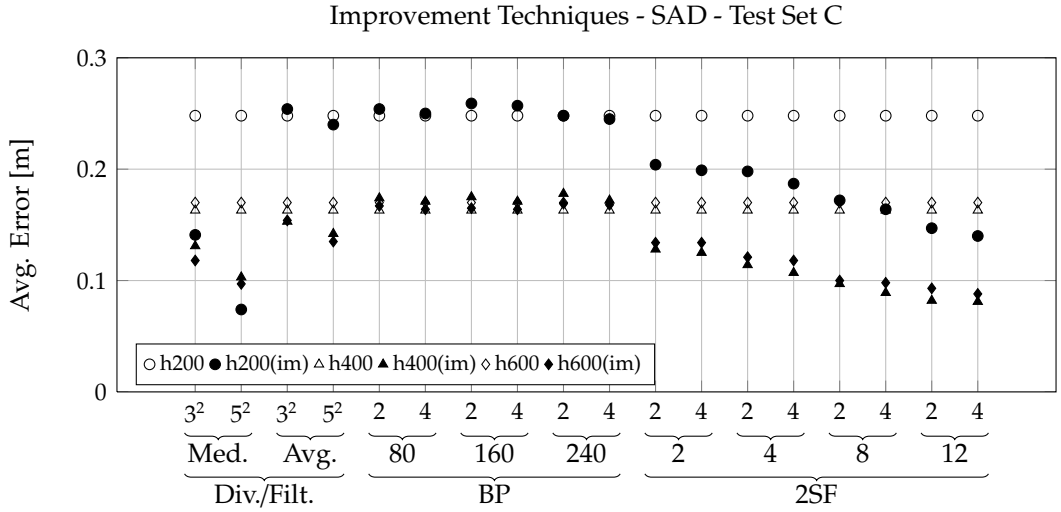


Figure 6.28: Evaluation of the impact of different improvement techniques applied to the area-based SAD algorithm using test data set C.

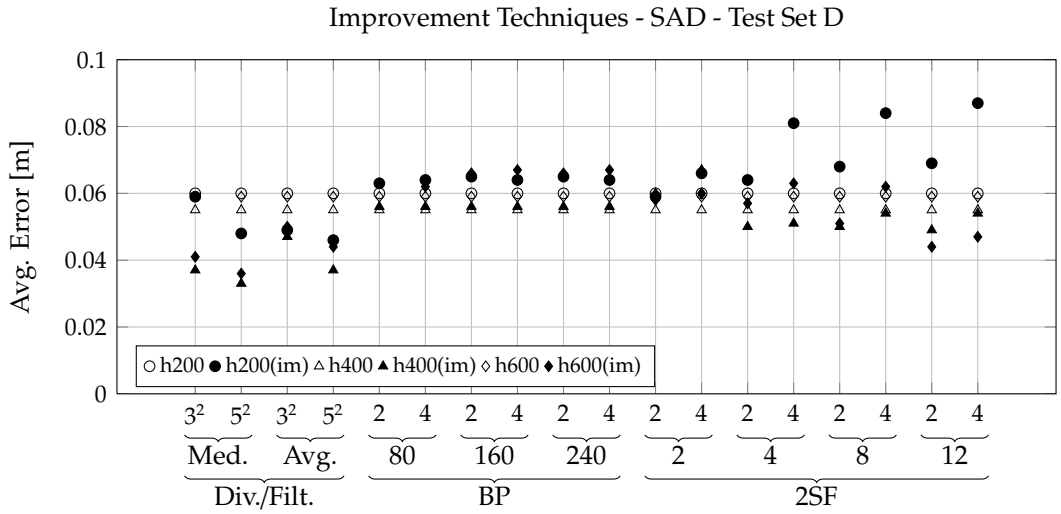


Figure 6.29: Evaluation of the impact of different improvement techniques applied to the area-based SAD algorithm using test data set D.

passing, as we have explained in Section 5.3.1. Further, the neighborhood for message passing is dependent on the chosen time history. Based on the chosen time history, the parameter *penalty costs* is set which again influences the outcome of the message passing. Regarding the event ratios, the belief propagation shows in all rows of Table 6.14 that the percentage of evaluated pixels does not change tremendously from the values of the SAD algorithm (Table 6.6). This means the BP does not change (manipulate) the

Improvement Method	Time History	A R_D/R_E	B R_D/R_E	C R_D/R_E	Time History	D R_D/R_E
BP	h200	61.4/82.7	66.6/85.7	66.4/86.3	h50	64.7/62.1
BP	h400	66.1/83.9	70.5/87.9	74.7/89.6	h100	64.9/55.7
BP	h600	67.2/89.2	71.9/92.4	76.2/89.3	h150	55.9/51.6
2SF	h200	60.8/91.2	67.4/92.3	67.1/87.6	h50	62.1/61.8
2SF	h400	66.0/92.5	70.8/94.2	75.8/89.0	h100	65.2/57.1
2SF	h600	67.0/93.6	72.2/93.7	76.8/90.6	h150	56.2/52.9
Avg. 3×3	h200	61.8/80.8	68.0/85.9	67.4/81.9	h50	62.1/62.1
Avg. 3×3	h400	66.4/87.1	71.4/89.8	76.0/87.3	h100	65.2/61.5
Avg. 3×3	h600	67.6/90.7	72.9/92.6	76.9/89.4	h150	56.2/51.1
Avg. 5×5	h200	62.4/84.2	68.0/87.6	67.3/84.5	h50	62.1/62.1
Avg. 5×5	h400	66.7/90.3	71.4/91.8	76.1/88.6	h100	65.2/61.0
Avg. 5×5	h600	67.5/93.5	72.5/93.5	76.9/90.6	h150	56.2/53.7
Med. 3×3	h200	19.5/68.4	22.2/76.0	15.9/86.1	h50	26.8/95.1
Med. 3×3	h400	41.8/72.3	50.9/80.5	44.2/91.6	h100	61.5/68.5
Med. 3×3	h600	53.5/83.6	64.5/89.0	76.9/89.4	h150	51.0/56.8
Med. 5×5	h200	4.4/76.8	4.1/77.4	3.0/96.2	h50	13.1/95.0
Med. 5×5	h400	21.2/68.7	33.3/71.3	28.4/95.0	h100	46.5/69.8
Med. 5×5	h600	39.3/73.2	55.2/83.0	41.0/90.5	h150	38.9/54.8

Table 6.14: Average ratios R_D and R_E obtained by using improvement techniques in conjunction with the area-based SAD algorithm for all test data sets and time histories.

amount of disparity such as the median filter does.

The results of the two-stage filter (2SF) show that the average distance error is improved by increasing the number of iterations from 2 to 4, and also by using a larger radius. With data set D (Figure 6.29), the 2SF encounters problems because the rotating disc at a time history of 50 does not provide enough disparity values for correct filtering. Looking at the event ratios, the 2SF technique achieves comparable ratios to belief propagation. Considering the good average distance error results achieved with the 2SF in conjunction with the event ratios the 2SF indicated to be a promising improvement technique.

6.3.4.2 Impact of Improvement Techniques Applied on Area-based Event Transform Algorithm

In this section, the evaluation of the improvement techniques applied to an area-based event transform stereo matching approach is carried out. For the evaluation of the event transform dual-state logic and a 9×9 window was used. In Figure 6.30, 6.31, 6.32 and 6.33 the impact of the improvement techniques applied to the event transform stereo matching results and the test data sets A, B, C and D is shown.

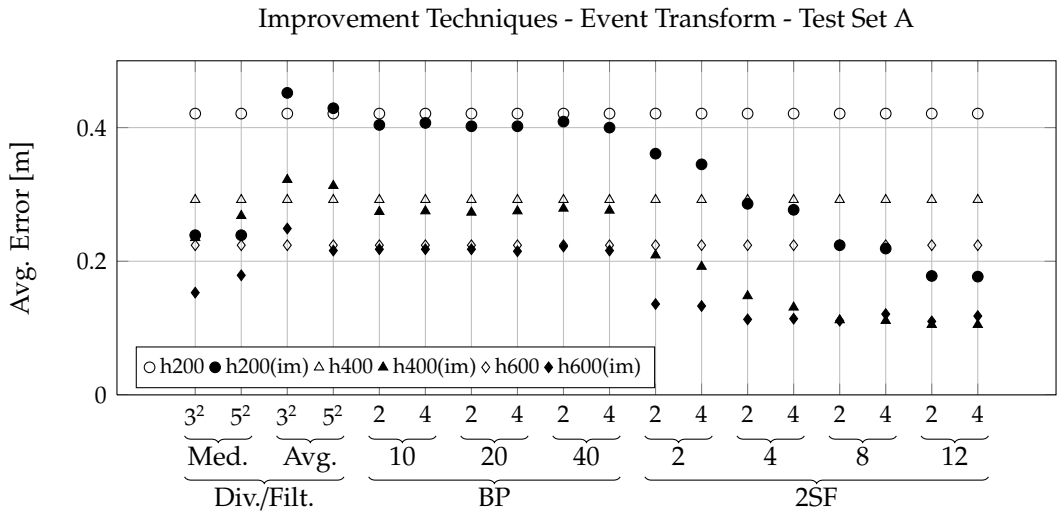


Figure 6.30: Evaluation of the impact of different improvement techniques applied to the area-based event transform algorithm and test data set A.

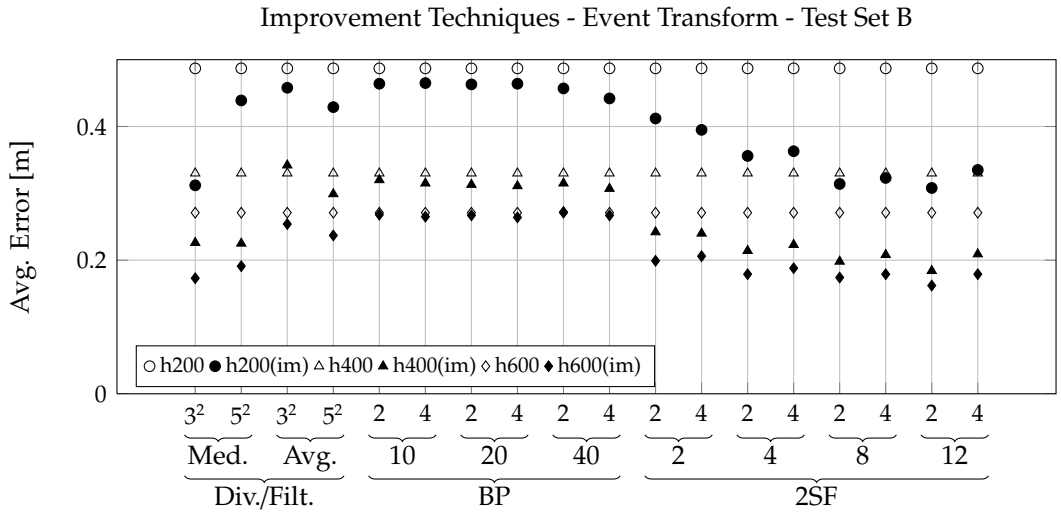


Figure 6.31: Evaluation of the impact of different improvement techniques applied to the area-based event transform algorithm and test data set B.

Table 6.15 shows the event ratios achieved by applying the investigated improvement techniques on the area-based event transform stereo matching algorithm described in Section 5.2.2.2. Comparing the results of the median filter applied to the event transform algorithm (Figure 6.26-6.29) and the SAD algorithm (Figure 6.30-6.33), the median filter performs in a similar way such that increasing time history decreases the average distance error. Additionally, with longer time histories the event ratio R_D is increasing

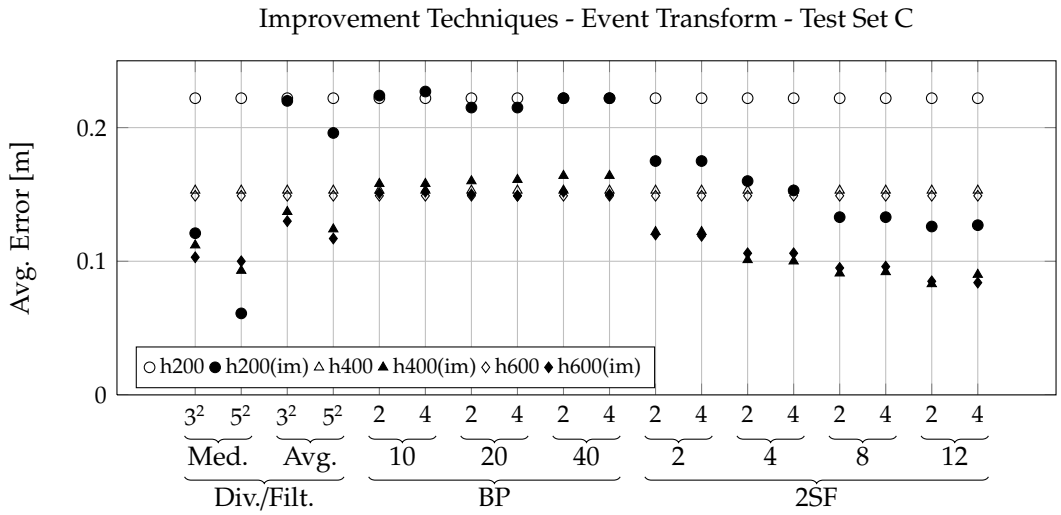


Figure 6.32: Evaluation of the impact of different improvement techniques applied to the area-based event transform algorithm and test data set C.

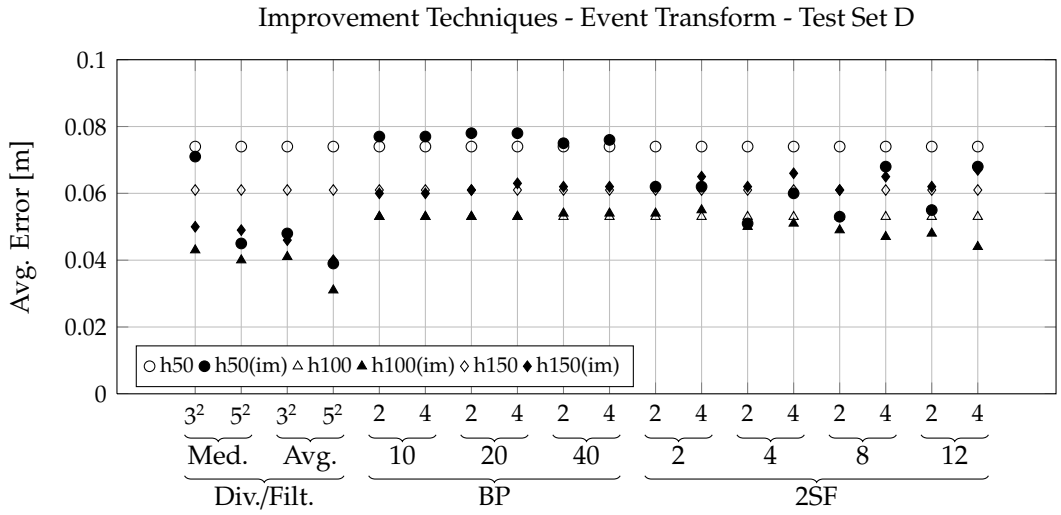


Figure 6.33: Evaluation of the impact of different improvement techniques applied to the area-based event transform algorithm and test data set D.

which is shown in all rows of Table 6.15 and makes the results of the median filter more valid, because more matches are considered for the calculation of the average distance error. Similar to the median filter the average filter performs better with longer time histories, but considering the ratio R_D in Table 6.15 also with short time histories a high percentage of matches used for the calculation of the average distance error is achieved.

Using the BP improvement technique with the event transform algorithm shows

Improvement Method	Time History	A R_D/R_E	B R_D/R_E	C R_D/R_E	Time History	D R_D/R_E
BP	h200	74.7/81.8	76.3/86.5	71.4/87.9	h50	66.7/63.2
BP	h400	79.8/88.1	83.2/88.3	80.3/88.6	h100	72.2/56.9
BP	h600	82.5/90.1	85.7/89.5	81.6/89.2	h150	71.5/53.2
2SF	h200	62.9/93.2	68.2/93.6	58.1/92.7	h50	61.4/67.4
2SF	h400	77.6/94.3	81.0/93.7	72.4/90.5	h100	68.2/58.3
2SF	h600	80.6/94.6	84.2/94.4	77.6/91.4	h150	70.0/54.8
Avg. 3×3	h200	63.5/87.6	69.1/90.4	58.2/92.4	h50	61.4/67.0
Avg. 3×3	h400	78.3/91.5	81.5/92.4	72.2/90.9	h100	68.2/59.8
Avg. 3×3	h600	81.5/94.1	84.8/92.3	59.4/91.3	h150	70.0/54.1
Avg. 5×5	h200	63.9/91.4	68.8/92.0	81.9/93.7	h50	61.4/71.3
Avg. 5×5	h400	78.5/94.6	81.9/93.7	72.3/91.9	h100	68.2/62.7
Avg. 5×5	h600	81.5/95.0	84.5/93.8	77.4/92.3	h150	70.0/54.8
Med. 3×3	h200	22.7/74.6	25.9/78.5	15.0/87.6	h50	24.2/86.5
Med. 3×3	h400	59.0/77.0	67.7/83.4	45.7/91.6	h100	64.9/66.0
Med. 3×3	h600	75.1/87.5	83.0/88.3	77.1/91.3	h150	67.6/62.3
Med. 5×5	h200	8.0/76.5	7.1/76.1	3.1/92.6	h50	9.2/92.9
Med. 5×5	h400	41.3/68.2	51.1/75.4	30.2/95.7	h100	51.5/71.4
Med. 5×5	h600	64.3/83.1	77.4/85.8	47.9/92.9	h150	60.6/63.7

Table 6.15: Average ratios R_D and R_E of using improvement techniques in conjunction with area-based event transform stereo matching algorithm for all test data sets and time histories.

in Figure 6.30-6.33 that the results are improved better than using BP with the SAD algorithm. This variance can be explained by the mentioned dependency between time history, penalty costs, and message passing, which lends the BP algorithm its behavior. The event transform uses the difference between the left and right vectors for the cost calculation and the penalty costs are derived from the maximal difference a vector can have. This means the cost structure is more balanced considering the vector difference and penalty costs, which enables a better operation of the BP with the event transform.

In contrast, the two-stage filter performs for all time histories and with all filter radii in a way that the overall average distance error is significantly decreased (see Figure 6.30- 6.33). The 2SF experiences added difficulties if objects of different depth levels are overlapping, as in in test set B (Figure 6.31), because then the 2SF technique - especially with larger radii - can degrade the results.

6.3.4.3 Impact of the 2SF Improvement Technique Applied on Feature-based Corner Matching Algorithm

In this section we evaluate the 2SF improvement technique applied on the results of the feature-based corner matching algorithm, introduced in Section 5.2.2.3. For the test a maximum corner count of 500, a minimum corner feature distance of 2 pixels, a block size of 5 and a maximum y-coordinate shift of 3 was chosen. In Figure 6.34(a), 6.34(b), 6.34(c) and 6.34(d) the impact of the 2SF improvement technique applied on the feature-based corner matching results of the test data set A, B, C and D is shown.

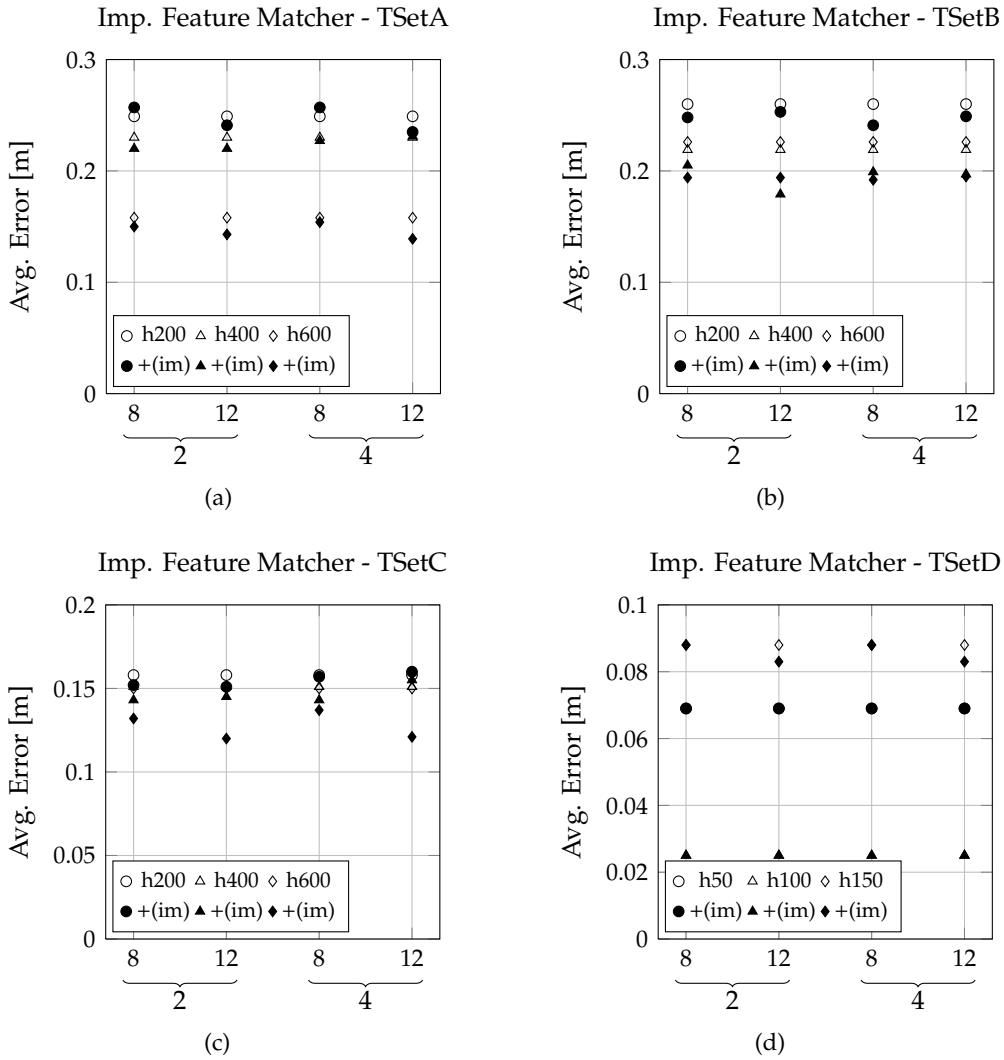


Figure 6.34: Evaluation of the impact of the 2SF improvement technique applied to the results of the feature-based corner matcher of all test data sets.

Table 6.16 shows the event ratios of the feature-based corner stereo matching algorithm. In general can be observed for all four test sets (Figure 6.34a-6.34d) that the 2SF

Improvement Method	Time History	A R_D/R_E	B R_D/R_E	C R_D/R_E	Time History	D R_D/R_E
2SF	h200	3.1/87.5	3.2/95.1	7.7/87.4	h50	3.9/66.7
2SF	h400	3.0/97.0	2.7/97.7	5.5/89.4	h100	3.0/22.2
2SF	h600	2.4/97.2	2.5/94.6	4.5/92.4	h150	2.2/44.4

Table 6.16: Average ratios R_D and R_E of the feature-based corner matching algorithm for all test data sets and time histories.

improves with a larger radius of 12 the average distance error, which can be explained with the large distance between the matched features. The 2SF method as additional improvement of the corner matching results is good until a certain point, because the radius has to be increased but should not connect features of intensely different depth. Using the radius of 12 can be considered as the maximum useful radius for test data sets processed.

Considering the ratios R_D and R_E in Table 6.10 the usage of the 2SF improvement technique does not change the ratios much from the values presented in Table 6.16. From all different 2SF settings considering radii and iterations the average of the event ratios was calculated, because mainly the difference of the ratios is caused by the time history.

Concluding, the feature-based corner matching algorithm generates a sparse depth output and only the 2SF is a useful improvement technique for this algorithm, which has not the significant impact on the results of the feature-based corner matching algorithm. This means that for the feature-based corner matching, it is not necessary to apply an improvement technique and spend valuable processing resources to achieve minor improvements.

6.3.4.4 Impact of Improvement Techniques Applied on Event-based Time Correlation Algorithm

In this section we evaluate the improvement techniques applied on the event-based time correlation stereo matching approach, which was described in Section 5.2.3. For event-based time correlation correspondence search the inverse linear cost calculation method was chosen. In Figure 6.35, 6.36, 6.37 and 6.38 the impact of the improvement techniques applied on the time-based correlation matching results of test data set A, B, C and D is shown.

In Table 6.17 the event ratios achieved applying improvement techniques to the event-based time correlation algorithm are shown. In case of the event-based time correlation algorithm no noise canceling is active for the input data and therefore, the performance of the median and average filter is moderate for test case A, B and C

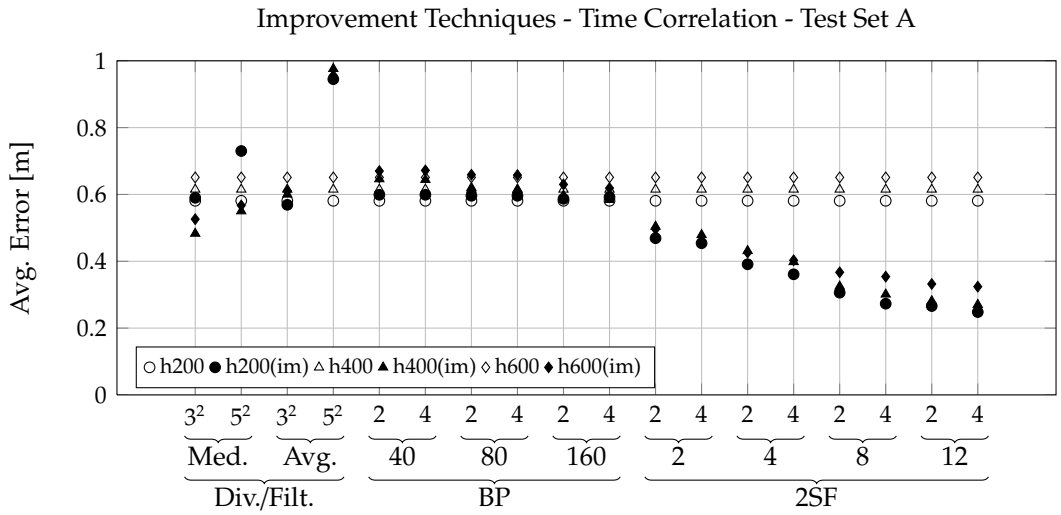


Figure 6.35: Evaluation of the impact of different improvement techniques applied to the event-based time correlation algorithm and test data set A.

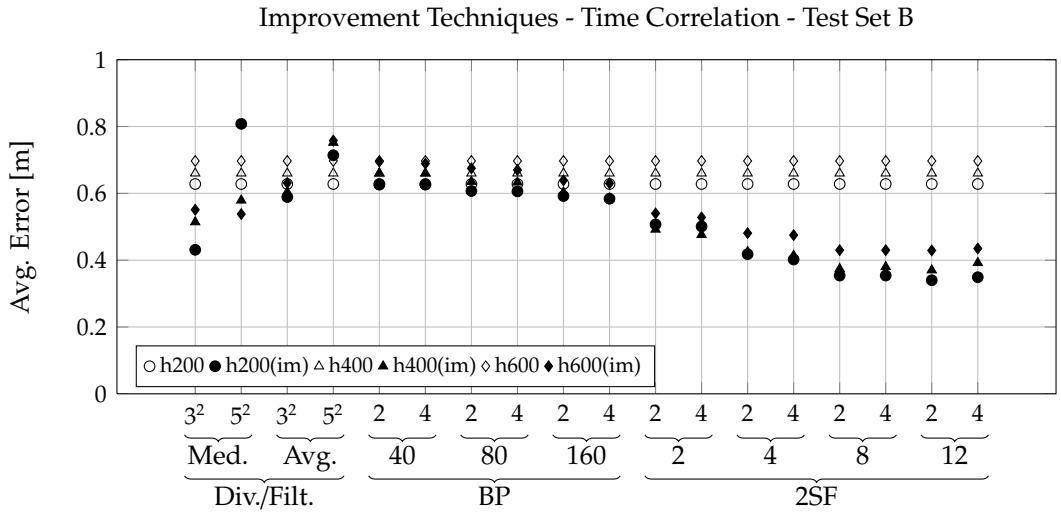


Figure 6.36: Evaluation of the impact of different improvement techniques applied to the event-based time correlation algorithm and test data set B.

(Figure 6.35-6.37). For test set D in Figure 6.38 the median filter and average filter achieve good results, which can be explained by the fact that the rotating disc is plane and results in many pixels with a similar disparity level used by the filters.

The usage of the BP improvement method has again for the test sets A, B and C only minor improvements, which can be explained with the algorithm's line-wise matching. This means the matching process does match different disparities for pixels close to each

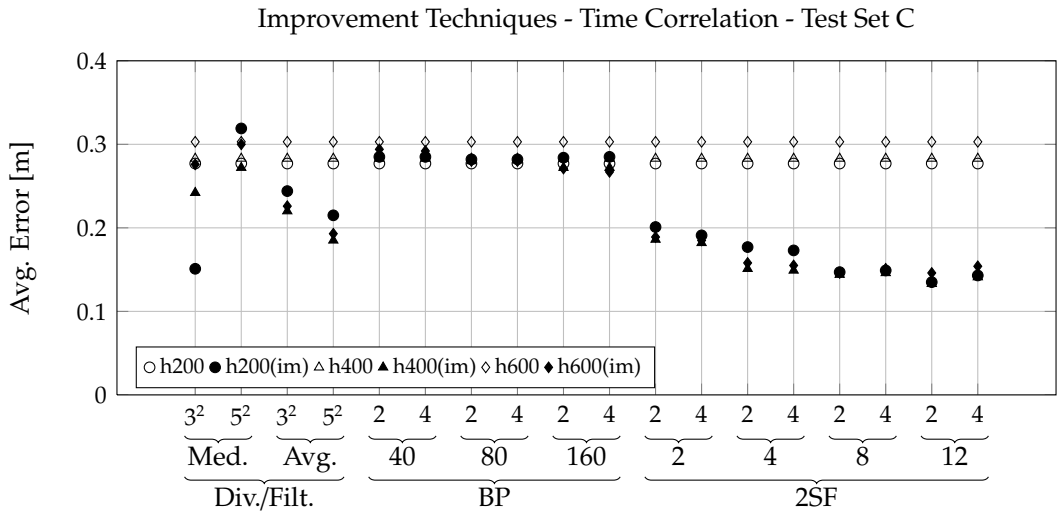


Figure 6.37: Evaluation of the impact of different improvement techniques applied to the event-based time correlation algorithm and test data set C.

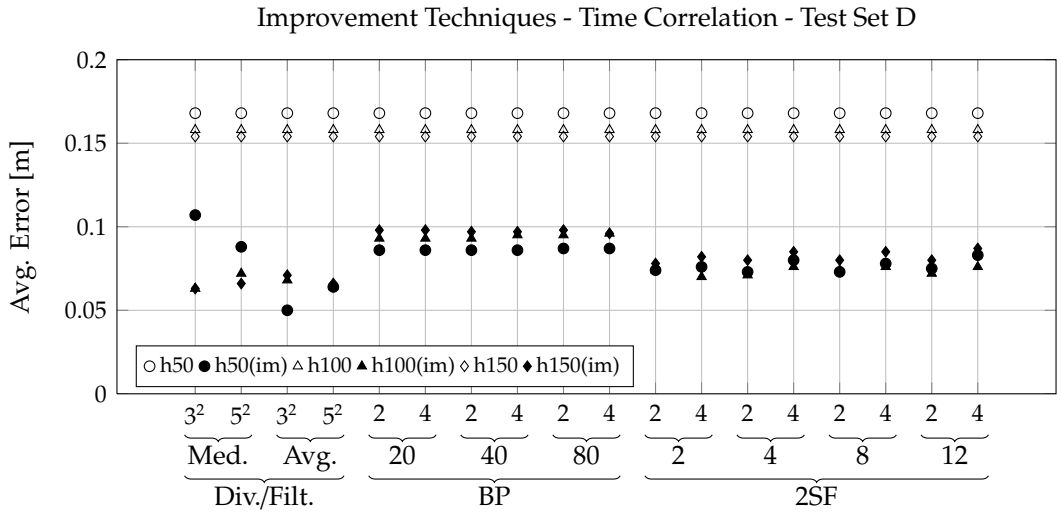


Figure 6.38: Evaluation of the impact of different improvement techniques applied to the event-based time correlation algorithm and test data set D.

other, which makes the matching passing process of the BP difficult. An exception is test set D, where the BP benefits from the plane rotating disc as already mentioned for the average filter and median filter. The BP technique improves the average distance in a level to be competitive with the 2SF shown in Figure 6.38.

Using the 2SF improvement approach shows for all four test sets (Figure 6.35-6.38) that the filter very constantly improves the distance values for all time histories used.

Improvement Method	Time History	A R_D/R_E	B R_D/R_E	C R_D/R_E	Time History	D R_D/R_E
BP	h200	94.7/76.0	95.1/76.5	91.9/83.2	h50	80.4/55.0
BP	h400	92.8/76.9	93.9/77.0	91.1/82.9	h100	84.6/47.5
BP	h600	94.0/77.5	94.2/78.2	90.1/83.9	h150	83.2/44.3
2SF	h200	78.7/92.3	82.5/91.7	73.2/91.6	h50	67.3/55.7
2SF	h400	85.7/92.7	86.9/92.4	83.4/90.5	h100	69.9/56.5
2SF	h600	87.5/92.3	87.9/92.5	84.7/90.2	h150	72.3/49.6
Avg. 3×3	h200	79.2/86.1	83.3/86.0	73.1/89.4	h50	67.3/63.1
Avg. 3×3	h400	86.1/87.9	87.9/89.1	82.9/89.7	h100	69.9/56.5
Avg. 3×3	h600	88.2/89.2	88.8/90.4	84.2/88.4	h150	72.3/50.7
Avg. 5×5	h200	66.0/61.3	75.4/52.9	73.4/90.2	h50	67.3/67.0
Avg. 5×5	h400	74.8/67.7	36.7/45.9	49.3/93.3	h100	69.9/56.5
Avg. 5×5	h600	24.0/59.4	82.7/54.6	84.4/89.9	h150	72.3/49.0
Med. 3×3	h200	26.8/62.4	28.7/63.9	13.5/81.9	h50	29.4/77.8
Med. 3×3	h400	62.2/65.3	70.8/70.4	48.2/82.7	h100	64.2/63.5
Med. 3×3	h600	76.6/74.8	84.8/75.3	60.8/86.4	h150	67.6/57.5
Med. 5×5	h200	13.0/57.2	7.7/52.5	2.7/78.3	h50	11.8/88.9
Med. 5×5	h400	44.6/55.5	55.1/60.7	32.1/78.8	h100	50.8/62.5
Med. 5×5	h600	67.5/66.5	80.4/67.9	49.8/83.1	h150	61.6/54.2

Table 6.17: Average ratios R_D and R_E of using improvement techniques in conjunction with event-based time correlation stereo matching algorithms for all test data sets and time histories.

The mismatches of the event-based time correlation algorithm are reliably removed from the 2SF because of its design (see Section 5.2.3) the two stages.

Considering the event ratios the event-based time correlation matching algorithm has more disparities because no input filter (noise removing see Section 6.3.1.1) is applied, which leads to high R_D ratios in Table 6.17. Not all of the matched events can be evaluated, which lowers the R_E ratios to a level comparable to the area-based correlation approaches (see Table 6.6). The shown performance of the 2SF makes this method to the preferred improvement technique using the event-based time correlation matching algorithm.

6.3.4.5 Depth Maps, Error Images and Processing Time of the Algorithms

Figure 6.39, 6.40, 6.41 and 6.42 show the depth maps of the different algorithms with and without the applied improvement techniques. The four columns represent the four test data sets and the rows differentiate the algorithm and improvement technique used. These depth images visualize some of the numeric results afore presented in the

certain sections. The disparity maps are color coded for a better visualization, where close distances appear in blue and far objects are represented by the color red.

The visualization in Figure 6.43 gives more insight into the location of the errors. The error map is color coded in order to highlight areas in the image which are more difficult to match. Columns represent test data sets A-D and the rows represent the different improvement techniques used. In the first row the result from the SAD algorithm without an applied improvement technique and with a 9×9 window and a time history of 600 timestamps is shown. Row 2-5 shows the results of the improvement techniques median filter, average filter, BP and 2SF, applied on the SAD results shown in the first row. Using only the SAD stereo matching shows mismatches and errors of up to 20% distributed over all image areas. Applying the median filter produces decent results, but reducing the number of depth values and the average filter does not reduce the number of depth values but rather decreases the quality. Both improvement techniques do not achieve the preferred outcome. In investigating the error maps of the BP improvement technique, the error is reduced in comparison to the SAD algorithm, but this is at the expense of any reduction in the number of depth values. With the 2SF method, the performance is acceptable and the error is reduced in most cases. If the number of mismatches within a certain neighborhood increases, then the 2SF method is not able to maintain improvement in the results, and can possibly generate worse results in comparison to the SAD (illustrated in Figure 6.43 column B row 5). This can be explained by the different points of view of the left and right camera (see Figure 6.10 row B), because the algorithm's correspondence search generates mismatches with high error values.

To better understand the computational effort of the stereo matching algorithms and the improvement techniques, the processing time was measured with different parameter settings, as presented in Table 6.18. All algorithms were implemented in C++ without optimization and executed on an Intel® Core™2 Quad processor running at 2.83GHz. This table shows, as expected, that the SAD algorithm without improvement techniques operates more rapidly depending on window size and time history. The computational effort of the SAD algorithm in generating the disparity map is between 1ms and 49ms. In Table 6.18 the number of events processed is written in parentheses next to the test set description. Different test sets and time periods (200(50) or 600(150)) change the number of events which have to be processed and, thus, directly influence the processing time of the algorithms. For the SAD algorithm the BP and 2SF method were evaluated and for the TC, ET and CF algorithm only the 2SF improvement technique were considered. The usage of belief propagation leads in average to a 10 times longer processing time in comparison to the SAD algorithm. The 2SF method only increases the processing time by a factor of 2-4. These measurements are not absolute time values because with optimizations better processing times can be achieved, but it gives us a better understanding on complexity and the potential of the different approaches.

	A-200 (2549) (ms)	A-600 (5649) (ms)	B-200 (2360) (ms)	B-600 (5347) (ms)	C-200 (1332) (ms)	C-600 (3303) (ms)	D-50 (197) (ms)	D-150 (524) (ms)
SAD 3×3	2	5	2	5	1	4	1	2
SAD 9×9	6	21	5	19	3	15	1	2
SAD 15×15	13	49	11	45	6	37	2	4
SAD 9×9 +BP Iter. 2	56	154	51	130	53	184	25	42
SAD 9×9 +BP Iter. 4	107	301	97	251	105	380	49	81
SAD 9×9 +2SF I2/R4	11	35	10	31	6	24	2	4
SAD 9×9 +2SF I4/R8	19	54	16	46	9	35	3	5
TC	1	2	1	2	1	1	1	2
TC +2SF I4/R8	17	46	16	44	8	26	2	5
ET 9×9	5	18	4	18	2	8	1	2
ET 9×9 +2SF I4/R8	20	62	18	63	7	29	2	6
CF	10	12	9	11	8	10	8	10
CF+2SF I4/R8	12	13	10	12	9	11	10	11

Table 6.18: Processing time of the un-optimized C++ implementation of the stereo matching algorithms (SAD, TC, ET, CF) and improvement techniques (+BP, +2SF). Rows show algorithm and improvement technique and the current parameter settings. Columns show the different test sets A-D with 200(50) or 600(150) timestamps, and in parentheses the number of events processed.

6.4 Summary

At the beginning of this chapter we introduced the methods we used for testing and evaluating the different algorithms. The first method used objects at fixed distances and calculated depth values which are compared with those distances (used in test series 1 with a silicon retina sensor resolution of 128×128). The disadvantage of this method is that objects with curved surfaces are not correctly and accurately evaluated. Therefore, we developed a set-up to capture ground truth test data which allowed a pixel-wise evaluation of the calculated depth values (used in test series 2 with a silicon retina sensor resolution of 304×240).

For an expand analysis of the different approaches, we focused on the pixel-wise evaluation method in TS2 because the results were more suitable for a quantitative interpretation of the algorithm outcome. In TS2 we used four different test sets with objects at distances between 1.5m and 4m. Before the evaluation of the algorithms, we introduced the two ratios R_D and R_E, which allowed a more precise interpretation of the algorithms' results.

Our first test was the evaluation of the pre-processing approaches of the event data. In this context we mainly addressed the reduction of noise using different filters for pre-processing. After that we tested the implemented area-based and feature-based algorithms, which require an event-to-image conversion before applying the algorithm. These algorithms are dependent on the conversion process because they do not operate directly on the event stream received from silicon retina sensor. Different correlation metrics and a non-parametric transform for events were chosen as representatives for the area-based category. For tests of a feature-based approach we chose a corner-feature matcher. Using feature-based approaches in our work was generally limited by the fact that the event data are sparse, which reduces the number of features that are candidates for matching.

We also developed an event-based time correlation algorithm which operates on the event data without a prior conversion step. This approach was implemented to exploit the uniqueness of event-data and to show the differences in stereo matching compared to the area-based and feature-based algorithms mentioned above.

In addition to the evaluation of different stereo matching approaches, we evaluated various methods to improve the results of the basic stereo matching algorithms. For this reason, two simple filters (Median Filter, Average Filter) were applied to the stereo matching outcome. A more advanced approach was an adapted *Belief Propagation* algorithm. This optimization technique was adapted to operate with the sparse event data. Another improvement method utilized was a specialized post-processing filter named *Two-Stage Filter*.

The evaluation of the different algorithms was concluded with a time analysis, where we analyzed and compared the time consumption of certain algorithms. This assisted in the understanding of the basic complexity of the algorithms. Additionally, we demonstrated an overlay of the calculated error and the captured grayscale test image to provide a visual overview of critical areas to match, as well as showing where areas of high error incidence occurred.

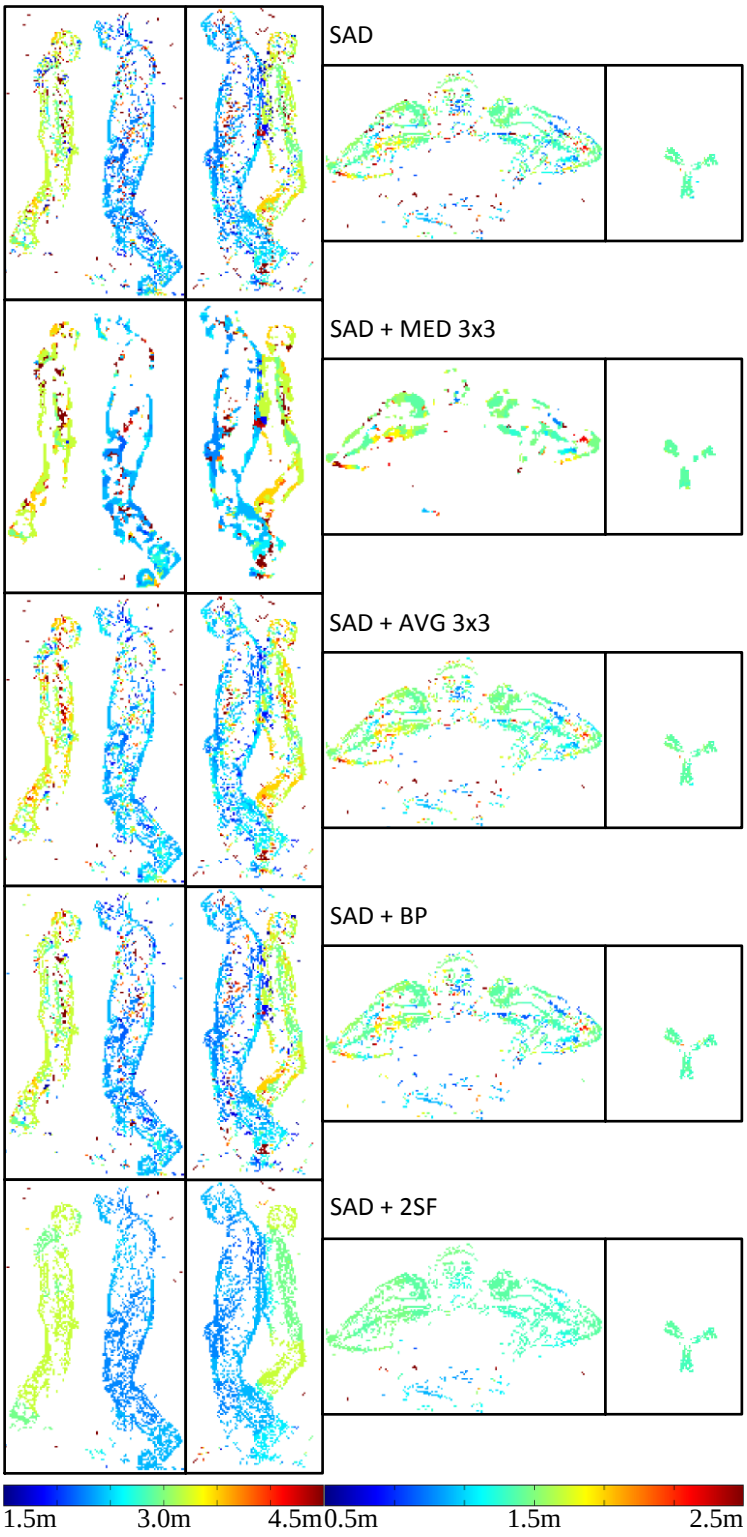


Figure 6.39: Depth results of the SAD algorithm and the applied improvement techniques.

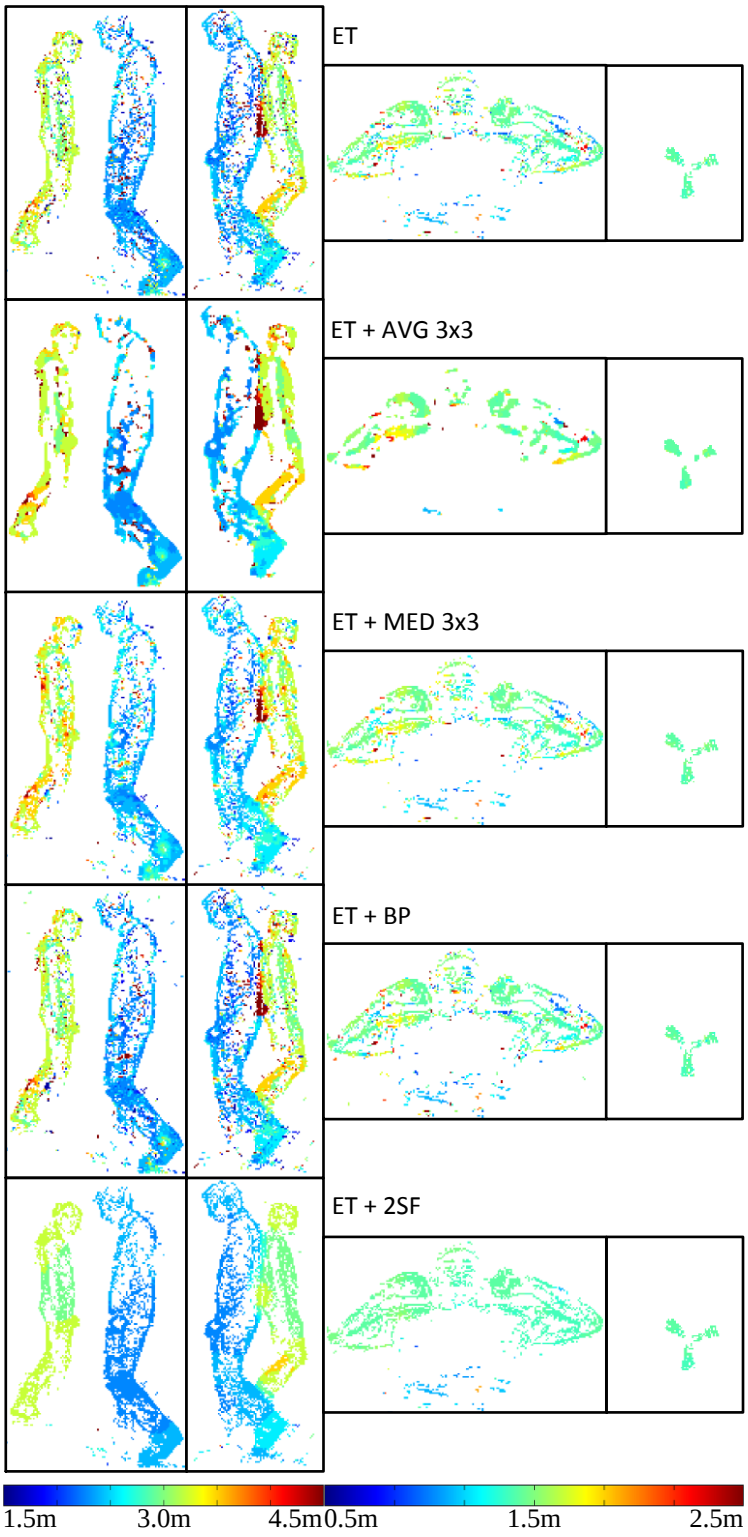


Figure 6.40: Depth results of the event transform algorithm and the applied improvement techniques.

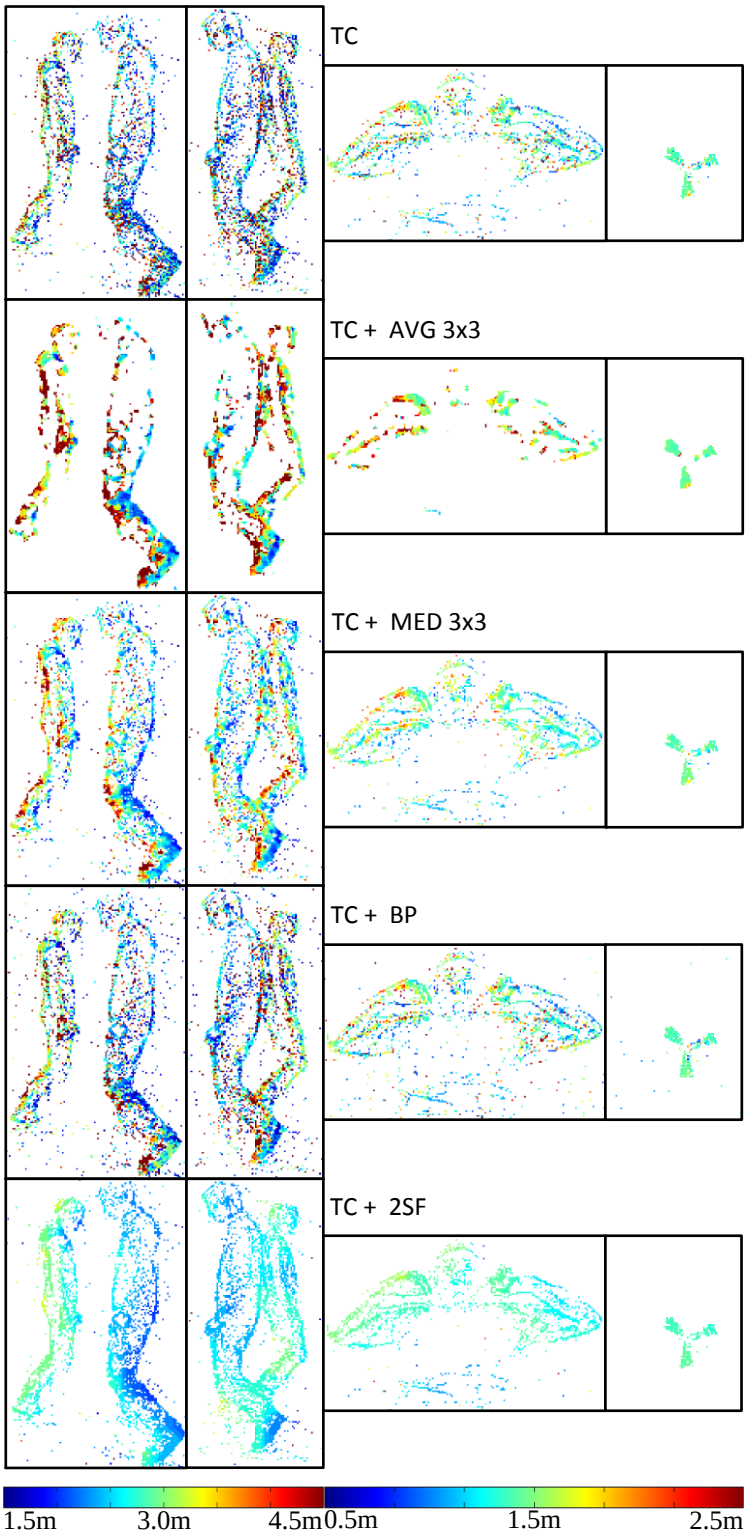


Figure 6.41: Depth results of the event-based time correlation algorithm and the applied improvement techniques.

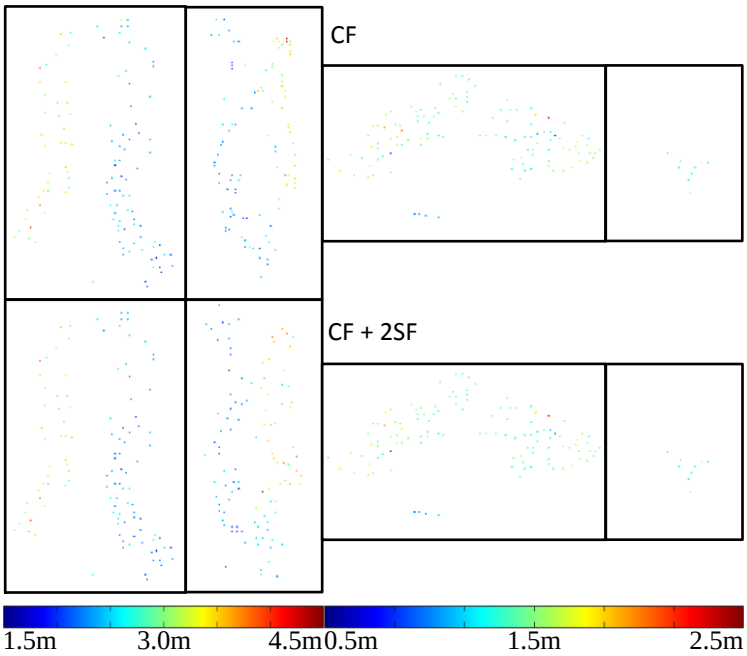


Figure 6.42: Depth results of the corner feature matcher algorithm and the applied improvement technique.

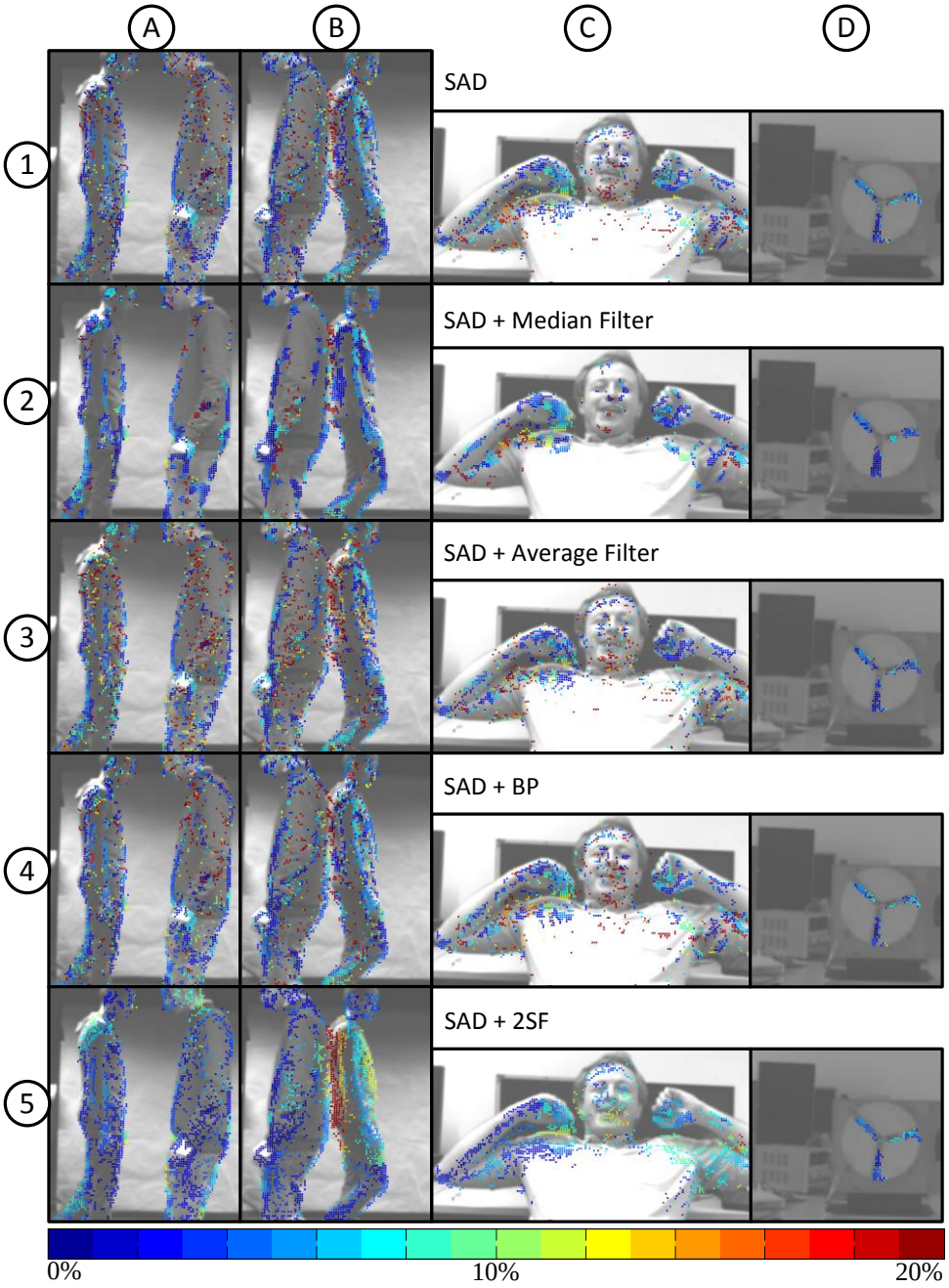


Figure 6.43: Overlay of the depth error with the corresponding, grayscale image (from the ground truth stereo vision system) of test data set A, B, C and D. Row 1 shows the result of the SAD algorithm without improvement technique. Rows 2-5 show the results if the improvement technique (Median, Average, BP, 2SF) were applied. The error range from 0 to 20% is shown in color coded representation. Errors larger than 20%, are scaled to 20% for better viewing.

CHAPTER 7

Conclusions and Outlook

In this thesis we have focused on solving the correspondence problem of sparse data captured with a silicon retina stereo camera system. The first part, Section 7.1, of this concluding section summarizes the results and findings of the thesis. In the second part, represented by Section 7.2, we give an overview of future work which could be carried out based on the thesis' results.

7.1 Conclusions

A major difference between the silicon retina camera and a conventional camera is the sparse appearance of the retina camera's data, as well as the little information encoded in a single event, in addition to the process for data retrieval and processing. This is why stereo matching using silicon retina cameras is considered to be a challenging task. One of the points that requires special attention is the transformation of the sparse silicon retina data into images suitable for stereo matching. The dynamics of a scene directly influence which time history must be chosen for the conversion process. As a consequence, the selection of a suitable time history for data acquisition can become difficult for scenes that contain several objects moving at significantly different speeds.

After converting the original event data into images that were more suitable for stereo matching, an additional pre-processing step was applied, with a focus on noise reduction (noise filtering). At this point, a connected component noise filter performed better in comparison to the conventional median filter used for the noise reduction. For our test data, the median filter considered 49%-87% of the input data as noise in contrast to the connected component filter, which filtered 32%-44% of the same input data as noise, while improving the average distance error up to 30%. Therefore, the connected component filter was found to be a suitable noise filter for the sparse silicon retina data and was applied for all further tests in this thesis.

We investigated both area-based and feature-based matching approaches, we compared seven different correlation metrics including a non-parametric local transform

(event transform) similar to the Census transform. The results showed that all seven correlation metrics achieved a similar range of average distance error depending on the time history and window size chosen. For the test set A, for example, the average distance error ranges between 0.2m-0.7m and for test set D between 0.05m-0.1m. The event transform in comparison to the other six correlation metrics (SAD, ZSAD, LSAD, SSD, ZSSD, LSSD) achieves - depending on the test set chosen - an up to 17% higher ratio R_D (represents the amount of events processed from all input events) and an up to 7% higher ratio R_E (represents the amount of events evaluated from the processed events and contribute to the calculation of the average distance error). The usage of tri-state or dual-state logic within the event transform has no recognizable impact on the results. We therefore suggest to use the the dual-state logic because of its lower calculation complexity.

The outcome of the chosen feature-based approach cannot be compared directly with the other algorithms, because instead of the whole event image only a few extracted features are used for the matching process. That means regarding the ratios R_D and R_E a direct mapping to the results of the seven area-based correlation metrics is not possible. But considering the absolute average distance error calculated for the features detected, the corner feature matcher achieved approximately 0.15m (test set A with a R_E of 98%) and 0.03m (test set D with a R_E 100%), which suggests that those feature points can be used as reliable pre-known anchors points for other matching approaches.

We implemented an event-based time correlation algorithm that works directly on the event data and showed different results regarding average distance error and evaluation ratios. The best average distance error was for test set A about 0.58m (time history 200) and for test set D about 0.14m (time history 50). Comparing the ratio R_E, which was approximately 80% for test set A and approximately 98% for test set D, the time correlation algorithm evaluates a high percentage of the processed input events but fails during the matching process due to the sparseness of valid input data.

After comparing the different stereo matching algorithms, we evaluated the ability of improving the matching results by applying additional methods including a sparse *Belief Propagation* (BP) approach and several post-processing techniques as described in Section 5.3. All the numerical results depend on the algorithm settings chosen, but the results summarized in the following are based on an area-based SAD algorithm with a time history of 400 (test set A-C)/100 (test set D) and a matching window of 9×9. Considering the median filter and average filter as improvement methods, the median filter is not recommended because it was found to remove an excessive number of results (R_D ratio is in average 41% lower than for the average filter) even though the average filter increased the results up to 33% in contrast to the median filter, which improved the average distance error up to 40%. Furthermore, we applied an adapted belief propagation approach and tested how much improvement could be derived. The BP optimized the matching procedure itself and was very sensitive to the sparse nature of the input data, which resulted mainly in minor improvements up to about 8%, and in some cases yielded even worse results. The *Two-Stage Filter* (2SF) showed the best performance of all of the improvement techniques. Regarding the reduction of the

average distance error, the 2SF decreased the error in average between 20% and 50% and achieves at the same time a high ratio R_E of up to 94%. This is in contrast to the SAD algorithm which has a ratio R_E of 83% and no applied improvement technique an increased R_E of approximately 13%. Comparing all improvement techniques the 2SF has the best overall performance and an execution time that is by a factor 3 faster than the BP technique (under similar algorithm settings).

Overall the results have shown that silicon retina cameras need specifically tailored approaches to process the data and to compute satisfying depth results, so that subsequent applications such as high speed analysis or automotive applications with different dynamic lighting conditions can benefit from the advantages of a silicon retina sensor.

7.2 Future Work

We consider this work as finished, however we suggest the following research activities as potential next steps.

The presented algorithms can be adapted regarding a dynamic time history based on the scene's dynamics. The matched corner features can be used as reliable anchor points for guided dense stereo matching approaches. The time correlation approach could be additionally extended and evaluated with different windowing techniques. From a more practical point of view, we suggest to reimplement the calibration procedure to reduce the manual workload.

Furthermore, an optimized implementation of the algorithms on different, e.g. embedded, platforms would be necessary to enhance applicability.

Bibliography

[1] P. Azad, T. Gockel, and R. Dillmann. *Computer Vision - Principles and Practice*. Elektor International Media BV, 1st edition, 2008.

[2] B. E. Bayer. Color imaging array, 1976. US Patent 3,971,065.

[3] R. Bellman. *Dynamic Programming*. Princeton University Press, 1st edition, 1957.

[4] R. Benosman, S.-H. Ieng, P. Rogister, and C. Posch. Asynchronous event-based hebbian epipolar geometry. *IEEE Transactions on Neural Networks*, 22(11):1723–1734, 2011.

[5] A. Bharath and M. Petrou. *Next Generation Artificial Vision Systems - Reverse Engineering the Human Visual System*. Artech House, 1st edition, 2008.

[6] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 35(3):269–293, 1999.

[7] M. Bleyer and M. Gelautz. Simple but effective tree structures for dynamic programming-based stereo matching. In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 415–422, Funchal/Portugal, 2008.

[8] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo - stereo matching with slanted support windows. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 1–11, Dundee/UK, 2011.

[9] K. Boahen. Retinomorphic chips that see quadruple images. In *Proceedings of the 7th International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems (MicroNeuro).*, pages 12–20, Granada/Spain, 1999.

[10] K. Boahen. Point-to-point connectivity between neuromorphic chips using address events. *IEEE Journal of Transactions on Circuits and Systems II*, 47(5):416–433, 2000.

[11] J. Y. Bouguet. Camera calibration toolbox for MATLAB. Published in the Internet, 2008. Computer Vision Research Group/Department of Electrical Engineering/California Institute of Technology - `www.vision.caltech.edu/bouguetj/calib_doc/index.html` (Accessed 16 June 2012).

[12] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001.

[13] G. Bradski and A. Kaehler. *Learning OpenCV - Computer Vision with the OpenCV Library*. O'Reilly Media Inc., 1st edition, 2008.

[14] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Journal of Transactions on Pattern Analysis and Machine Intelligence*, 25:993–1008, 2003.

[15] W. Burger and M. J. Burge. *Digital image processing an algorithmic introduction using JAVA*. Springer-Science/Business Media LLC, 2005.

[16] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *Proceedings of the 11th European Conference on Computer Vision: Part IV (ECCV)*, pages 778–792, Heraklion/Greece, 2010.

[17] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–139, 1990.

[18] J. Carneiro, I. Sio-Hoi, C. Posch, and R. Benosman. Event-based 3d reconstruction from neuromorphic retinas. *Journal of Neural networks*, 45:27–38, 2013.

[19] J.-H. Cho and M. Humenberger. Fast patchmatch stereo matching using cross-scale cost fusion for automotive applications. In *Proceedings of the International Intelligent Vehicles Symposiom (IV)*, pages 802–807, Seoul/Korea, 2015.

[20] J. Costas-Santos, T. Serrano-Gotarredona, R. Serrano-Gotarredona, and B. Linares-Barranco. A spatial contrast retina with on-chip calibration for neuromorphic spike-based aer vision systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(7):1444–1458, 2007.

[21] E. Culurciello, R. Etienne-Cummings, and K. A. Boahen. A biomorphic digital image sensor. *IEEE Journal of Solid-State Circuits*, 38(2):281–294, 2003.

[22] L. De-Maeztu, A. Villanueva, and R. Cabeza. Near real-time stereo matching using geodesic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):410–416, 2012.

[23] M. Domínguez-Morales, E. Cerezuela-Escudero, A. Jiménez-Fernandez, R. Paz-Vicente, J. L. Font-Calvo, P. Iñigo-Blasco, A. Linares-Barranco, and G. Jiménez-Moreno. Image matching algorithms in stereo vision using address-event-representation - A theoretical study and evaluation of the different algorithms. In *Proceedings of the International Conference on Signal Processing and Multimedia Applications (SIGMAP) which is part of the International Joint Conference on e-Business and Telecommunications (ICETE)*, pages 79–84, Seville/Spain, 2011.

[24] F. Eibensteiner. *Hardware Architecture of an Event-Driven Stereo Vision Algorithm Based on Silicon Retina Sensors*. Phd-thesis, Johannes Kepler University Linz, 2016.

[25] F. Eibensteiner, A. Gschwandtner, and M. Hofstätter. A high-performance system-on-a-chip architecture for silicon-retina-based stereo vision systems. In *Proceedings of the International Congress on Computer Application and computational Science (IRAST)*, pages 976 – 979, Bali/Indonesia, 2010.

[26] F. Eibensteiner, J. Kogler, and J. Scharinger. A high-performance hardware architecture for a frameless stereo vision algorithm implemented on a FPGA platform. In *Proceedings of the 10th IEEE Embedded Vision Workshop EVW (held in conjunction with IEEE CVPR)*, Columbus/USA, 2014.

[27] F. Eibensteiner, J. Kogler, M. Schörghuber, and J. Scharinger. Automated stereo calibration for event-based silicon retina imagers. In *Proceedings of the 6th International Conference from Scientific Computing to Computational Engineering (IC-SCCE)*, Athens/Greece, 2014.

[28] F. Eibensteiner, J. Kogler, C. Sulzbachner, and J. Scharinger. Stereo-Vision Algorithm Based on Bio-Inspired Silicon Retinas for Implementation in Hardware. In *Proceedings of the 13th International Conference on Computer Aided Systems Theory (EUROCAST)*, Lecture Notes in Computer Science, pages 624–631, Las Palmas/Spain, 2011.

[29] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International Journal on Computer Vision*, 70(1):41–54, 2006.

[30] A. Fusiello, V. Roberto, and E. Trucco. Efficient stereo with multiple windowing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 858–863, Washington/USA, 1997.

[31] K. R. Gegenfurtner. *Gehirn und Wahrnehmung*. Fischer Taschenbuch Verlag, 2006.

[32] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence/USA, 2012.

[33] M. Gong and Y. Yee-Hong. Fast stereo matching using reliability-based dynamic programming and consistency constraints. In *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 610–617, Nice/France, 2003.

[34] R. C. Gonzales and R. E. Woods. *Digital image processing - Second edition*. Prentice Hall/Pearson Education International, 2002.

[35] R. W. Hamming. Error detecting and error correcting code. *Bell System Technical Journal*, 29(2):147–160, 1950.

[36] R. Harteley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2003.

[37] R. I. Hartley. An algorithm for self calibration from several views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 908–912, Seatlle/USA, 1994.

[38] R. I. Hartley. Theory and practice of projective rectification. *International Journal of Computer Vision*, 35:115–127, 1999.

[39] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1106–1112, Washington/USA, 1997.

[40] P. Hess. Low-level stereo matching using event-based silicon retinas, 2006. Semesterarbeit am Institut für Neuroinformatik, ETH Zürich - `http://www.ini.uzh.ch/~tobi/studentProjectReports/hessAERStereo2006.pdf` [Accessed 16 March 2009].

[41] H. Hirschmüller. Improvements in real-time correlation-based stereo vision. In *Proceedings of the IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV)*, pages 141–148, Washington/USA, 2001.

[42] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 807–814, San Diego/USA, 2005.

[43] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[44] H. Hirschmüller, P. R. Innocent, and J. Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47(1-3):229–246, April 2002.

[45] H. Hirschmüller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1582–1599, 2009.

[46] A. Hosni, M. Bleyer, and M. Gelautz. Secrets of adaptive support weight techniques for local stereo matching. *Journal of Computer Vision and Image Understanding*, 117(6):620–632, 2013.

[47] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):504–511, 2013.

[48] P. M. Hubel and M. Bautsch. Resolution for color photography. In *Proceedings of SPIE 6069, Digital Phototgraphy II, 60690M*.

[49] M. Humenberger, C. Zinner, M. Weber, W. Kubinger, and M. Vincze. A fast stereo matching algorithm suitable for embedded real-time systems. *Journal of Computer Vision and Image Understanding*, 114(11):1180–1202, 2010.

[50] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, 1994.

[51] J. Kogler, C. Sulzbachner, F. Eibensteiner, and M. Humenberger. Address-event matching for a silicon retina based stereo vision system. In *Proceedings of the 4th International Conference from Scientific Computing to Computational Engineering (IC-SCCE)*, pages 17–24, Athens/Greece, 2010.

[52] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 508–515, Vancouver/Canada, 2001.

[53] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128x128 120 dB 30 mW asynchronous vision sensor that responds to relative intensity change. In *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC)*, SanFrancisco/USA, 2006.

[54] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128x128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2), 2008.

[55] D. Litwiller. CCD vs. CMOS: Facts and fiction. *Photonics Spectra*, 36(1), 2002.

[56] D. Litwiller. CMOS vs. CCD: Maturing technologies, maturing markets. *Photonics Spectra*, 39(8), 2005.

[57] M. A. Mahowald and T. Delbrück. Cooperative stereo matching using static and dynamic image features. In Carver Mead and Mohammed Ismail, editors, *Analog VLSI Implementation of Neural Systems*, volume 80 of *The Kluwer International Series in Engineering and Computer Science*, pages 213–238. Springer US, 1989.

[58] M. K. Mahowald. *VLSI analogs of neuronal visual processing: a synthesis of form and function*. Phd-thesis, California Institute of Technology, 1992.

[59] M. K. Mahowald and C. Mead. *Analog VLSI and Neural Systems - Chapter Siliocn Retina*. Addison Wesley Publishing Company, 1st edition, 1989.

[60] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194(4262):283–287, 1976.

[61] C. Mead and M.A. Mahowald. A silicon model of early visual processing. *Journal of Neural Networks*, 1(1):91–97, 1988.

[62] S. Meister, B. Jähne, and D. Kondermann. Outdoor stereo camera system for the generation of real-world benchmark data sets. *Journal of Optical Engineering*, 51(02):021107, 2012.

[63] J. J. Moré. The levenberg-marquardt algorithm: Implementation and theory. In *Numerical Analysis*, Lecture Notes in Mathematics. Springer Berlin Heidelberg.

[64] E. Mueggler, B. Huber, and D. Scaramuzza. Event-based, 6-DOF pose tracking for high-speed maneuvers. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 2761–2768, Chicago/USA, 2014.

[65] E. Piatkowska, A. N. Belbachir, and M. Gelautz. Asynchronous stereo vision for event-driven dynamic stereo sensor using an adaptive cooperative approach. In *Proceedings of the 3rd Workshop on Consumer Depth Cameras for Computer Vision (CDC4CV) (held in conjunction with IEEE International Conference on Computer Vision (ICCV))*, Sydney/Australia, 2013.

[66] E. Piatkowska, A. N. Belbachir, and M. Gelautz. Cooperative and asynchronous stereo vision for dynamic vision sensors. *Measurement Science and Technology*, 25(5):1–8, 2014.

[67] C. Posch. Bio-inspired vision. *Journal of Instrumentation*, 7(01):C01054, 2012.

[68] C. Posch, D. Matolin, and R. Wohlgenannt. An asynchronous time-based image sensor. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2130–2133, Seattle/USA, 2008.

[69] C. Posch, D. Matolin, and R. Wohlgenannt. High-dr frame-free pwm imaging with asynchronous aer intesity encoding and focal-plane temporal redundancy suppression. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2430–2433, Paris/France, 2010.

[70] C. Posch, D. Matolin, and R. Wohlgenannt. A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. *IEEE Journal of Solid-State Circuits*, 46(1):259–275, 2011.

[71] R. Ramanath, W. E. Snyder, Bilbro G. L., and W. A. Sander III. Demosaicking methods for Bayer color arrays. *Journal of Electronic Imaging*, 11:306–315, 2002.

[72] R. W. Rodieck. *The First Steps in Seeing*. Sinauer Associates, Inc., Sunderland, Massachusetts, 1998.

[73] P. Rogister, R. Benosman, I. Sio-Hoi, P. Lichtsteiner, and T. Delbruck. Asynchronous event-based binocular stereo matching. *IEEE Transactions on Neural Networks and Learning Systems*, 23(2):347–353, 2012.

[74] P. F. Ruedi, P. Heim, F. Kaess, E. Grenet, F. Heitger, P. Y. Burgi, S. Gyger, and P. Nussbaum. A 128x128 pixel 120 dB dynamic-range vision-sensor chip for image contrast and orientation extraction. *IEEE Journal of Solid-State Circuits*, 38(12):2325–2333, 2003.

[75] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Journal of Computer Vision*, 47(1-3):7–42, 2002.

[76] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, pages 195–202, 2003.

[77] M. Schörghuber. *Kalibrierung von Stereo-Visionen basierend auf Silicon Retina*. Bachelorthesis, University of Applied Sciences Hagenberg, 2011.

[78] S. Schraml, P. Schön, and N. Milosevic. Smartcam for real-time stereo vision - address-event based embedded system. In *Proceedings of the 2nd International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 2, pages 466–471, Barcelona/Spain, 2007.

[79] O. Schreer. *Stereoanalyse und Bildsynthese*. Springer Verlag Berlin Heidelberg, 2005.

[80] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, pages 593–600, Seattle/USA, 1994.

[81] K. Shimonomura, T. Kushima, and T. Yagi. Binocular robot vision emulating disparity computation in the primary visual cortex. *Journal of Neural Networks*, 21(2-3):331–340, 2008.

[82] C. Shoushun and A. Bermak. A Low Power CMOS Imager based on Time-to-First-Spike encoding and Fair AER. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2005.

[83] C. Shoushun and A. Bermak. Arbitrated time-to-first spike cmos image sensor with on-chip histogram equalization. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 15(3):346–357, 2007.

[84] M. Sivilotti. *Wiring consideration in analog vlsi systems with application to field programmable networks*. Phd-thesis, California Institute of Technology, 1991.

[85] G. P. Stein. Accurate internal camera calibration using rotation, with analysis of sources of error. In *Proceedings of the 5th International Conference on Computer Vision (ICCV)*, pages 230–236, Boston/USA, 1995.

[86] G. W. Stewart. On the early history of the singular value decomposition. *SIAM Review*, 35(4):551–566, 1993.

[87] C. Sulzbachner, J. Kogler, and F. Eibensteiner. A novel verification approach for silicon retina stereo matching algorithms. In *Proceedings of the 52nd International Symposium Electronics in Marine (ELMAR)*, pages 467–470, Zadar/Croatia, 2010.

[88] C. Sulzbachner, C. Zinner, and J. Kogler. An optimized silicon retina stereo matching algorithm using time-space correlation. In *Proceedings of the 7th IEEE Embedded Computer Vision Workshop ECVW (held in conjunction with IEEE CVPR)*, pages 1–7, Colorado Springs/USA, 2011.

[89] J. Sun, N.-N. Zheng, and H.-Y. Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800, 2003.

[90] F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda. Classification and evaluation of cost aggregation methods for stereo correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Anchorage/USA, 2006.

[91] B. Triggs. Autocalibration from planar scenes. In *Proceedings of the 5th European Conference on Computer Vision (ECCV)*, pages 89–105, Freiburg/Germany, 1998.

[92] R. Y. Tsai. A versatile camera calibration techniaue for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, 1987.

[93] T. Vaudrey, C. Rabe, R. Klette, and J. Milburn. Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In *Proceedings of the 23rd International Conference Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6, Christchurch/NewZealand, 2008.

[94] O. Veksler. Fast variable window for stereo correspondence using integral images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 556–561, Washington/USA, 2003.

[95] T. Wright. *Visual Impact: Culture and the Meaning of Images*. Bloomsbury Academic, 1st edition, 2009.

[96] G. Xiaochuan, Q. Xin, and J. G. Harris. A time-to-first-spike CMOS image sensor. *IEEE Sensors Journal*, 7(8):1165–1175, august 2007.

[97] K.-J. Yoon and I. S. Kweon. Locally adaptive support-weight approach for visual correspondence search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[98] K.-J. Yoon and I. S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):650–656, April 2006.

[99] S. Yoon, D. Min, and K. Sohn. Fast dense stereo matching using adaptive window in hierarchical framework. In *Proceedings of the 2nd International Symposium on Visual Computing (ISVC)*, pages 316–325, Lake Tahoe/USA, 2006.

[100] R. Zabih and J. Woodfill. Non-parametic local transforms for computing visual correspondence. In *Proceedings of the 3rd European Conference on Computer Vision (ECCV)*, pages 151–158, Stockholm/Sweden, 1994.

[101] K. A. Zaghloul and K. Boahen. Optic nerve signals in a neuromorphic chip i: Outer and inner retina models. *IEEE Transactions on Biomedical Engineering*, 51(4):657–666, 2004.

[102] K. A. Zaghloul and K. Boahen. Optic nerve signals in a neuromorphic chip ii: Testing and results. *IEEE Transactions on Biomedical Engineering*, 51(4):667–675, april 2004.

[103] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 666–673, Kerkyra/Greece, 1999.

[104] L. Zhou, Z. Lv, H. Song, and C. Hao. Accelerated belief propagation for hardware implementation. In *Proceedings of the International Conference on Multimedia Technology (ICMT)*, pages 128–131, Hangzhou/China, 2011.

Curriculum Vitae

Jürgen Kogler

Vienna University of Technology
Favoritenstrasse 9-11
1040 Vienna, Austria

Email: juergen.kogler@tuwien.ac.at

Education

2008-2016	Doctoral program in technical sciences at the Vienna University of Technology. Institute of Software Technology and Interactive Systems. Thesis: *Design and Evaluation of Stereo Matching Techniques for Silicon Retina Cameras*.
2006-2008	MSc. - Master studies at the University of Applied Sciences Technikum Wien. Field of study: *Information Systems Management*. Thesis: *Machbarkeitsstudie und Realisierung eines SMS-Dienstes zur Verwaltung von Zugangsberechtigungen*.
2001-2005	Dipl.-Ing. (FH) (with distinction) - Diploma studies at the Upper Austrian University of Applied Sciences in Hagenberg. Field of study: *Hardware/Software Systems Engineering*. Thesis: *Modellbasierte Entwicklung von Computer Vision Algorithmen für eingebettete Systeme*.
1999-2000	University entrance exam at the Upper Austrian University of Applied Sciences in Hagenberg.
1995-1999	Technical school for electrical engineering at the Linzer Technikum in Linz.
1991-1995	Secondary school in Linz.
1987-1991	Primary school in Linz.

Work Experience

2015-now	Senior Lecturer at the Vienna University of Technology, Institute of Computer Languages.
2014-2015	Assistant Lecturer at the Vienna University of Technology, Institute of Computer Languages.
2008-2014	Research Fellow at the AIT Austrian Institute of Technology GmbH, Department Safety and Security / Safe and Autonomous Systems.
2007-2008	Technical Consultant at the T-Systems Austria GmbH, Department Systems Integration / Enterprise Solutions.
2005-2007	Research Fellow at the ARC Seibersdorf research GmbH, Department Softwaresystems.
2004-2005	Internship at the ARC Seibersdorf research GmbH, Department Softwaresystems.

Research interests

- Stereo vision and 3D reconstruction
- Robots (UGV, UAV)
- Driver assistance systems
- Image processing applications in general