Panoramic Vision for Automotive Applications: From Image Rectification to Ambiance Monitoring and Driver Body Height Estimation

Christian N. Scharfenberger

TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Realzeit-Computersysteme

Panoramic Vision for Automotive Applications: From Image Rectification to Ambiance Monitoring and Driver Body Height Estimation

Christian N. Scharfenberger

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. (Univ. Tokio) M. Buss
Prüfer der Dissertation:	1. Univ.-Prof. Dr.-Ing. G. Färber (em.)
	2. Univ.-Prof. Dr.-Ing. E. Steinbach

Die Dissertation wurde am 18.11.2010 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 03.02.2011 angenommen.

Acknowledgements

This thesis originates from my works as research scientist and teaching assistant at the Institute for Real-Time Computer Systems at the Munich University of Technology. During my time at the Institute for Real-Time Computer Systems, many people have contributed to this dissertation.

First of all, I would like to thank my supervisor Professor Dr.-Ing. Georg Färber for his guidance, his patience and for always being available whenever I needed him. I owe gratitude to him to give my the opportunity to work on a fascinating topic and also for the multitude of challenges besides my scientific work. Professor Dr.-Ing. Georg Färber always supported me with his confidence, and he gave me the liberty to develop and to work out my own ideas dealing with „strange cameras“.

I want to thank Professor Dr.-Ing. Eckehard Steinbach for beeing the co-examiner of my thesis and Professor Dr.-Ing. Martin Buss for heading the committee. I would like to thank Professor Dr.-Ing. Samarjit Chakraborty for his support during my time at RCS and for being available whenever I needed him.

An important part is the great contribution of all the students who supported me over the entire process of writing this thesis. They did a really great job in their bachelor and master thesis and I want to thank Martin Almstätter, Florian Böhm, Khalil Fergani, Mohamed Sofiene Trigui, Pit Hoffmann, Michael Eibl, Martina Mayer, Markus Deicke, Chudong Mao, Tim Friederich, Zhiliang Zhou, Daniel Carton, Rui Zou, Hong Li, Jia Huang, Tarek Keskes, and Ichraf Ouannes. I will miss the time working with highly motivated students who found solutions for „unsolvable problems“. I really enjoyed supervising and working with them.

To all my colleagues and the staff at RCS, I am deeply grateful for their unconditional help whenever I needed it. It was a great time and the atmosphere at the RCS is really a welcoming one. Special thanks go to my office-mate Alejandro Masrur who always extended my limited knowledge of german grammar in discussions about „*Schönes Tag vs. Schönen Tag*“, and who always supported me during my time at RCS. Many thanks to all my colleagues involved in the MechaTUM project. I was happy to work with them in a great team even when the realization of the project leads to sleepless nights and to tensions (not always) before deadlines and demonstrations. Many thanks to the BMW-Group Germany for supporting this project and my research in the framework of CAR@TUM.

My work would not have been that enjoyable without the institute's administrative and technical staff. First, I want to thank Kurt Hettler for his help to build up the hardware I worked with. In particular, many thanks to him for his help in the never ending polishing of the mirrors for the omnidirectional camera and for his support during my time as a working student at RCS. Many

thanks to Renate Schwarz and Irene Dippold for their help with administrative duties and things like *Werkstudentenverträge* and *Reisekostenabrechnungen*.

I want to thank my parents for their support during my study and for everything they did for me. Most of all, I wish to thank Michelle Karg for her support, for all her patience and understanding during the entire process of writing this thesis – and for the time with *SCHLÄPPIE*.

Munich, 18th November 2010

Contents

List of Publications

The research presented in this thesis has resulted in the following publications and student's theses:

M. Strolz, Q. Mühlbauer, and C. Scharfenberger, G. Färber, and M. Buss. Towards a generic control system for actuated car doors with arbitrary degrees of freedom. *In Procedings of IEEE Intelligent Vehicles Symposium (IV)*, Eindhoven, Netherlands, 2008.

M. Fischer, S. Braun, D. Hellenbrand, C. Richter, O. Sabbah, C. Scharfenberger, M. Strolz, and G. Färber. Multidisciplinary developement of new door and seat concepts as part of an ergonomic ingress/egress support system. *In FISITA 2008 World Automotive Congress, Munich, Germany*, sep. 2008.

C. Scharfenberger, F. Böhm, and G. Färber. Image rectification: Evaluation of various projections for omnidirectional vision sensors using the pixel density. *In Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP), Lisbon, Portugal*, feb. 2009.

C. Scharfenberger, M. Fischer, S. Braun, D. Hellenbrand, O. Sabbah, C. Richter, M. Strolz and G. Färber. Multidisziplinäre Entwicklung von neuen Türkonzepten als ein Teil einer ergonomisch optimierten Ein/Ausstiegsunterstützung. *In VDI-OEM Forum Fahrzeugt¨uren und -klappen, Sindelfingen, Germany*, may 2009.

C. Scharfenberger, S. Chakraborty, and G. Färber. Robust image processing for an omnidirectional camera-based smart car door. *In Proceedings of the 7th IEEE Workshop on Embedded Systems for Real-time Multimedia (ESTIMedia)*, oct. 2009.

C. Scharfenberger and G. Färber. Robuste Erkennung von Verkehrsteilnehmern zur Vermeidung von Unfällen beim Türöffnen für ein Fahrerassistenzsystem. *In 4. Tagung Sicherheit durch Fahrerassistenzsysteme, TUEV-Sued, Munich, Germany*, apr. 2010.

C. Scharfenberger, S. Chakraborty, and G. Färber. Driver body-height prediction for an ergonomically optimized ingress using a single omnidirectional camera. *In 20th International Conference on Pattern Recognition (ICPR)*, aug. 2010.

C. Scharfenberger, P. Hoffmann, J. Leupold, and G. Färber. Vorrichtung zum Verhindern einer Verklemmung eines Fremdkörpers in einem Fahrzeugzugang. *In German Patent Disclosure, No. DE 102009040994 A1*, apr. 2010.

C. Scharfenberger, M. Eibl, J. Leupold, and G. Färber. Vorrichtung zum Vermessen einer sich einem Fahrzeug annähernden Person. *In German Patent Disclosure, No. DE 102009040995 A1*, apr. 2010.

C. Scharfenberger, F. Böhm, J. Leupold, and G. Färber. Vorrichtung zur Überwachung eines Raumbereiches ausserhalb eines Fahrzeuges. *In German Patent Disclosure, No. DE 102009057336 A1*, july 2010.

C. Scharfenberger, S. Chakraborty, and G. Färber. Robust image processing for an omnidirectional camera-based smart car door. *In ACM Transactions on Embedded Computing Systems (to appear)*, 2011.

M. Almstätter. Design und prototypische Realisierung einer positionsgeregelten Kameraplattform für ein optisches Sensorsystem. Master's thesis, TU Munich, Institute for Real-Time Computer Systems, aug. 2007.

T. Friederich. Bewegungsdetektion anhand eines adaptierten Referenzbildes mit Kompensation der Kameraeigenbewegung. Studienarbeit, TU Munich, Institute for Real-Time Computer Systems, feb. 2008.

F. Böhm. Tiefenschätzung mit omnidirektionalem Kamerasystem. Master's thesis, TU Munich, Institute for Real-Time Computer Systems, apr. 2008.

D. Carton. Fahrzeugtracking und Gefahreneinschätzung unter Verwendung eines omnidirektionalen Kamerasystems. Studienarbeit, TU Munich, Institute for Real-Time Computer Systems, oct. 2008.

L. Hong. Vollautomatische Detektion von Schachbrettmustern zur Kalibrierung omnidirektionaler Kamerasysteme. Bachelor's thesis, TU Munich, Institute for Real-Time Computer Systems, oct. 2008.

M.S. Trigui. Person Height Measurement with Unknown Camera Orientation to the Ground. Master's thesis, TU Munich, Institute for Real-Time Computer Systems, apr. 2009.

J. Huang. Weiterentwicklung und Verbesserung einer automatischen Kalibration für omnidirektionale Kameras. Master's thesis, TU Munich, Institute for Real-Time Computer Systems, sep. 2009.

List of Symbols

BG	Background
CPU	Central Processing Unit
DP	Dynamic Programming
FG	Foreground
FOV	Field of View
FPGA	Field Programmable Gate Array
FR	Frame
HMM	Hidden Markov Model
IC	Illumination Changes
MAX	Maximum
MIN	Minimum
NCC	Normalized Cross Correlation
NoW	Number of Search Windows
OCAM	Omnidirectional Camera
ODVS	Omnidirectional Vision System
RCS	Lehrstuhl für Realzeit–Computersysteme
RTP	Real–Time Processor
SAD	Sum of Absolute Differences
SGM	Semi-Global Matching
SM	Similarity Matrix
SPOV	Single Point of View
SQP	Sequential Quadratic Programming
SSD	Sum of Squared Differences
SVD	Singular Value Decomposition
VGA	Video Graphics Array
ZNCC	Zero–Mean Normalized Cross Correlation
ZSAD	Zero–Mean Sum of Absolute Differences
ZSSD	Zero–Mean Sum of Squared Differences

Abstract

Nowadays, omnidirectional cameras are becoming more and more attractive for industrial applications. Since omnidirectional cameras have a large field of view, there is a great potential for novel applications in the automotive domain such as advanced driver assistance systems among others.

This thesis presents a novel information system for a smart door. The smart car door consists of an actuated two-hinge kinematic along with a haptic support system which allows for situation-dependent door opening to provide optimal space for ingress/egress in tight parking lots. The information system is based on an omnidirectional camera which is integrated with the side-view mirror to monitor the ambiance next to the car in its entirety. The information system generates 3D ambiance information from the surroundings to compute situation-dependent door opening paths in order to avoid collisions with obstacles. The information system also extracts approaching drivers and estimates their body heights to automatically pre-adjust the seat for a better ingress.

This thesis focuses on the development of robust image processing algorithms for absolute body height estimation and on the generation of 3D ambiance information with a single omnidirectional camera. First, the field of omnidirectional cameras is introduced with particular focus on camera calibration and image rectification. The thesis presents the physical and mathematical properties of omnidirectional cameras and describes the underlying camera model to obtain perspective panoramic images. A calibration scheme is presented that estimates the parameters of the camera model using chessboard corners in calibration patterns. An extension to the calibration scheme is proposed that overcomes the previous manual selection of corners in existing calibration procedures. Instead, a robust extraction algorithm is presented that is able to detect chessboard corners in calibration images captured under different illumination conditions.

Different projections – such as the spherical or cylindrical projection – are presented to transform original images into panoramic images. Additionally, a new measure – the pixel density – is proposed as a new tool to compare omnidirectional cameras and projections in terms of best utilization of sensor pixels in panoramic images. This way, optimal utilization of sensor pixels and, hence, optimal resolution in panoramic images can be obtained for any omnidirectional camera. It is also shown that the commonly used cylindrical projection is not suitable for some omnidirectional cameras.

Next, the problem of absolute body height estimation of humans using a single omnidirectional camera is analyzed. A novel algorithm is proposed which estimates the absolute body height of approaching drivers in order to adjust the seat position for a better ingress in narrow parking situations. Body height estimation is realized as a two stage process: Driver extraction using a

Kalman-based background model and body height determination using a model-based function. The thesis describes a Kalman-based background model that has been extended by statistical functions to increase its robustness against shadows and illumination changes. Additionally, an initialization scheme is presented allowing for background initialization in scenarios with high volume of traffic. The key feature for enabling absolute body height estimation with a single camera is a known position and orientation of the camera relative to the ground. This position and orientation varies for each parking scenario and must, hence, automatically be determined from image data only. For this reason, a new model-based camera-ground function is introduced that estimates the orientation and the position of the camera relative to the ground. This estimation is based on image data obtained from approaching drivers only. The function explicitly considers camera tilt caused by inclined parked cars and has a global minimum when the estimated camera position best matches the real camera position. Then, body heights of approaching drivers are determined based on n-sets of extracted foot and head points.

The omnidirectional cameras, which is attached to each side-view mirror of a car, is also used to obtain 3D ambiance information of the surrounding next to the car. A mechanical device within the side-view mirror vertically positions the camera to provide a motion-based stereo configuration. The device is also equipped with a position sensor to determine the vertical positions of the camera. However, clearances in the mechanical device cannot be detected by the position sensor and lead to wrongly determined camera positions. Fur this purpose, an egomotion estimation algorithm refines the estimation of the camera positions using image correspondences only. The key problem addressed in this thesis is the generation of solid 3D ambiance information from low-textured and low-resolution panoramic images. First, the fundamentals of stereo vision with omnidirectional cameras are introduced. Secondly, a method is proposed to rectify panoramic images in order to ease 1D correspondence search in pairs of panoramic images. Thereafter, a stereo algorithm is presented to produce dense disparity maps from low-textured and low-resolution panoramic images. 3D ambiance information is generated using triangulation, and a new refinement stage is proposed to remove disturbances and outliers in 3D data obtained. Lastly, a new method is introduced to determine the position error in 3D-data. This position error depends both on the calibration error and on the quantization error of the omnidirectional camera system used in the stereo setup. Finally, the thesis analyzes measurement ranges and dead-zones of stereo setups based on omnidirectional cameras for several, common projections.

Zusammenfassung

Omnidirektionale Kameras sind heutzutage überwiegend in der Robotik zu finden. Durch ihren sehr großen Sichtbereich gegenüber perspektivischen Kameras und auch aufgrund ihres mittlerweile zunehmend kompakteren Aufbaus ergeben sich neue Anwendungsmöglichkeiten im Bereich der Fahrerassistenzsysteme.

In dieser Arbeit wird ein neuartiges Informationssystem vorgestellt, das den seitlichen Außenraum einer Fahrzeugtüre auf Hindernisse überwacht. Die gewonnenen Hindernisinformationen dienen einer Türsteuereinheit als Input für eine haptisch unterstützte Führung der Fahrzeugtür zur Vermeidung von Kollisionen beim Türöffnen. Das Informationssystem besteht aus einer omnidirektionalen Kamera, die in den Seitenspiegel des Fahrzeuges integriert ist und aus einer Bildverarbeitungseinheit zur Auswertung der gewonnenen Bilddaten. Neben der Hinderniserfassung wird mit diesem System die Größe von sich annähernden Fahrern bestimmt. Anhand der Größe wird die Position des Fahrersitzes individuell eingestellt und damit ein verbesserter Einstieg ermöglicht. Die Schwerpunkte dieser Arbeit liegen zum einen auf der Entwicklung von Bildverarbeitungsalgorithmen zur robusten Generierung von Umgebungsinformationen, und zum anderen auf der Entwicklung von Methoden zur absoluten Größenbestimmung des Fahrers mit nur einer Kamera.

Im ersten Teil der Arbeit werden die Grundlagen zu omnidirektionalen Kameras mit Fokus auf Kamerakalibration und Bildtransformation beschrieben. Neben der Beschreibung der physikalischen und mathematischen Eigenschaften omnidirektionaler Kameras wird ein Kameramodell vorgestellt, welches zur Transformation von Originalbildern der Kamera in perspektivisch korrekte Panoramabilder benötigt wird. Das Kameramodell stellt dabei gleichzeitig einen Zusammenhang zwischen Welt- und Kamerakoordinaten her. Die intrinsischen und extrinsischen Parameter der Kamera werden mit Hilfe von Schachbrettmustern und eines Kalibrierverfahrens bestimmt. In dieser Arbeit wird das bestehende Kalibrierverfahren um einen robusten Schachbrett- und Eckpunktextraktor erweitert, um die bisher notwendige manuelle Selektion der Eckpunkte zu automatisieren.

Es gibt viele Projektionsarten, die zur Transformation von Originalbildern in Panoramabilder verwendet werden können. In dieser Arbeit wird die Pixeldichte als ein neues Werkzeug zur Bewertung unterschiedlicher Projektionsarten und zur Bewertung verschiedener Kamerakonfigurationen vorgeschlagen. Die Pixeldichte ist ein Maß, welches die Ausnutzung der Sensorpixel in transformierten Bildern angibt. Es wird zudem gezeigt, dass die häufig verwendete zylindrische Projektion aufgrund schlechter Sensorpixelnutzung in Panoramabildern für viele Kamerakonfigurationen nicht geeignet ist.

Im zweiten Teil der Arbeit wird ein neuer Algorithmus zur absoluten Größenbestimmung des Fahrers mit einer omnidirektionalen Kamera vorgestellt. Anhand der Körpergröße des Fahrers wird die Position des Sitzes zur Verbesserung des Einstiegs voreingestellt. Die Größenvermessung wird als zweistufiges Verfahren realisiert: In einem ersten Schritt wird der Fahrer vom Hintergrund separiert und dessen Größe in einem zweiten Schritt mit Hilfe einer modellbasierten Funktion bestimmt. Zur Extraktion des Fahrers wird ein Kalman-basiertes Hintergrundmodell beschrieben, welches um Verfahren zur Kompensation von Schatten und Beleuchtungsänderungen erweitert wird. Weiterhin wird eine Initialisierung des Hintergrundes vorgestellt, welche eine Hintergrundgenerierung auch an vielbefahrenen Straßen ermöglicht. Eine absolute Schätzung der Körpergröße wird mittels der Relation der Kamera gegenüber dem Boden ermöglicht. Es wird eine neue, modellbasierte Funktion vorgeschlagen, mit deren Hilfe die Relation allein anhand der Bilddaten bestimmt werden kann. Diese Funktion berücksichtigt explizit Kameraverkippungen und hat ein globales Minimum, wenn die geschätzte Relation mit der tatsächlichen Relation übereinstimmt. Basierend auf dieser Relation und extrahierten Fuß- und Kopfpunkten kann die absolute Größe von sich annähernden Fahrern bestimmt werden.

Im letzten Teil der Arbeit wird ein auf Motion-Stereo basierender Ansatz zur Bestimmung von Hindernissen im Arbeitsraum der Tür vorgestellt. Dabei werden die Grundlagen zu Stereo mit omnidirektionalen Kameras sowie die Rektifizierung von Panoramabildern beschrieben. Bei der Rektifizierung werden Originalbilder derart in Panoramabilder transformiert, dass eine 1D-Korrespondenzsuche möglich ist. Mit Hilfe eines mechanischen Aufbaus wird die Kamera an verschiedene, vertikale Positionen gefahren und damit eine Stereokonfiguration erreicht. Aufgrund von Spiel in der Mechanik kann die Lage der Kameras zueinander sensorisch nicht genau bestimmt werden. Anhand von Bildkorrespondenzen jedoch kann die Lage der Kamera zueinander geschätzt und damit die sensorisch bestimmte Lage verbessert werden. Es wird weiterhin ein Stereoverfahren vorgestellt, mit dessen Hilfe dichte Disparitätskarten von texturarmen Panoramabildern mit geringer Auflösung berechnet werden können. 3D-Hindernisinformationen werden mit Hilfe von Triangulation erzeugt. Weiterhin wird ein Verfahren vorgeschlagen, welches Ausreißer in den Daten erkennt und beseitigt. Zudem wird der durch Quantisierungseffekte und durch Ungenauigkeiten in der Kalibration hervorgerufene Fehler der Hindernisinformationen beschrieben und die Messbereiche sowie die blinden Zonen der am häufigsten verwendeten Projektionen vorgestellt.

1 Introduction

Mechatronic systems are nowadays essential for the automotive industry and have gained continuously in importance. They are used for increasing car efficiency and reduce emission through efficient electronic engine management with exhaust aftertreatment. Mechatronic systems are also essential to meet global homologation requirements and environmental laws and numerous components are integrated and connected to in-car networks. They are employed to meet and to satisfy customers' needs for more comfort and safety. Despite the advantages achieved, there is still potential for further improvements and for new developments addressing passenger comfort related issues in the domain of automotive system engineering [2].

For these reasons, a novel research project originated in the context of an industry project, which was conducted at the Technische Universität München and the BMW Group, addresses new developments in mechatronic systems and comfort related issues. Based on the example of an ergonomically optimized ingress/egress support system, the main concern of the project is to improve mechatronic systems and the development processes deployed. The underlying scenario dealt with is an everyday problem of ingress and egress in tight parking lots: Particularly, long car doors cannot be opened wide enough to provide a comfortable and quick ingress/ egress in tight parking lots. For some car doors, passengers sometimes even have to steady the door during their movement due to weak door breaks or inappropriate locking positions. This leads to uncomfortable situations for passengers and to an increased discomfort during ingress/ egress. For this reason, passenger comfort-related issues have attracted a lot of attention and are intense research topics in the area of automotive ergonomics, in particular for ingress/egress in tight parking lots (see Figure 1.1).

Figure 1.1: Cars parked in tight parking lots. The car doors cannot be opened wide enough to provide optimal space for comfortable ingress/ egress.

With help of the potential of mechatronics, an innovative car door system – the *smart car door* – with ingress/ egress support was developed to maximize passengers comfort in today's two door cars. Ergonomic investigations are conducted and function as constraints for the technical

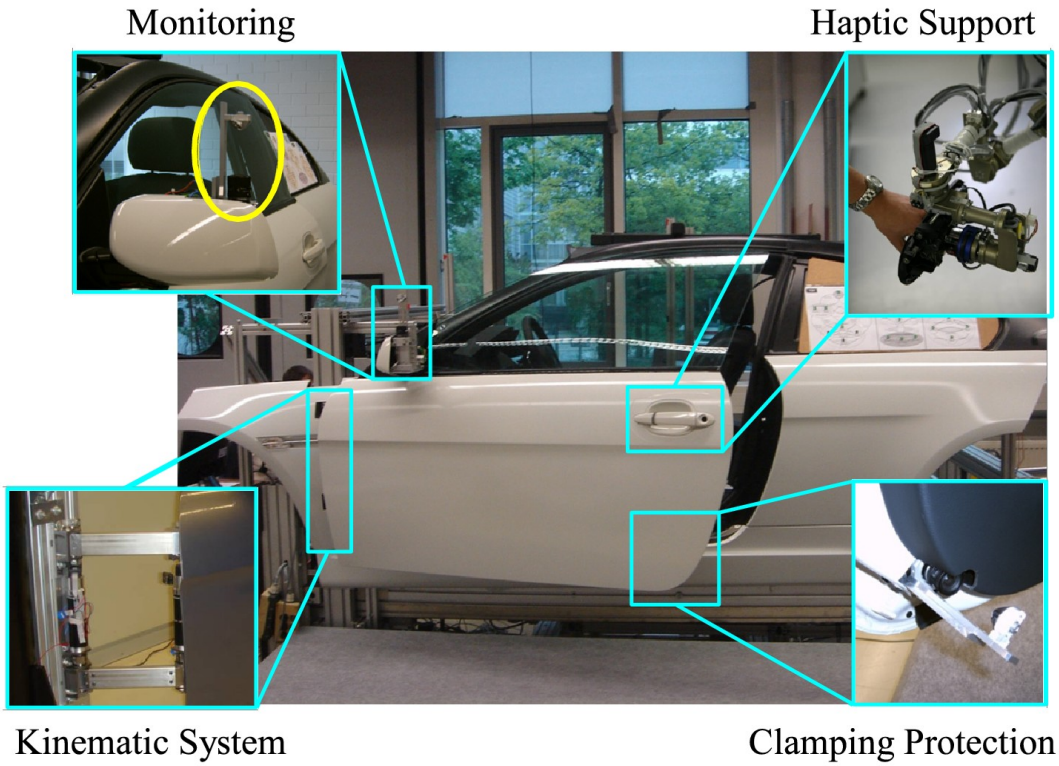


Figure 1.2: Prototype of the smart car door. The door consists of a multi-kinematic door, a haptic support system and an integrated camera to detect clamping situations. Additionally, a camera is integrated with the side-view mirror to monitor the ambiance next to the door and to estimate the body height of approaching drivers.

realization in different fields of development. On this account, a method has been built up to measure arising discomfort and to generate ergonomic information that is required for the development progress deployed [19]. Based on these constraints a multi-kinematic door was implemented that fulfills all necessary functions. The multi-disciplinary development process of the smart car door included five different fields of research. These fields are kinematics, sensorics, virtual prototyping and cybernetics, ergonomics and process development.

The smart car door consists of an actuated two hinge kinematic system to allow situation-dependent door openings and to provide optimal space for ingress/egress [20], [21]. A haptic support system along with a control unit supports the door user with haptic guidance along situation-dependent opening paths while opening the door [22]. The smart car door is equipped with sensors to obtain environment information of the surroundings next to the door: The information serves as an input for the control unit to compute situation-dependent opening paths in order to avoid potential collisions with collateral obstacles. Additionally, the sensors are used to detect approaching car drivers and passengers to estimate their body heights. The determined body heights function as a basis for personalized pre-adjustments of the seat and the vehicle's door opening path. Finally, the actuated smart car door has been evaluated by means of user studies. Figure 1.2 illustrates the smart car door prototype. The prototype consists of the actuated two-hinge door, the haptic support system and a camera integrated with the side-view mirror. The camera in question is an omnidirectional camera whose data are used to obtain 3D

Sensor Type	Sensor	Driver Assistance System
Non-Vision	Lidar	Adaptive Cruise Control (ACC)
	Radar	Blind Spot Monitoring (BSM)
	Ultrasound	Parking Assistance
		Collision Mitigation Systems (CMS)
		Automatic Emergency Break (AEB)
Vision	Rear View	Parking Assistance
	Side View	Parking Assistance
	Near Infra Read	Active Night View (Night View)
	Far Infra Read	Passive Night View (Night Vision)
	Single Camera Front View	Lane Departure Warning (LDW)
	Stereo Front View	Traffic Sign Recognition (TSR)
		Drowsiness Detection (DDS)
		Obstacle Detection
		Pedestrian Recognition

Table 1.1: Overview of commonly used sensors for advanced driver assistance systems (Murphy [23]).

ambiance information and to estimate body heights of approaching drivers. Another omnidirectional camera is attached inside the car door to avoid clamping situations by detecting obstacles between the car door and the door frame.

1.1 Motivation

This thesis focuses on the development of the sensor subsystem along with image processing algorithms for ambiance monitoring and driver body height estimation. The algorithms should obtain 3D-ambiance information of the surroundings next to the car door and should estimate body heights of approaching drivers. Based on the ambiance information the control unit computes collision-free, situation-dependent door openings and pre-adjusts the driver seat according to body heights. Ideally, the sensor subsystem should be suitable both to obtain ambiance information and to extract body heights of approaching drivers.

In today's automobiles a large number of sensors are used for driver assistance systems in high-end cars. These sensors monitor the surroundings of the car to detect obstacles or pedestrians. Their information serves as an input to advanced driver assistance systems to support drivers during driving or to assist them with parking vehicles into tight parking lots. Table 1.1 provides an overview of commonly used sensors for advanced driver assistance systems.

Most of the sensor systems integrated in high-end cars are designed to monitor the ambiance behind or in front of the car. For example, stereo cameras detect obstacles and pedestrians in front of the car, or Lidar-based sensor systems are used for Adaptive Cruise Control (ACC). However, there are only a few sensor systems that might be used to monitor lateral areas of vehicles. Side-view cameras are an example of a sensor system that might be used for ambiance monitoring. These cameras are attached to the side-view mirrors to display the areas next to the car and to compute a bird-eye view of the surroundings along with image data from front-view and rear-view cameras. Driver assistance systems use this bird-eye view to assist drivers when

(a) Blind spot monitoring (b) Parking assistance

Figure 1.3: Omnidirectional cameras attached to large vehicles for blind spot monitoring [24] (a) and as sensor system for parking assistance aid [25] (b).

parking the vehicle. The side-view cameras are designed to monitor the ambiance next to the car and are suitable for detecting obstacles in the proximity of the car door. But the side-view cameras have only a limited field of view, which is focused towards the floor regions close to the car so that only the lower parts of approaching drivers could be detected. For these reasons, side-view cameras are not suitable for extracting approaching drivers and for estimating their body heights.

In contrast to vision sensors, ultrasonic-, radar- or lidar-based sensors provide 3D-information of measured environments directly. But the vertical measurement range of such sensors is limited: Therefore, many sensors would be necessary to monitor the ambiance next to the car door in its entirety. Moreover, it is hard to measure body heights of approaching drivers with radar sensors or lidar sensors.

Omnidirectional cameras, however, overcome this limitation and provide both a large vertical field of view ($\approx 110°$) and a large horizontal field of view of $360°$. Figure 1.3 illustrates automotive applications that use omnidirectional cameras. In the first application, the cameras are attached to trucks in order to monitor blind spots of large vehicles [24] (see Figure 1.3(a)). The cameras are also attached to a car for providing a rear view of a car for a parking assisting systems [25] (see Figure 1.3(b)). Integrated with the side-view mirror of a car, omnidirectional cameras can both monitor the ambiance close to the door in its entirety and can extract approaching drivers. They are also suitable for combining the functions ambiance monitoring and driver body height estimation within one sensor subsystem.

The omnidirectional camera, which is integrated with the side-view mirror of the smart car door, consists of a perspective, monochromatic camera focusing on a hyperboloidal mirror. A very compact camera with VGA-resolution of 640×480 pixels has been chosen due to space and cost constraints. VGA-resolution might be sufficient for applications based on perspective cameras, but the use of imaging devices such as mirrors to enhance the horizontal field of view leads to a loss of resolution in panoramic images. Hence, image processing algorithms must be able to process panoramic images with low resolution.

The omnidirectional camera is suitable both to obtain 3D-ambiance information and to estimate body heights of approaching drivers. 3D-ambiance information can be obtained from two cameras or from one camera positioned at different poses. A fold in and fold out movement of the side-view mirror in high-end cars might be used to position the camera if the camera is integrated with the side-view mirror of a car. However, such a fold in and fold out movement was not available: Therefore, a mechanical device has been attached to the side-view mirror to vertically position the camera. The algorithm proposed in this thesis computes 3D-ambiance information from images captured at different poses.

1.1.1 Scenario

A typical interaction between an approaching driver and the smart car door for ingress may look something like this: The driver activates the sensor system when unlocking the door with the key. The camera is moved to several positions. At these positions images are captured and 3D-ambiance information is computed. Thereafter, the camera remains static and the algorithms extract drivers to estimate their body heights. Based on height data and on the 3D-ambiance information, the control unit pre-adjusts the driver seat to ease ingress and pre-computes individual opening paths to avoid collisions when opening the car door.

A similar procedure may look like this for standard egress scenarios: After parking, the car activates the smart car door. The camera is moved to several positions and 3D-ambiance information are computed. Based on this ambiance information, the control unit pre-computes potential opening paths to avoid collisions with obstacles close to the door when opening the car door. These scenarios allow to formulate certain preconditions under which driver extraction, body height estimation and 3D-ambiance generation algorithms may and should operate:

- All relevant obstacles in a parking scenario must be static while the camera moves.
- Obstacles must have a minimum distance of $50cm$ to the car in order to allow car door openings.
- To generate 3D-ambiance information, only obstacles in the required space for door opening are of interest.
- Less textured objects such as white walls or posts are very common in typical parking scenarios: Therefore, the algorithms must be able to obtain ambiance information from low-textured, gray-scaled and low-resolution panoramic images.
- Body height estimation is performed assuming a static camera for a short time interval. This interval may be used to compute a background image from the environment next to the door. The background image helps to separate approaching drivers from the environment and to estimate their body heights.
- The algorithms must be able to quickly extract approaching drivers – even if they are far away from the car – in low-resolution panoramic images.

1.1.2 Contributions of this thesis

This thesis focuses on the camera sub-system of the smart car door and on image processing algorithms to monitor the ambiance next to the door and to estimate body heights of approaching driver. In particular, the thesis addresses contributions in the area of camera calibration and image transformation, absolute body height estimation with a single omnidirectional camera and ambiance modeling of static surroundings close to the car door.

- A new extension to existing calibration algorithms is presented to automatically extract calibration pattern for camera calibration. The proposed method addresses robust extraction of calibration patterns in low resolution images under various illumination conditions. Additionally, this thesis analyzes the suitability of several projections to transform original images into panoramic images. It also demonstrates that the commonly chosen cylindrical projection is not the best projection to transform original images into panoramic images. Therefore, a novel measurement value is introduced that allows for an analysis of projections in terms of best utilization of sensor pixels in panoramic images. This value is also suitable for designing well-dimensioned mirror/ camera configurations to obtain best utilization of sensor pixels in panoramic images.

- A new method is presented to estimate absolute body heights of approaching drivers with single omnidirectional cameras integrated within a car. Body heights are used to individually adjust the driver seat in order to improve ingress in tight parking lots. The thesis describes how drivers can be extracted in low-resolution, gray-scaled panoramic images and how absolute body height estimation can be performed with a single camera. The proposed method is suitable for a wide range of parking scenarios and overcomes the scaling problem for camera-based measurements with single cameras.

- A motion-stereo-based algorithm is presented which generates 3D-ambiance information of the surroundings next to the car door. The proposed algorithm is able to obtain 3D-information from pairs of low-resolution panoramic images captured in low-textured environments – such as white walls in a parking garage. A mechanical device integrated with the side-view mirror vertically positions the camera to obtain a stereoscopic configuration. Camera positions are initially determined by a position sensor attached to the mechanical device and are refined by an image-based ego-motion estimation algorithm to overcome inaccuracies in the camera positions caused by mechanical clearances. The thesis also presents a new study that addresses 3D-position errors of the ambiance information computed from stereo disparity maps. The position error depends on the calibration and quantization error of the omnidirectional camera and strongly influences the quality of 3D-data obtained.

1.2 Thesis overview

All contributions described in Section 1.1.2 are presented in separate chapters. Each of these chapters includes a section that is dedicated to state-of-the-art, to discussion and to conclusion.

Chapter 2 introduces the field of omnidirectional cameras with particular focus on camera calibration and image rectification. It presents the physical and mathematical properties of omnidirectional cameras and describes the underlying camera model to obtain perspectively correct images. The camera model is also required to transform original images into panoramic images and to obtain 3D-data from sensor coordinates. A calibration scheme is presented which estimates the calibration parameters using calibration pattern. A novel extension to this calibration scheme is proposed to enable robust, automatic extraction of calibration patterns in low resolution images under various illumination conditions.

Several projections are presented to transform original images into panoramic images in order to ease image processing with omnidirectional cameras. Additionally, a new value – the pixel density – is proposed as a new method to evaluate projections for image transformation and to compare camera/ mirror configurations in terms of best utilization of sensor pixels for panoramic images. In this manner, best resolution in panoramic images can be achieved for any omnidirectional camera. It is also shown that the commonly chosen cylindrical projection is not suitable for some omnidirectional cameras. The pixel density as a new tool to compare several projections and camera configurations was first proposed in the field of camera calibration and image transformation.

Chapter 3 addresses the problem of absolute body height estimation using a single omnidirectional camera in the automotive domain. In this application, body heights of approaching drivers are estimated for adjusting the seat position in order to improve ingress in tight parking lots. Body height estimation is realized in two stages. These stages are driver extraction using a Kalman-based background model and body height determination using a model-based camera-ground function.

This chapter describes a Kalman-based background model that has been extended by statistical functions to increase its robustness against shadows and illumination changes. Additionally, an initialization scheme is presented to allow background initialization in scenarios with high volume of traffic. The key feature of body height estimation with a single omnidirectional camera is the estimated position and orientation of the camera relative to the ground. Position and orientation of the camera vary for each parking scenario and must automatically be estimated from image data only. Therefore, a novel, model-based camera-ground function is introduced that estimates the orientation of the camera relative to the ground. Estimation is based on image data captured from approaching drivers. The function explicitly considers tilt caused by inclined parked cars and has a global minimum when the estimated camera orientation best matches real camera orientation. This method was first proposed in the field of omnidirectional cameras and was also first used in the domain of automotive system engineering.

Chapter 4 presents a method to generated 3D-ambiance information with a single omnidirectional camera and motion-stereo. The control unit of the smart door requires ambiance information to avoid collisions by computing situation-dependent opening paths. The key problem addressed in this chapter is the generation of solid 3D-ambiance information from low-textured low-resolution panoramic images.

This chapter introduces the fundamentals of stereo vision with panoramic images. It also presents a method to rectify panoramic images enabling 1D-correspondence search in pairs

of panoramic stereo images. Rectification is based on estimated camera positions provided by the mechanical device. Clearances in the mechanical device, however, cannot be detected and lead to wrongly determined camera poses. Therefore, an egomotion estimation algorithm is presented to refine the camera positions using images correspondences only.

The chapter describes the semi-global-matching stereo algorithm to generate dense disparity maps from low-textured, low-resolution panoramic images. It also introduces the generation of 3D-ambiance information using triangulation and proposes a new refinement stage to remove disturbances and outliers in 3D-data. Additionally, this chapter proposes a method to determine the position error and the accuracy of 3D-data obtained. The position error depends on the calibration and on the quantization error of the camera system. Finally, measurement ranges and dead-zones for stereo setups based on omnidirectional cameras are analyzed for common projections. In this thesis, motion-stereo-based generation of 3D-ambiance information by means of omnidirectional cameras and the algorithms to estimate position errors of 3D-data were first proposed in the field of automotive system engineering for smart car doors.

Chapter 5 summarizes the thesis by drawing a general conclusion and indicates potential research directions for future work.

2 Omnidirectional cameras

2.1 Introduction

Omnidirectional cameras consist of a perspective camera together with an imaging device composed of a mirror-lens combination. Due to this imaging device, omnidirectional cameras provide a very large field of view compared to other cameras and are, therefore, particularly suitable for surveillance or ambiance monitoring. A lot of research has been done over the last decades addressing geometry and properties of omnidirectional cameras, mathematical formalism (camera models) especially targeting omnidirectional cameras and calibration of these camera systems.

In general, camera systems that have a projection center are called central projection systems. The projection center, also called the single effective viewpoint, permits the generation of geometrically correct perspective panoramic images from images captured by an omnidirectional camera. It is also required to extract metric information about the environment from 2D-images and to apply the known theory of epipolar geometry to omnidirectional cameras. To achieve this, a camera function must be determined to project 3D-world-points into 2D-image-points on the camera sensor. This function is also called the *camera model*, and its parameters have to be determined during camera calibration.

During the calibration procedure, a planar chessboard pattern – with known geometry – is shown at several positions and orientations. Calibration images are captured by the omnidirectional camera and serve as an input to the calibration procedure. The calibration procedure computes the camera parameters and determines the camera model from the 2D-position of chessboard corners in calibration images. These chessboard corners, however, had to be selected manually in state of the art calibration procedures [26, 27]. This is very time-consuming and can lead to inaccuracies in the calibration results. Therefore, an automatic chessboard corner extraction algorithm would be highly desirable both to automate the calibration procedure to be used in the automotive domain and to improve the calibration results. In this chapter, an algorithm is proposed that automatically extracts chessboard corners in calibration images and that is strongly robust under various illumination conditions.

Original images from omnidirectional cameras are highly distorted and cannot simply be interpreted for normal image processing routines. For example, straight borders of real, rectangular objects might be projected as curves in original images. For this reason, conventional image processing algorithms such as Hough-transformation are no longer suitable to be run directly on images from omnidirectional cameras. Although methods and procedures exist to process original images even from uncalibrated omnidirectional cameras, it is hard to obtain useful ob-

ject properties like width and size from the images. A common way to overcome this limitation is to transform original images into panoramic images. The transformation process projects the intensity values of each sensor pixel onto a new pixel position on the panoramic image plane. This process is also called *image rectification* and requires a calibrated omnidirectional camera model. Besides spherical, conic and plane projections, the most common projection used to transform original images into panoramic images is the cylindric projection, which is mainly used for applications in robotics. However, there has been no evaluation in literature that studies different projections in terms of best utilization of intensity values from sensor pixels projected onto pixel positions in panoramic images. For this reason, this thesis proposes a new value – the *pixel density* – as a measurement value for comparing different projections in terms of best utilization of sensor pixels in panoramic images. In this manner, best utilization of sensor pixels and, hence, best resolution in panoramic images can be obtained for any camera configuration.

This chapter describes the geometrical and the mathematical principles of omnidirectional cameras, which are prerequisites for image-based body-height estimation and stereo-based ambiance monitoring. For both applications, a known camera model and, hence, camera calibration along with image rectification are prerequisites to extract object properties like body heights of approaching drivers or metric, 3D ambiance information from an omnidirectional camera.

The outline of this chapter is as follows: The single point of view theorem, the geometry of omnidirectional cameras and the camera model used are described in Section 2.2. Section 2.3 illustrates the calibration procedure and describes the automatic chessboard corner extraction algorithm. Section 2.4 introduces methods to transform original images into panoramic images. The pixel density as a new measurement value for evaluating projections in terms of best utilization of sensor pixels in rectified images is proposed in Section 2.5, and experiments and results are presented in Section 2.6. This chapter ends with a discussion and conclusion in Section 2.7.

2.2 Geometry of omnidirectional cameras

2.2.1 The single point of view property

This section summarizes the geometry of omnidirectional cameras and the single point of view theorem. Very nice descriptions of the geometry of omnidirectional cameras and the single point of view theorem were provided by Barreto *et al.* [28], Micusik [29] and by Scaramuzza [30] and are briefly presented below.

A camera has a *single point of view* (SPOV) (i.e. projection center) and is called a *central projection system* if the light rays from 3D-scene points meet in a single point [30, 31]. Perspective cameras are examples of central projection systems and project points from a 3D-scene into points on a 2D-image plane (see Figure 2.1(a)). The projection is linear and can be described with a 4×4 projection matrix $\mathbf{P}$ using homogeneous coordinates. It is also known as the *pin-hole model* of perspective cameras [32, 33, 34, 35] and can be modeled by a bundle of rays passing 3D-scene points, the single point of view and intersect the image plane. Imaging devices can be

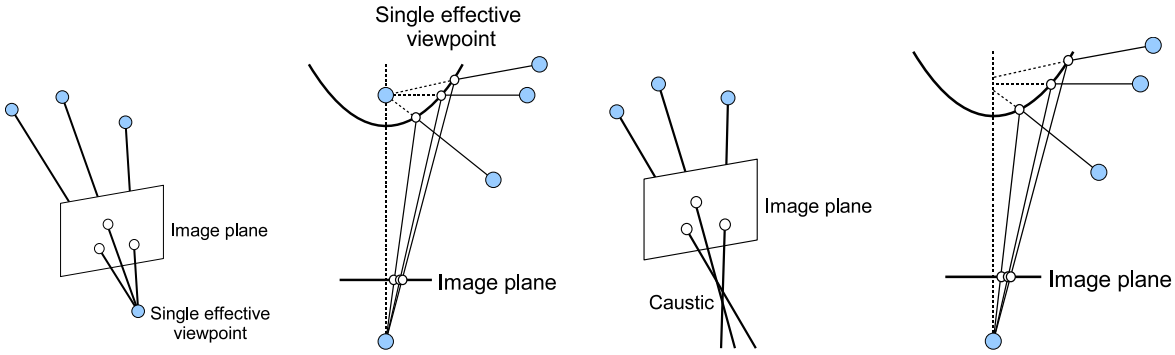
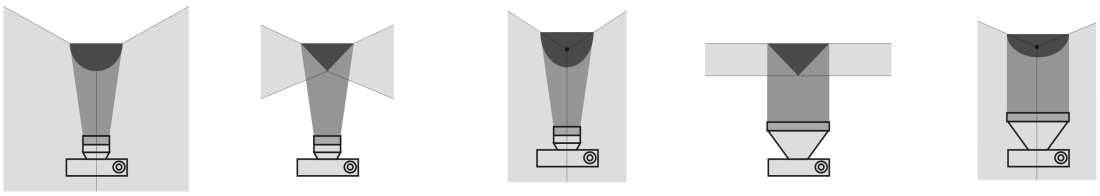


Figure 2.1: Examples of cameras with a single point of view (SPOV) (a,b), and examples of cameras without a single point of view (c,d) [30].

used to enhance the horizontal and the vertical field of view of conventional perspective cameras. However, some of these systems are also central projection systems but their projections cannot be modeled by a pin-hole model due to their very high distortions. A *central omnidirectional camera* is such a projection system and is also called a *catadioptric camera* due to its imaging device consisting of mirrors and lenses (see Figure 2.1(b)). Many vision systems do not have a single point of view. Instead, a locus of viewpoints is formed and the vision system is called a *non-central camera system*. Figure 2.1 illustrates examples of non-central camera systems both for perspective cameras (see Figure 2.1(c)) and for omnidirectional cameras (see Figure 2.1(d)). In general, it is rather difficult to manufacture precise central projection systems due to small construction errors and misalignments of lenses and mirrors. For these reasons, Micusik [29] shows that the calibration techniques developed for central projection systems are also suitable for non central projection systems, in particular for omnidirectional cameras . These calibration techniques estimate a potential location of an effective single point of view. Previous work related to non central cameras may be found in [36, 37, 38, 39].

A single effective viewpoint permits the generation of geometrically correct panoramic images from images captured by an omnidirectional camera. In other words, geometric properties such as object height or object size can easily be obtained from panoramic images. The single effective viewpoint is also a prerequisite for applying the known theory of epipolar geometry (e.g., see [32]) to omnidirectional cameras. The theory of epipolar geometry developed for perspective cameras can be easily adapted to omnidirectional cameras to perform ego-motion estimation and to obtain structure from motion from panoramic images. In this thesis, ego-motion estimation and structure from motion are used to refine estimated camera poses and to obtain 3D ambiance information from the surroundings in the proximity of the car door.

The transformation of original images into perspectively correct panoramic images is possible with the help of the single point of view constraint since every pixel in an original image measures the intensity of light passing through the viewpoint in one particular direction. If the camera is calibrated and if the camera model of the omnidirectional camera is available, then the 3D-direction of each light ray can be precomputed for each sensor pixel. The intensity values, which are related to particular light rays and measured by the sensor pixels, can then directly be mapped onto a projection plane to form panoramic images.



(a) (b) (c) (d) (e)

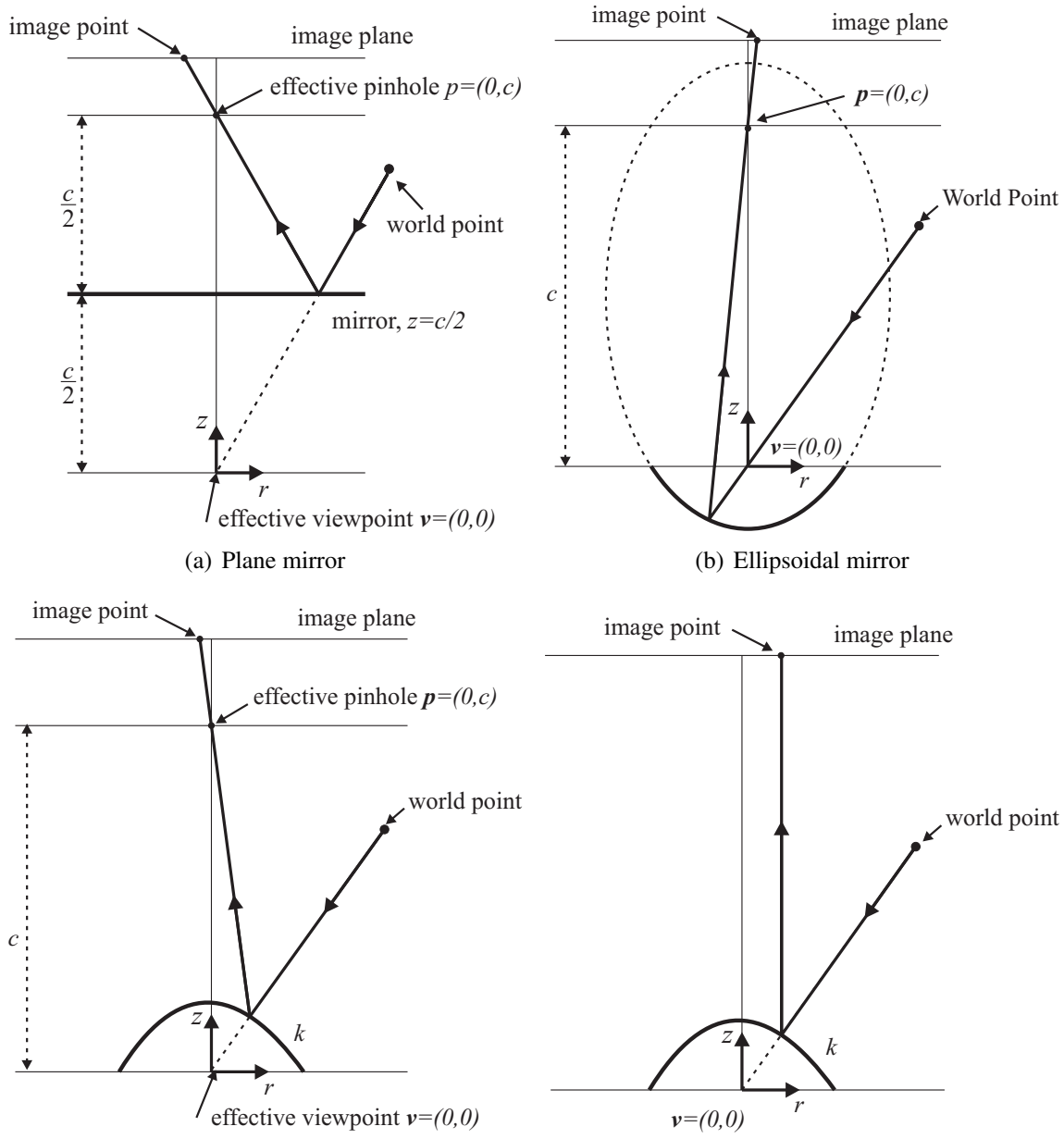
Figure 2.2: Imaging devices for omnidirectional cameras: spherical mirror (a), conical mirror (b), hyperbolic mirror (c), conical mirror (d) and parabolic mirror with telecentric lens (e) [14].

2.2.2 Geometry of omnidirectional cameras

René Descartes [40] presented the concept of central catadioptric cameras. This concept already appeared in Descartes' treatise Discours de la Methode *Discours de la Methode* in 1637 and describes the phenomenon that reflective as well as refractive ovals focus light into a single point when they are illuminated from another properly chosen point. Later, Feynman *et al.*[41] extended this idea in 1963 and Rees [42] successfully applied an omnidirectional camera system for a patent in 1970 for military applications. Ishiguro *et al.* [43] gave an overview of omnidirectional cameras and compared them in terms of mirror types, field of view and in terms of manufacturing in 1998. Figure 2.2 illustrates common omnidirectional cameras described by Ishiguro *et al.*. Figures 2.2(a), 2.2(b) and Figure 2.2(c) illustrate omnidirectional cameras that consist of a spherical, a conical and a hyperbolic mirror along with a conventional perspective camera. By contrast, Figures 2.2(d) and Figure 2.2(e) illustrate omnidirectional cameras with special mirrors. Omnidirectional cameras with such mirrors require the use of telecentric lenses.

At the same time, Nayar and Baker [31, 44, 45] made the concept of omnidirectional cameras popular in a general mathematical formalism and introduced it to the computer vision community in 1998. Their work describes the mathematics for the complete class of omnidirectional cameras that have a single effective viewpoint. These cameras can be constructed with a single mirror and a conventional perspective camera assuming a pinhole camera model. The important contribution of their work is the derivation of the complete class of omnidirectional cameras with a single effective viewpoint using only two parameters. These parameters are the distance c between the projection center of the perspective camera ($\mathbf{p} = (\mathbf{0}, c)$) and the effective viewpoint ($\mathbf{v} = (\mathbf{0}, 0)$) and parameter k (also called *mirror constant*). Parameter k represents the curvature of mirrors used for an omnidirectional camera (see Figure 2.3). Four potential camera- and mirror-configurations within that class are feasible for constructing central projection cameras, but two configurations are not useful for camera design (degenerated configurations). These four configurations combine a perspective camera together with a planar mirror (see Figure 2.3(a)), with an ellipsoidal mirror (see Figure 2.3(b)), with a hyperbolic mirror (see Figure 2.3(c)) and an orthographic camera with a paraboloidal mirror (see Figure 2.3(d)).

There exist two degenerated configurations that also fulfill the single point of view constraint. These configurations combine a perspective camera with a conical mirror (see Figure 2.4(a))



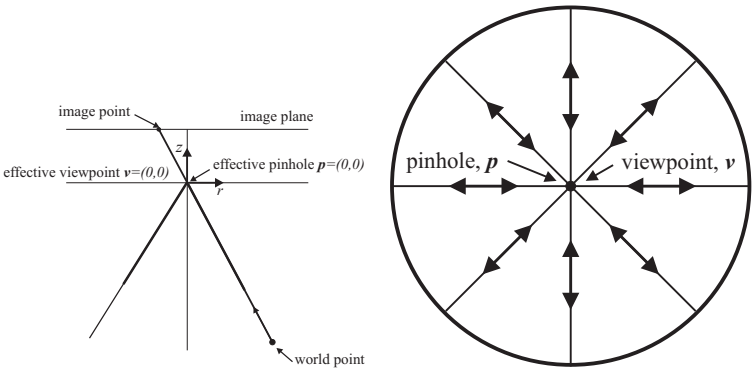
(a) Plane mirror (b) Ellipsoidal mirror

(c) Hyperbolic mirror (d) Parabolic mirror and telecentric lens

Figure 2.3: Non-degenerated omnidirectional cameras with a single point of view [31].

and with a spherical mirror (see Figure 2.4(b)). However, the degenerated configurations are not useful for constructing omnidirectional cameras. In the first case, the perspective camera has to be placed into the center of the sphere meaning that the camera would see only itself. To obtain the single point of view property for the second case, the shape of the conical mirror would be the limit of the field of view of the perspective camera and the top of the cone must be in the projection center of the camera. The single point of view property might be satisfied for this case but the camera would see nothing.

Following Eq. 2.1 and Eq. 2.2, Nayar *et al.* proposed a general, algebraical constraint for all omnidirectional cameras with a single point of view. This constraint depends only on the mirror



(a) Conical mirror (b) Spherical mirror

Figure 2.4: Degenerated omnidirectional cameras with single point of view [31].

characteristic k and on the distance c between the projection center and the single point of view.

$$\left(z-\frac{c}{2}\right)^2 - r^2\left(\frac{k}{2}-1\right) = \frac{c^2}{4}\left(\frac{k-2}{k}\right) \quad (k \leq 2) \tag{2.1}$$

$$\left(z-\frac{c}{2}\right)^2 + r^2\left(1+\frac{k}{2}\right) = \left(\frac{2k+c^2}{4}\right) \quad (k > 0) \tag{2.2}$$

where $r = \sqrt{x^2+y^2}$. Thus, the world coordinate system is located within the projection center of the mirror. Nayar *et al* proposed solutions for several mirror types by choosing different camera constants k and c. However, the largest vertical field of view is obtained with a hyperbolic mirror using the camera constants $k > 2$ and $c > 0$. For this configuration, Eq. 2.1 becomes

$$\frac{1}{{a_h}^2}\left(z-\frac{c}{2}\right)^2 - \frac{1}{{b_h}^2} = 1 \quad \text{with} \quad a_h = \frac{c}{2}\sqrt{\frac{k-2}{k}}, \quad b_h = \frac{c}{2}\sqrt{\frac{2}{k}}. \tag{2.3}$$

In this thesis, the constraint is utilized to design an omnidirectional camera that is based on a hyperbolic mirror and that has a single point of view. In general, cameras designed in this manner have only a pseudo single point of view due to misalignments and inaccuracies that occurred during manufacturing. However, the camera model and the calibration process described in the next sections are based on this important constraint and have proved to be feasible for cameras with a pseudo single point of view.

2.2.3 Camera model

In this section, the camera model of an omnidirectional camera is presented. It describes a mathematical relation between 3D-world points of a 3D-scene and their projection onto the 2D-image plane (sensor plane). The camera model is also prerequisite for transforming original images into panoramic images. The camera model of a standard perspective camera can be described following Eq. 2.4.

$$\lambda \mathbf{x} = \mathbf{P} \cdot \mathbf{X}. \tag{2.4}$$

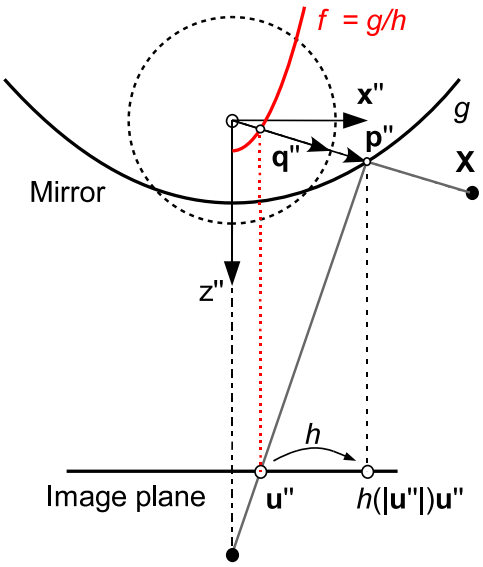


Figure 2.5: Omnidirectional camera model proposed by Micusik [29].

Thereby, $\mathbf{X} = [X, Y, Z, 1]$ represents the homogeneous coordinates of a 3D-scene point, whereas vector $\mathbf{x} = [x, y, 1]$ describes the normalized coordinates of an image point $\mathbf{x}$. Furthermore, let $\mathbf{P} \in \mathbb{R}^{4\times3}$ be a projection matrix so that $\mathbf{P} = [\mathbf{R}|\mathbf{T}]$. The translation $\mathbf{T}$ and the rotation $\mathbf{R}$ express a relation between the world frame and the camera reference frame. This projection can be understood as a mapping of 3D-scene points $\mathbf{X}$ into an image point $\mathbf{x}$ on the sensor plane using a straight line connecting both the 3D-scene points and the projection on the sensor plane passing through the projection center [29]. In other words, all 3D-scene points located on this line are projected into the same image point on the sensor plane. With this representation, image points on the sensor plane are projections of 3D-scene points that can be located in front or behind the camera.

In contrast to conventional perspective cameras, omnidirectional cameras project 3D-scene points located in front of the camera into certain image points and 3D-scene points located behind the camera into others. Thus, Micusik [29] introduced a formalism to represent image points of omnidirectional cameras as a set of unit vectors in $\mathbb{R}^3$ using a unified spherical camera model. He derived the projection function for omnidirectional cameras from Eq. 2.4 as follows.

$$\lambda\mathbf{q} = \mathbf{P} \cdot \mathbf{X}, \lambda > 0, \tag{2.5}$$

where $\mathbf{q} = [x\ y\ z]$ is a unit vector (i.e. $\|q\| = 1$) representing the direction of an image point. Imagine a 3D-scene point $\mathbf{X}$ that is observed by an omnidirectional camera (see Figure 2.5). Assuming the camera model proposed by Micusik [29], a vector $\mathbf{p}'' = (\mathbf{x}''^T, z'')$ describing the direction to the point $\mathbf{X}$ can be found that has the same direction as the unit vector $\mathbf{q}$. The projection of vector $\mathbf{p}''$ is mapped into an image point $\mathbf{u}''$ on a virtual sensor plane so that $\mathbf{u}''$ is collinear with $\mathbf{x}''$ (see Figure 2.5). In other words, the camera maps vector $\mathbf{p}''$ into an image point $\mathbf{u}''$ using two transformation functions g and h. Following Eq. 2.6, Micusik [29]

introduced a formalism to describe this projection:

$$\lambda \mathbf{p}'' = \begin{bmatrix} h(|\mathbf{u}''|)\mathbf{u}'' \\ g(|\mathbf{u}''|) \end{bmatrix} \tag{2.6}$$

Thereby, the two functions $g,\ h\ \in \mathbb{R} \rightarrow \mathbb{R}$ are introduced to project vector $\mathbf{p}''$ to $\mathbf{u}''$. These functions depend on the mirror type (e.g. parabolic, hyperbolic mirror) and on the mirror shape. The function g can be seen as a function to describe the mirror profile and h as a function to orthographically project vector $\mathbf{p}''$ into a point $h(\|\mathbf{u}''\|)\mathbf{u}''$ on the sensor plane. Scaramuzza modified this camera model to the effect that he chooses one function $f = g/h$ instead of two distinctive functions $g,\ h$. This allows to set function h to unity and to facilitate Eq. 2.6 to Eq. 2.7.

$$\lambda \mathbf{p}'' = \lambda \begin{bmatrix} \mathbf{u}'' \\ f(|\mathbf{u}''|) \end{bmatrix} = \mathbf{P} \cdot \mathbf{X} \tag{2.7}$$

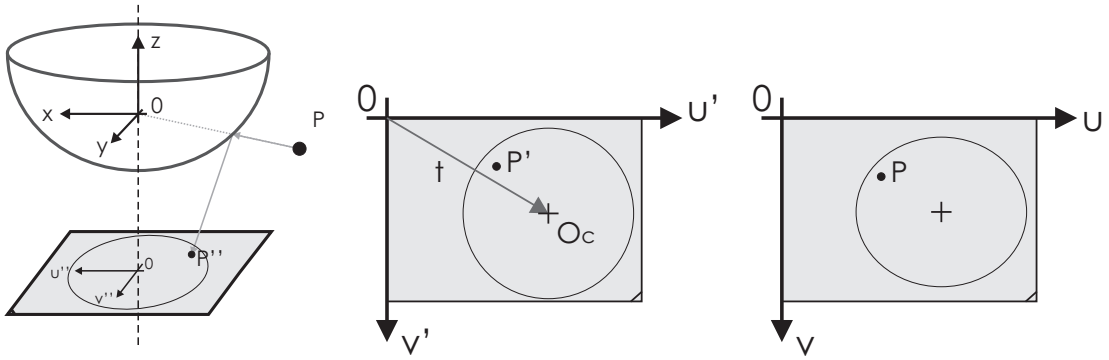
However, three important assumptions are made and must be met when dealing with omnidirectional cameras using the camera model described by Micusik [29]:
i): The mirror is about approximately rotationally symmetric with respect to its axis.
ii): The mirror is perfectly aligned to a virtual sensor plane so that its z-axis is perpendicular to the sensor plane.
iii): The omnidirectional camera is assumed to be a central projection camera meeting the single point of view theorem.

2.2.4 Scaramuzza's representation of omnidirectional cameras

Besides other camera models, Scaramuzza [30] proposed a unified Taylor model to derive the projection function that maps 3D-scene points onto a 2D-image plane and is briefly presented below. The origin of the camera-coordinate system is also the origin of the world-coordinate system and is located in the single point of view and, hence, in the projection center of the mirror. 3D-scene points are projected onto a virtual, ideally aligned sensor plane. The image points from the ideal sensor plane are mapped onto the real sensor plane using affine transformations.

Figure 2.6(a) illustrates the projection of a 3D-scene point P'' onto the ideal sensor plane E''. The ideal sensor plane E'' has the coordinates u'', v'' and is assumed to be perfectly aligned to the mirror. For this reason, the projection center of the mirror and the origin of the camera coordinate system u'', v'' are assumed to be coincident with the image center O_c on the ideal sensor plane. Additionally, the boundary of the mirror is mapped as a circle onto the ideal plane. The advantage of introducing an ideal sensor plane along with a coordinate system that is identical with the mirror coordinate system is the direct consideration of lens distortions within the Taylor model.

An affine transformation maps points from the ideal sensor plane to the real sensor plane. A translation t transforms the projection center O_c from the image center of the ideal sensor plane into the origin of an additional, virtual sensor plane E' with its coordinates u', v' (see Figure 2.6(b)). Finally, an affine transformation $\mathbf{A}$ takes the misalignments of the mirror into



(a) World point P'' mapped onto an ideal sensor plane E''.

(b) Point P' on the virtual sensor plane.

(c) World point P mapped onto the real sensor plane.

Figure 2.6: The camera model proposed by Scaramuzza [30]: A scene point P is mapped onto an ideal, virtual sensor plane E'' (a). Its projection on a virtual sensor plane E' (b) is projected onto a real sensor plane (c) using affine transformations.

account and projects the image points from the virtual sensor plane E' onto the real sensor plane E (see Figure 2.6(c)).

The projection of a 3D-scene point into an image point P on the 2D-sensor plane E can be described as follows.

$$\vec{P} = \begin{bmatrix} u_P \\ v_P \end{bmatrix} = f(\vec{p}) \quad \text{with} \quad \vec{p} = \lambda \cdot \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix}, \lambda > 0 \tag{2.8}$$

Eq. 2.8 expresses a relation between vector $\vec{p}$ in world coordinates x_P, y_P, z_P and the camera coordinates u_P and v_P. Moreover, a scale factor λ is introduced to determine all 3D-scene points in world coordinates along the direction of vector $\vec{p}$. These 3D-points are all projected onto the same sensor pixel u, v. Since the camera coordinates on the virtual sensor plane E'' are identical with the $x,\ y$ coordinates of the world coordinates, Eq. 2.8 can be rewritten following Eq.2.9:

$$\vec{p} = \begin{bmatrix} x_{\vec{p}} \\ y_{\vec{p}} \\ z_{\vec{p}} \end{bmatrix} = \lambda \cdot \begin{bmatrix} u_{P''} \\ v_{P''} \\ f(\rho) \end{bmatrix} \quad \text{with} \quad \rho = \sqrt{u_{P''}^2 + v_{P''}^2}. \tag{2.9}$$

Scaramuzza [30] uses a polynomial function $f(\rho)$ to model the mirror characteristic for various omnidirectional cameras (see Eq. 2.10). This function specifies the world coordinate z_P depending on $u_{P''}$, $v_{P''}$ and, hence, on the world coordinates x_P and y_P.

$$f(\rho) = a_0 + a_1\rho + a_2\rho^2 + ... + a_N\rho^N \tag{2.10}$$

The mirror shape is assumed to be rotationally symmetric and parallel to the ideal sensor plane E'' at $p'' = 0$: Therefore, the derivative $\frac{df}{d\rho}\rho^2\Big|_{p''=0} = 0$ at $p'' = 0$ must be zero and coefficient

a_1 can be set to zero.

$$\vec{p} = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \lambda \cdot \begin{bmatrix} u_{P''} \\ v_{P''} \\ f(\rho) \end{bmatrix} = \begin{bmatrix} x_p \\ y_p \\ a_0 + a_2\rho^2 + \ldots + a_N\rho^N \end{bmatrix} \tag{2.11}$$

where $\rho = \sqrt{u_{P''}^2 + v_{P''}^2}$. Variable λ can be set to unity since only the direction of vector $\vec{p}$ is of interest. As mentioned above, an affine transformation is proposed to map an image point P'' from the ideal sensor plane E'' onto the real sensor plane E. Following Eq. 2.12, the virtual sensor plane E' results from the real sensor plane E using an affine transformation.

$$\vec{P'} = \begin{bmatrix} u_{P'} \\ v_{P'} \end{bmatrix} = \mathbf{A} \cdot \vec{P} \quad \text{with} \quad \mathbf{A} = \begin{bmatrix} c & d \\ d & 1 \end{bmatrix} \quad \text{and} \quad \vec{P} = \begin{bmatrix} u_P \\ v_P \end{bmatrix} \tag{2.12}$$

The ideal sensor plane E'' results from a translation of the virtual sensor plane E' into the image center and into the projection center of the mirror so that both coordinate systems are coincident (see Eq. 2.13).

$$\vec{P''} = \begin{bmatrix} u_{P''} \\ v_{P''} \end{bmatrix} = \vec{P'} + \vec{t} \quad \text{with} \quad \vec{t} = \begin{bmatrix} u_{center} \\ v_{center} \end{bmatrix} \tag{2.13}$$

The projection of a 3D-scene point P on the real sensor plane E to the corresponding point P'' on the virtual sensor plane E'' is presented in Eq. 2.14.

$$\vec{P''} = \mathbf{A} \cdot \begin{bmatrix} u_P \\ v_P \end{bmatrix} + \vec{t} \quad \text{with} \quad \mathbf{A} = \begin{bmatrix} c & d \\ d & 1 \end{bmatrix} \quad \text{and} \quad \vec{t} = \begin{bmatrix} u_{center} \\ v_{center} \end{bmatrix} \tag{2.14}$$

2.3 Calibration

In the last section, the camera model used in this thesis was introduced. The camera model is required to transform original images into panoramic images and is a prerequisite for extracting metric information from objects in panoramic images. The parameters for the camera model, in particular the coefficients of the mirror function f and the parameters of the affine transformation have to be determined during *camera calibration*. During the calibration procedure, a planar chessboard pattern – with known geometry – is shown at several positions and orientations and calibration images are captured. These calibration images serve as an input to the calibration procedure to determine the camera parameters. In actual calibration procedures, chessboard corners have to be selected manually for each calibration image. Manual selection of chessboard corners is very time-consuming and may lead to inaccuracies in the calibration results. For these reasons, automatic chessboard corner extraction is highly desirable both to improve the calibration results (see Section 2.6.1) and to automatically perform calibration for omnidirectional cameras in the automotive domain. In the next section, an algorithm is proposed to extract chessboard corners in calibration images captured by omnidirectional cameras. Thereafter, an efficient calibration algorithm to determine the camera parameters proposed by Scaramuzza is presented in Section 2.3.2.

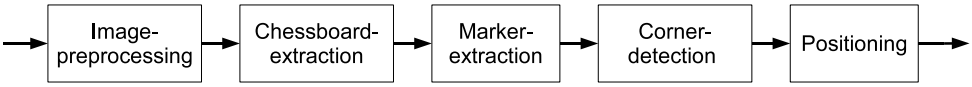


Figure 2.7: Block diagram of the chessboard corner extraction algorithm.

2.3.1 Automatic chessboard corner extraction

Figure 2.7 provides a high-level overview of the proposed chessboard corner extraction procedure. The locations of chessboard corners in calibration images are extracted during calibration and function as basis for estimating the camera parameters. Similarly to camera calibration for perspective cameras, the order of the extracted chessboard corners must be identical for each calibration image. Markers are used to guarantee an identical corner order (see Figure 2.8) for all calibration images. Commonly used chessboard patterns, which have one modified border, are not suitable for omnidirectional cameras due to ambiguities in determining the corner order. In other words, the orientation of the calibration pattern seems to be identical in Figure 2.8(b) and Figure 2.8(c), but the chessboard in Figure 2.8(c) has a different orientation compared to the chessboard presented in Figure 2.8(b).

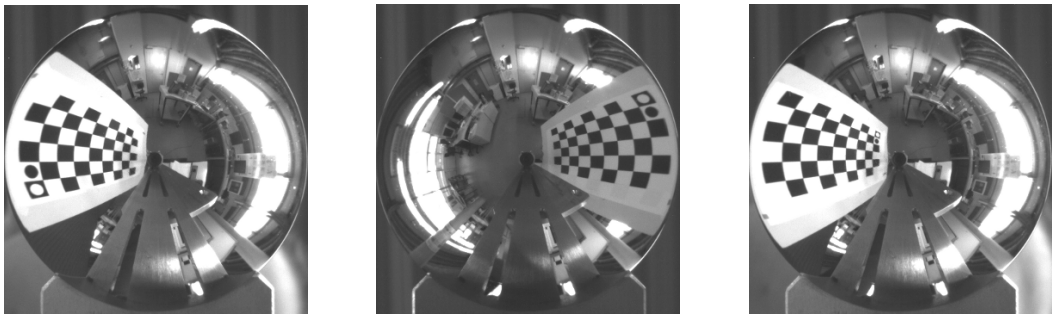
Image preprocessing

To enable camera calibration for a wide range of applications, it is highly desirable to record calibration images under different illumination conditions. This may lead to perfectly illuminated calibration images, but also to calibration images captured under very bright or very dark illumination conditions. However, standard chessboard corner extraction algorithms are tuned to perfectly extracting chessboard corners under dedicated illumination conditions. Different illumination conditions or illumination changes in calibration images may lead those algorithms to fail. Therefore, image preprocessing is necessary to reduce the influence of illumination for a large number of scenarios. Experiments demonstrated that best chessboard detection results are obtained for calibration images whose mean intensity over all image pixels is located within an intensity range between 50 and 90. For this reason, the first step in the calibration procedure is to adjust the brightness of calibration images to a mean intensity range between 50 and 90.

In literature, *gamma correction* [46] is feasible to adjust the brightness of input images in order to adjust several input images to the same intensity space. Gamma correction is a nonlinear operation and is used to code and decode luminance values in video or image systems. In its simplest case, it is a point operation to adjust the brightness of different input images to the same intensity space. Eq. 2.15 describes the transformation function.

$$I_{out} = f(I_{in}, \gamma) = \left(\frac{I_{in}}{I_{max}}\right)^{\gamma} \cdot I_{max} \tag{2.15}$$

Here, I_{in} represents the intensity of an image pixel and I_{max} represents the maximum intensity over all image pixels. The brightness of an image is estimated by computing the average of all intensity values over all image pixels. Thereafter, an adaptation algorithm checks whether the average is located in the optimum range: If not, γ is automatically modified so that the average



(a) (b) (c)

Figure 2.8: Calibration pattern proposed for camera calibration (a): Commonly used calibration pattern with one modified border cannot be used due to ambiguities in their orientation (b,c).

of all intensity values is located within the optimal range. Beginning at unity, the algorithm modifies the brightness by increasing γ (for bright images) or decreasing γ (for dark images) step by step.

Adapting the brightness is the first stage for image preprocessing. However, the contrast of some images may still be poor and may lead to difficulties in extracting the region containing the chessboard pattern. For this reason, contrast enhancement is performed as a next stage to increase the robustness of chessboard region extraction. Contrast enhancement is realized with a *histogram equalization* algorithm. Histogram equalization arranges pixel values located within a small range in a histogram (see Figure 2.9(a)) so that the complete range of intensities is used (see Figure 2.9(b)). Therefore, the *probability density function* (p.d.f.) (see Eq. 2.17) and its corresponding *cumulative distribution function* (c.d.f) are computed as a first step.

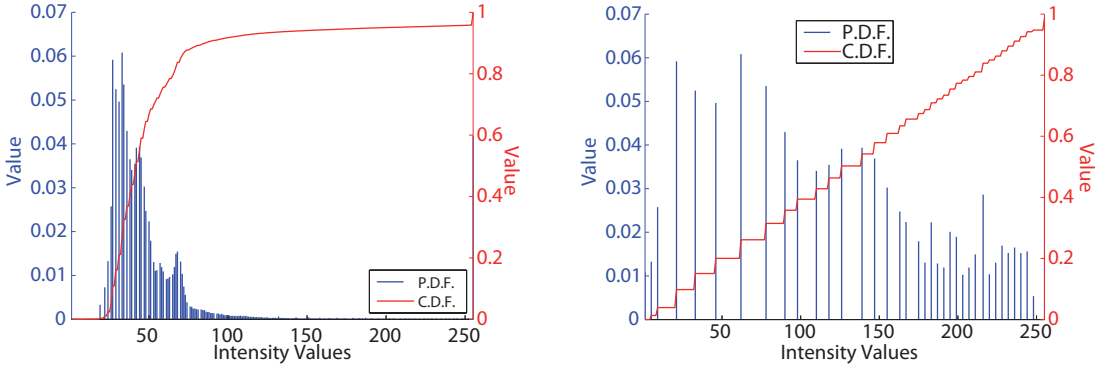
$$p_I(i) = \frac{n_i}{n}, \quad 0 \leq i < N \tag{2.16}$$

$$P_I(i) = \sum_{j=0}^{i} p_I(i), \quad 0 \leq i < N \quad \text{with} \quad \sum^{i} p_I(i) = 1 \tag{2.17}$$

Histogram equalization is performed by defining a transformation $I_{out} = T(I_{in})$ to linearize the non-linear cumulative distribution function (see Figure 2.9(a)) over the whole image histogram (see Figure 2.9(b)). This enhances the contrast in calibration images and leads to a more robust extraction of chessboard regions in calibration images.

Detection of chessboard regions

The first stage for chessboard corner extraction is in finding the region in which the chessboard pattern might be located. The rectangular chessboard patterns help to determine such regions, but calibration images may also contain rectangular objects from the surroundings that may be wrongly classified as true chessboard regions. For this reason, determining the region that contains rectangles and distinguishing whether the region belongs to the calibration pattern



(a) (b)

Figure 2.9: Histogram of calibration images after brightness adaptation (a). Enhancing the contrast in calibration images using histogram equalization leads to a better extraction of chessboard pattern in calibration images (b).

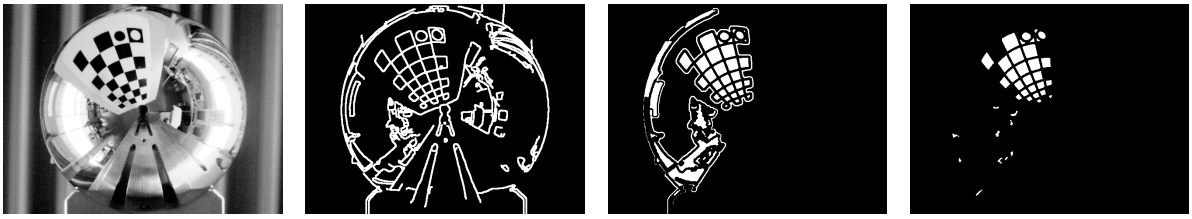
or to background is important for robust chessboard corner detection. A common way for extracting chessboard patterns is to extract image regions with high contrast. Such regions could be determined using static or dynamic thresholds. However, static or dynamic thresholding over the whole image may lead to wrong binarization results due to variously illuminated regions. Subdividing the calibration images into regions of interest and thresholding each region using certain threshold values would help to better extract the chessboard in calibration images. For this reason, the first step of chessboard corner detection is to find regions of interest in which the calibration pattern may be located.

The region of interest, which contains the chessboard, in a calibration image (see Figure 2.10(a)) is initially determined using a Canny edge detector [47] (see Figure 2.10(b)). Morphological operations reduce the large number of potential regions to a few (see Figure 2.10(c)). The region that may contain the calibration pattern is assumed to be the region with the largest number of rectangles. Figure 2.10(d) illustrates the final extraction result of rectangles from the calibration pattern. Despite image distortion, these rectangles are similar to squares and could be distinguished from other rectangles related to the background as follows: A metric m_1 is introduced to distinguish between squares from the calibration pattern or from rectangles related to the environment. Following Eq. 2.18, a rectangle belongs to the calibration pattern if m_1 is less than a chosen threshold th_{metric}. Hereby, s represents the size of an extracted rectangle and l represents its boundary.

$$m_1 = \left(\frac{s}{s^*}\right)^2 \quad \text{with} \quad s^* = \left(\frac{l}{4}\right)^2 \tag{2.18}$$

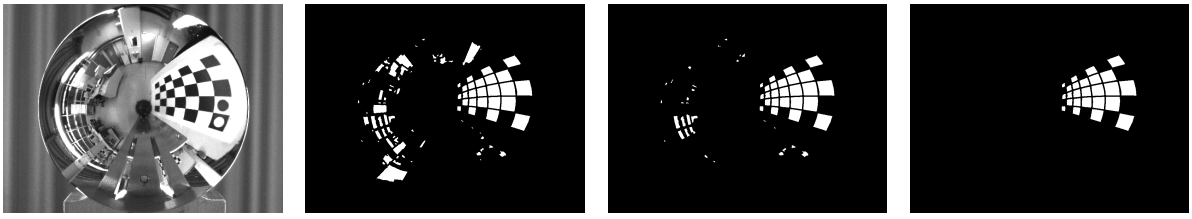
Detected rectangles that are similar to squares result in small values of metric m_1 whereas other rectangles result in high values for m_1. For this reason, metric m_1 can distinguish between rectangles from a calibration pattern and from rectangles related to the environment.

Figure 2.10(d) illustrates the successful detection of chessboard squares, but a large number of squares induced by the environment is still detected. Figure 2.11(a) shows an example of a highly structured environment that contains many rectangular objects. The resulting detection



(a) (b) (c) (d)

Figure 2.10: Calibration image (a) and detected edges (b). Potential chessboard regions (c) and validated chessboard squares along with noise (d).



(a) (b) (c) (d)

Figure 2.11: Difficult extraction of chessboard squares: The calibration image contains many rectangular objects besides the chessboard pattern (a) and the resulting detection result (b). Elimination of wrongly classified chessboard squares using the metric m_1 (c) and the valid chessboard region after the refinement stage using the metric m_2 (d).

of rectangles within a complex region is illustrated in Figure 2.11(b). In such regions, extraction of a valid chessboard region is very difficult: Therefore, a similarity m_2 is introduced that explicitly considers the similarity of sizes of chessboard squares within a certain neighborhood (see Eq. 2.19). In other words, the size of a detected square serves as a reference size and is compared with the sizes of neighboring squares. Squares that have a similar size within a certain neighborhood must relate to the chessboard pattern and are denoted as valid chessboard squares.

$$m_2 = \frac{|S_1 - S_2|}{min(S_1, S_2)} \tag{2.19}$$

Here, S_1 and S_2 represent the sizes of detected rectangles. Again, squares in a certain neighborhood of an extracted square are assumed to have a similar size if they belong to the calibration pattern. Small values for m_2 indicate a high similarity, whereas high variances are indicated by high values. Thereafter, the chessboard region can be determined by finding the region with the largest number of valid rectangles. Figure 2.11(c) illustrates the detection result after region refinement using the similarity metric m_2. Figure 2.11(d) shows the detected region that contains the chessboard pattern.

However, such extracted rectangles in a chessboard region are not suitable for further image processing since some chessboard squares might not be detected. In particular, very small squares in low resolution images close to the image center are easily deleted during the chessboard ex-

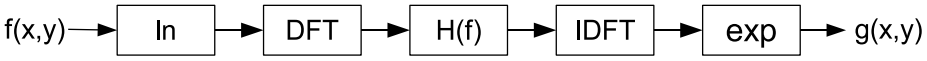


Figure 2.12: Block diagram of homomorphic high-pass filtering.

tracting process (see Figure 2.10(d)). Thus, extracted chessboard regions serve as an input for further processing stages only. Only the region that potentially contains the chessboard pattern in the calibration image is of interest for further processing stages. The remaining part of a calibration image is masked out.

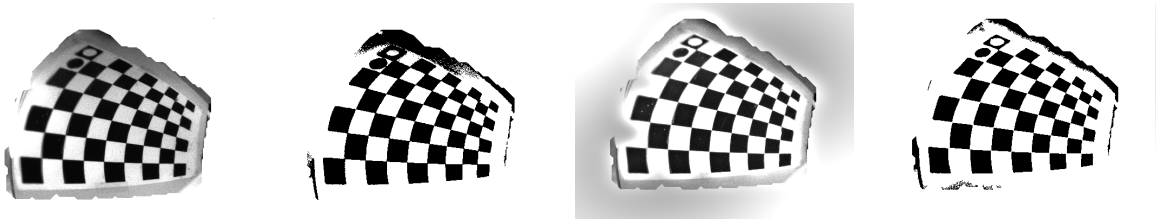
As mentioned above, image binarization is suitable for separating the chessboard from background. Binarization yields good extracting results if there is a homogeneous illumination over the entire calibration pattern. But this is difficult to realize in common calibration environments. Figure 2.13(a) illustrates an inhomogeneous illuminated chessboard and the result after binarization (see Figure 2.13(b)). It can be seen that the chessboard borders cannot be clearly separated from the rest of the calibration pattern. Homomorphic filtering [48, 49] of the input image overcomes this limitation by improving the image quality in order to facilitate object extraction. For this purpose, variations in illuminations are assumed to be multiplicative noise in the intensity domain and can, hence, be reduced by filtering in the logarithmic domain. The transformation of an image from the intensity domain into the logarithmic domain and a transformation of the result into the frequency domain using the Discrete Fourier Transformation (DFT, $\Im$) allows to make multiplicative image components such as noise additive. Consequently, image components can be easily separated in the frequency domain (see Eq. 2.21). Here, $l(x, y)$ and $h(x, y)$ represent the high- and the low-frequency parts of a calibration image. A general overview of the linear filter process is illustrated in Figure 2.12.

$$\begin{aligned} f(x,y) &= l(x,y) \cdot h(x,y) \\ \Im(ln(f(x,y))) &= \Im(ln(l(x,y))) + \Im(ln(h(x,y))) \\ F(u,v) &= L(u,v) + H(u,v) \end{aligned} \tag{2.20}$$

Low-frequency components in the frequency domain represent the variations of illuminations on the chessboard. To achieve a more homogeneous illumination on chessboard patterns, high frequencies are amplified and low frequencies are damped. Hence, high-pass filtering can be used to suppress low frequencies and to amplify high frequencies in logarithmic images. Eq. 2.21 presents a modified high-pass filter where ρ represents the limiting frequency and α the filter gain. A preservation factor β is introduced to avoid the suppression of very low frequencies. Very low frequencies represent the homogeneous image regions in calibration patterns such as the white and black regions and must, hence, be preserved.

$$H(f) = \begin{cases} 1, & \text{for } f \leq \beta \\ 1 - \alpha \exp\left(-\frac{|f|^2}{\rho^2}\right), & \text{otherwise} \end{cases} \tag{2.21}$$

Figure 2.13(c) shows the extracted chessboard after homomorphic high-pass filtering. It can be seen that the white area within the calibration pattern is more homogeneous after homomorphic



(a) (b) (c) (d)

Figure 2.13: Inhomogeneously illuminated chessboard pattern (a) and its binarization result (b). A more homogeneous chessboard region after homomorphic filtering (c) and the binary image after binarization (d).

filtering than the white region within the original image. This leads to a better binarization result (see Figure 2.13(d)). The results obtained can now be used for further image processing such as marker detection or chessboard corner extraction. Further morphologic operations allow a final detection of the chessboard regions.

Marker extraction

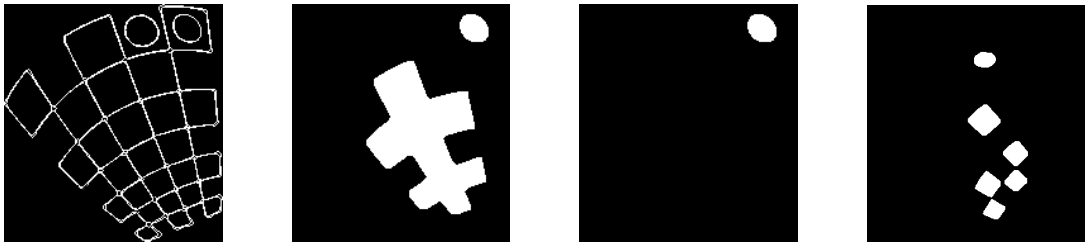
Marker extraction is the next stage for an automatic detection of chessboard corners. Markers are required to obtain a unique order of chessboard corners in calibration images. Although a detection of the black marker seems to be very easy, the proposed algorithm starts with detecting the white one. Therefore, the edge image obtained during chessboard region extraction (see Figure 2.14(a)) serves as an input for extracting the inner image parts of the calibration pattern and the white marker (see Figure 2.14(b)). As shown in Figure 2.14(c), the white marker can be separated after removing the inner chessboard regions. However, some images exist where the inner regions cannot be removed (see Figure 2.14(d)). To overcome this limitation, a circularity metric m_3 is proposed to determine circular objects by using of the object size s and the object boundary l (see Equation 2.22).

$$m_3 = \left(\frac{s}{s^*}\right)^2 \quad \text{with} \quad s^* = \frac{l^2}{4\pi} \tag{2.22}$$

Circular objects result in small values, whereas rectangular objects result in large values for m_3. In this manner, the algorithm identifies circular markers and distinguishes them from rectangular objects by the circularity metric m_3. Once the white marker is found in a calibration image, the black marker is extracted within a region of interest close to the white marker. Morphologic operations together with the metric m_3 are used to distinguish between chessboard squares and the black marker.

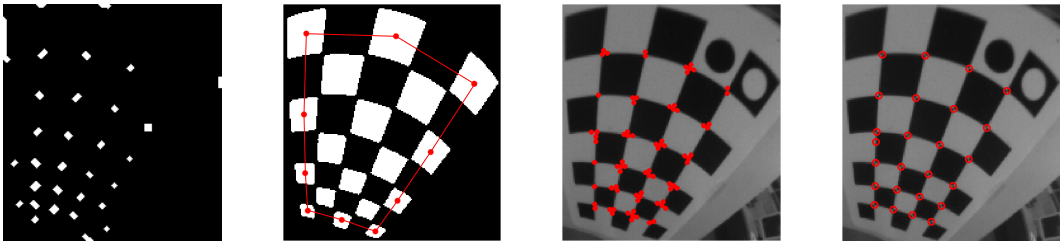
Corner detection and refinement

In this section, a method is proposed to extract chessboard corners, which are required for camera calibration. A preprocessing routine is introduced that extracts grid points on the chessboard



(a) (b) (c) (d)

Figure 2.14: Edge image used for marker extraction (a), inner region of the chessboard pattern (b) and extraction of the white marker (c). Bad detection result due to many extracted chessboard squares (d).



(a) (b) (c) (d)

Figure 2.15: Grid points detected on a chessboard pattern (a) and the region of interest to determine potential chessboard corners (b). Extracted chessboard corners (c) and the final detection result verified by a Harris corner detector (d).

pattern on which chessboard corner extraction is based. This routine increases the performance of the extraction algorithm by predicting the regions in which chessboard corners might be located. The edge image and the binary chessboard image serve as input for the preprocessing routine. The markers are removed and the intersection points of the chessboard pattern are separately extracted both for the edge image and for the binary image. The results of both images are used to determine the grid points on the calibration pattern and to close potential gaps within the grid. A mask is then generated whose dimension is limited by the center points of the outer chessboard squares (see Figure 2.15(b)). This mask reduces the number of potential grid points that have to be validated during the corner extracting process. Figure 2.15(c) illustrates potential grid points on which chessboard corners might be located.

The extracted grid points indicate the potential location of the chessboard corners. As shown in Figure 2.15(c), the grid points are not exactly located on the intersection points of the chessboard grid. For this reason, a Harris-Corner detector [50] is required to determine the final corner positions and to refine previous detection results.

The Harris corner detector is a powerful corner detector and is robust against illumination changes, image size and image orientation. The main idea of this detector is to find corners based on eigenvalues of a correlation matrix. This allows a robust detection of weak corners that appear ofte in omnidirectional calibration images due to image distortion. Figure 2.15(d)

illustrates the final detection result of potential chessboard corners obtained by the Harris corner detector. Although most of the grid corners are identified as valid intersection points, some points located somewhere else are also initially classified as valid chessboard corners. These points can be identified and can be removed using correlation-based *template matching*.

Template matching is often used in image processing to retrieve small image regions from a whole image. Matching is performed with a convolution mask, which is also called the reference template, and can be easily performed on gray-scaled or on binary images. The convolution mask is specially tailored to specific features that have to be retrieved within an image. Figure 2.17 illustrates the chosen feature mask both to identify valid points located on chessboard corners (see Figure 2.16(b), template 1) and to detect points located on the border of a chessboard square (see Figure 2.16(c), template 2).

Following Eq. 2.23, the zero means normalized cross correlation function (ZNCC) is used to determine the convolution output $c(s,t)$. Here, $f(x,y)$ describes a reference template with size $N \times N$ and its mean gray value $\bar{f}$. $g(x,y)$ describes the corresponding search template with size $N \times N$ and its mean gray value $\bar{g}$.

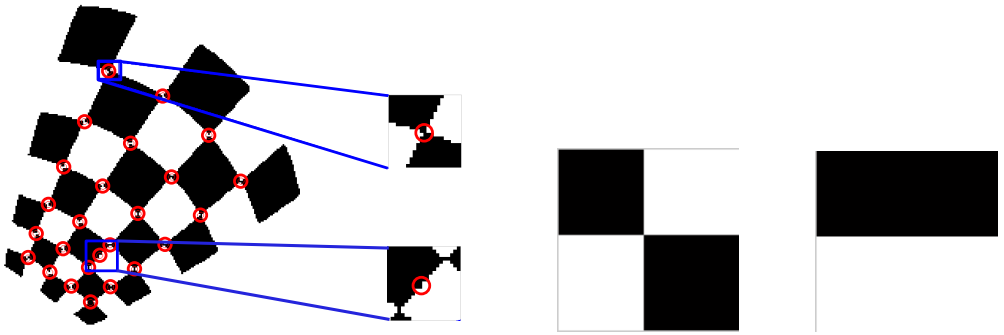
$$c(s,t) = \frac{\sum_{x=1}^{N}\sum_{y=1}^{N}\left[f(x,y) - \bar{f}_{N\times N}\right]\left[g(x-s,y-t) - \bar{g}_{N\times N}\right]}{\sqrt{\sum_{x=1}^{N}\sum_{y=1}^{N}\left[f(x,y) - \bar{f}_{N\times N}\right]^2 \sum_{x=1}^{N}\sum_{y=1}^{N}\left[g(x-s,y-t) - \bar{g}_{N\times N}\right]^2}} \tag{2.23}$$

The reference template is moved over the whole image, and the convolution output will be the highest at image positions on which the reference template best matches the search template. Figure 2.16(a) illustrates a typical corner detection result after chessboard corner extraction. There is one wrongly determined chessboard corner located on the border of a chessboard square. This wrongly determined chessboard corner can be detected by template matching. Figure 2.17(a) illustrates the convolution result for templates 1 and 2 performed on a valid chessboard corner. Figure 2.17(b) illustrates the correlation result for a wrongly detected chessboard corner located at the border of a chessboard square. Template 1 best matches positions of chessboard corners, whereas template 2 best matches on positions that contain chessboard square borders.

However, testing whether each potential corner position belongs to a valid chessboard corner is very time-consuming. Testing can be sped up by determining a likely hood of a point being located on a chessboard corner or on a chessboard border. This probability is determined by calculating the distances between a corner point and the center points of the closest neighboring squares. For a valid corner point, this distance is assumed to be approximately constant. The distances of points located at one of the square borders vary and are candidates that have to be checked for belonging to chessboard corners or to square borders.

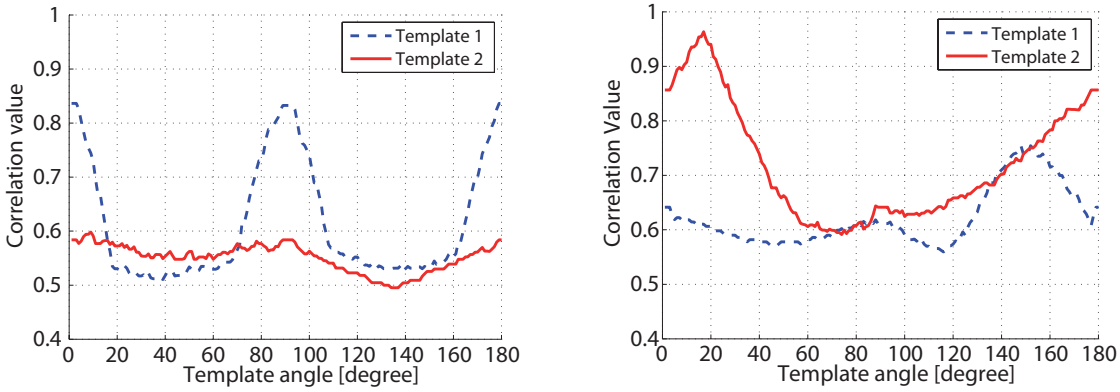
Corner positioning

The order of extracted chessboard corners must be identical in each chessboard image for the calibration algorithm. The extracted markers help to guarantee an identical corner order in each calibration image and to determine the first corner point on the chessboard.



(a) (b) Template 1 (c) Template 2

Figure 2.16: Identified chessboard corners in a calibration image (a). Reference templates to identify points located at corners (template 1) (b) and points located at borders of chessboard squares (template 2) (c).



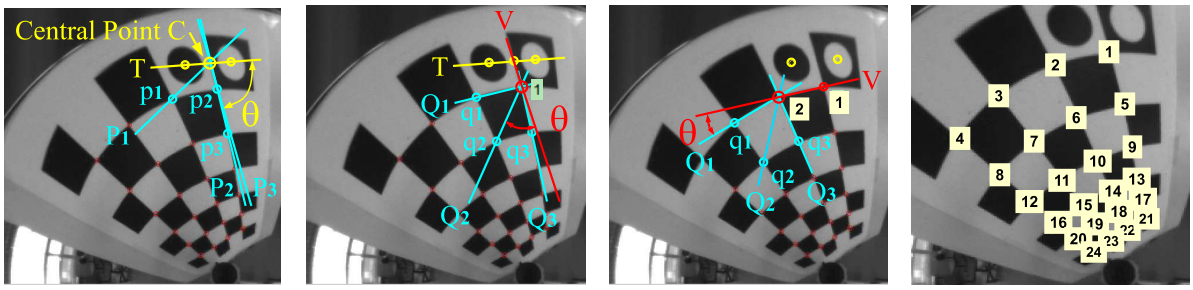
(a) (b)

Figure 2.17: Convolution output for points located at corners using template 1 (a) and for points located at borders using template 2 (b).

In a first step, the center of the two markers, the straight line T connecting these centers and the central point C bisecting this line are extracted (see Figure 2.18(a)). The first chessboard corner is the point with the shortest distance to the central point C and the one whose connecting line with C is approximately perpendicular to T. To obtain this point, the three closest chessboard corners ($p1,\ p2,\ p3$, see Figure 2.18(a)) to the central point C are determined following Eq. 2.24. In this case, d represents the distance of the central point $[x_{central}\ y_{central}]^T$ to one of the closest chessboard corners $p1,\ p2,\ p3$ with their image coordinates $[x\ y]^T$. The coordinates $[x_{bl}\ y_{bl}]^T$ and $[x_{wh}\ y_{wh}]^T$ represent the centers of the black and the white marker.

$$d = \sqrt{(y - y_{central})^2 + (x - x_{central})^2} \quad \text{with}$$
$$y_{central} = \frac{1}{2}(y_{bl} + y_{wh}) \quad \text{and} \quad x_{central} = \frac{1}{2}(x_{bl} + x_{wh}) \tag{2.24}$$

The point with the shortest distance d to C may be the potential first chessboard point. To increase the robustness of the detection, angle θ is determined for these points as follows (see Eq. 2.25).



(a) (b) (c) (d)

Figure 2.18: Determination of central-point C located between the two markers and the first corner point (a), the second corner point (b) and the third corner point (c). The resulting corner order serves as input for camera calibration (d).

$$\theta = \arctan\left(\frac{|m_1 - m_2|}{1 + m_1 m_2}\right) \quad \text{with} \quad m_1 = \frac{y_{wh} - y_{bl}}{x_{wh} - x_{bl}}, m_2 = \frac{y - y_{central}}{x - x_{central}} \tag{2.25}$$

As shown in Figure 2.18(a), chessboard corner $p2$ has the shortest distance to C and is located on the straight line $P2$ that is approximately perpendicular to T. For this reason, this corner must be the first point on the calibration pattern. A second point can then be identified using the first point and the connecting line V connecting the first point with the central point C (see Figure 2.18(b)). Similarly to the previous stage, the three closest grid points $q1$ $q2$ $q3$ to the first points are extracted and the segments $Q1$ $Q2$ $Q3$ connecting these points with the first point are determined. The second point is the one whose segment is approximately perpendicular to V. Therefore, the cutting angle θ between this segment and V is computed following Eq. 2.25. The detection result illustrates Figure 2.18(c).

After detecting the second grid point, the third point is again the one with the shortest distance to the second grid point and the one whose segment with the second grid point leads to a small cutting angle θ. This process is repeated until all grid-points have been determined. Figure 2.18(d) illustrates the complete detection result and the resulting corner order.

2.3.2 Camera calibration

The next step in the calibration procedure is to estimate the parameters $a_0, a_2 \cdots, a_n$ and the affine parameters A, t to obtain the omnidirectional camera model. The determination of these parameters is based on the image positions of the extracted chessboard corners and is called camera calibration. This section briefly describes the calibration procedure that has been proposed by Scaramuzza *et al.* [26, 27] and that enables an accurate determination of the camera parameters. The calibration process is separated into two stages. At the first stage, the coefficients $a_0, a_2 \cdots, a_n$ – also called intrinsic parameters – are estimated assuming an ideal projection of the calibration corners onto the ideal sensor plane (see Section 2.2.4). This is realized by setting the affine transformation matrix A to the identity matrix and the translation to zero ($t = 0$). In a second stage, the affine transformation parameters A, t are determined

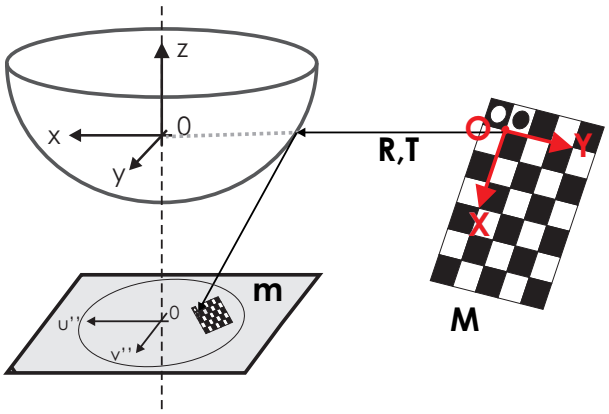


Figure 2.19: Coordinate system of a calibration pattern and its relation to the camera system.

and the results obtained are refined using nonlinear optimization. Regarding Figure 2.6(a) in Section 2.2.4, (u'', v'') are the pixel coordinates of a chessboard corner point $\mathbf{P}''$ and $\rho = \|u''\|$ is assumed to be the length of a vector to point P''.

In a first step, the parameters $a_0, a_2, ..., a_n$ are estimated. For this purpose, a planar chessboard pattern – with known geometry – is shown at several positions and orientations during the calibration process. The chessboard corners are extracted for each pose and their positions in image coordinates are determined. Each chessboard in a calibration image is related to the camera coordinate system by a rotation matrix $\mathbf{R}$ and by a translation $\mathbf{T}$ (see Figure 2.19). Variable I_i is denoted as an observed calibration image where $M_j^i = [X_j^i \; Y_j^i \; Z_j^i]^T$ are the 3D-coordinates of the chessboard corners within the chessboard pattern coordinate system $\mathbf{O}$. Furthermore, let $m = [u_j^i \; v_j^i]$ be the corresponding pixel coordinates on the ideal sensor plane. Superscript i denotes the observed calibration image and subscript j indicates the j-th chessboard corner on the i-th calibration image. Due to planar chessboard patterns in calibration images, the coordinates Z_j^i of chessboard corners can be set to zero. By using the camera model (see Section 2.2.4) and by stacking the image coordinates of the chessboard corners into Eq. 2.11.

$$\begin{aligned} \lambda_j^i \cdot \mathbf{p}_j^i &= \lambda_j^i \cdot \begin{bmatrix} u_j^i \\ v_j^i \\ a_0 + a_2\rho_j^i + \cdots + a_N {\rho_j^i}^N \end{bmatrix} = \mathbf{P}^i \cdot \mathbf{X}_j^i \\ &= \begin{bmatrix} \mathbf{r}_1^i \; \mathbf{r}_2^i \; \mathbf{r}_3^i \; \mathbf{T}^i \end{bmatrix} \cdot \begin{bmatrix} X_j^i \\ Y_j^i \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_1^i \; \mathbf{r}_2^i \; \mathbf{T}^i \end{bmatrix} \cdot \begin{bmatrix} X_j^i \\ Y_j^i \\ 1 \end{bmatrix} \end{aligned} \tag{2.26}$$

where $\mathbf{r}_1^i$, $\mathbf{r}_2^i$, $\mathbf{r}_3^i$ are the column vectors of the rotation matrix $\mathbf{R}^i$. Furthermore, Scaramuzza *et al.* [26, 27] make Eq. 2.26 independent from the scale-factor λ_j^i by multiplying both sides of

Eq. 2.26 vectorially with $\mathbf{p}_j^i$ as follows:

$$\begin{aligned} \lambda_j^i \cdot \mathbf{p}_j^i \times \mathbf{p}_j^i = \mathbf{p}_j^i \times \left[\mathbf{r}_1^i \; \mathbf{r}_2^i \; \mathbf{T}^i\right] \cdot \begin{bmatrix} X_j^i \\ Y_j^i \\ 1 \end{bmatrix} &= 0 \\ \begin{bmatrix} u_j^i \\ v_j^i \\ a_0 + a_2\rho_j^i + \cdots + a_N {\rho_j^i}^N \end{bmatrix} \times \left[\mathbf{r}_1^i \; \mathbf{r}_2^i \; \mathbf{T}^i\right] \cdot \begin{bmatrix} X_j^i \\ Y_j^i \\ 1 \end{bmatrix} &= 0 \end{aligned} \tag{2.27}$$

From that point on, Scaramuzza *et al.* [26, 27] generate three homogeneous equations Eq. 2.29, Eq. 2.30 and Eq. 2.30 with $g(\rho_j) = a_0 + a_2\rho_j^2 + \cdots + a_N\rho_j^N$. The world coordinates X_j, Y_j of the chessboard corners are obtained using the known length of each chessboard square in a calibration pattern and the corresponding image coordinates u_j, v_j are obtained from the corner extractor (see Section 2.3.1).

$$v_j(r_{31}X_j + r_{32}Y_j + t_3) - g(\rho_j)(r_{21}X_j + r_{22}Y_j + t_2) = 0 \tag{2.28}$$
$$g(\rho_j)(r_{11}X_j + r_{12}Y_j + t_1) - u_j(r_{31}X_j + r_{32}Y_j + t_3) = 0 \tag{2.29}$$
$$u_j(r_{21}X_j + r_{22}Y_j + t_2) - v_j(r_{11}X_j + r_{12}Y_j + t_1) = 0 \tag{2.30}$$

Thereafter, the unknown extrinsic parameters r_{11}, r_{12}, r_{21}, r_{22}, t_1, t_2 are stacked into a vector $\mathbf{H}$ (see Eq. 2.33). The linear solution of Eq. 2.32 and, hence, a solution for the vector $\mathbf{H}$ is computed by minimizing the least-squares criterion $min|\mathbf{M} \cdot \mathbf{H}|^2$ using the Singular Value Decomposition (SVD). Since vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ are orthonormal, the missing unknown entries r_{31} and r_{32} and the scale factor λ_j^i can be computed together in one step.

$$\mathbf{M} \cdot \mathbf{H} = 0 \tag{2.31}$$
$$\text{with} \quad \mathbf{H} = [r_{11} \; r_{12} \; r_{21} \; r_{22} \; t_1 \; t_2]^T \quad \text{and} \tag{2.32}$$
$$\mathbf{M} = \begin{bmatrix} -v_1X_1 & -v_1Y_1 & u_1X_1 & u_1Y_1 & -v_1 & u_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -v_LX_L & -v_LY_L & u_LX_L & u_LY_L & -v_L & u_L \end{bmatrix} \tag{2.33}$$

After that, Eq. 2.29 and Eq. 2.30 are reorganized into a matrix vector representation to obtain a set of linear equations. The intrinsic parameters $a_0, a_1, ..., a_n$ represent the geometry of the mirror shape whereas the missing extrinsic parameter t_3 is estimated by the linear least-squares solution of Eq. 2.34 for an arbitrary polynomial degree N.

$$\begin{bmatrix} A_j^1 & A_j^1{\rho_j^1}^2 & \cdots & A_j^1{\rho_j^1}^N & -v_j^1 & 0 & \cdots & 0 \\ C_j^1 & C_j^1{\rho_j^1}^2 & \cdots & C_j^1{\rho_j^1}^N & -u_j^1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_j^K & A_j^K{\rho_j^K}^2 & \cdots & A_j^K{\rho_j^K}^N & 0 & \cdots & 0 & -v_j^K \\ C_j^K & C_j^K{\rho_j^K}^2 & \cdots & C_j^K{\rho_j^K}^N & 0 & \cdots & 0 & -u_j^K \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_2 \\ \vdots \\ a_N \\ t_3^1 \\ t_3^2 \\ \vdots \\ t_3^K \end{bmatrix} = \begin{bmatrix} B_j^1 \\ D_j^1 \\ \vdots \\ B_j^K \\ D_j^K \end{bmatrix} \tag{2.34}$$

$$\text{where} \quad \begin{array}{c} A_j^i = r_{21}^i X_j^i + r_{22}^i Y_j^i + t_2^i \\ B_j^i = v_j^i (r_{31}^i X_j^i + r_{32}^i Y_j^i) \\ C_j^i = r_{11}^i X_j^i + r_{12}^i Y_j^i + t_1^i \\ D_j^i = u_j^i (r_{31}^i X_j^i + r_{32}^i Y_j^i) \end{array}$$

After the initial estimation of intrinsic and extrinsic parameters, Scaramuzza *et al.* proposed a two-stage linear refinement algorithm to optimize the initial calibration results. In a first stage, the extrinsic parameters $\mathbf{R}$ and $\mathbf{T}$ are recomputed by solving Eqs. 2.29, 2.30, 2.30 using the obtained intrinsic parameters $a_0, ..., a_n$. Secondly, the intrinsic parameters are refined by stacking the computed extrinsic parameters into Eqs. 2.29, 2.30. The intrinsic parameters are then recomputed by solving the linear equation system.

The center of distortion of the camera is estimated in another step. The distortion center is initially assumed to be coincident with the image center on the virtual sensor plane. Scaramuzza uses an iterative, exhaustive search process that performs many calibration trials for a fixed number of potential location center candidates that are uniformly spread over a certain image region in calibration images. For each trial, the reprojections – viz. the potential positions of chessboard corners in calibration images – are recomputed with the camera parameters determined during the calibration procedure. The reprojection result is compared with the true positions of the chessboard corners in the calibration images and a reprojection error for each trial is determined. This reprojection error has a global minimum when the estimated position best matches the real distortion center. The point with the smallest reprojection error is identified as the potential distortion center. New candidate locations in the region close to this point are specified and the process is repeated until the global minimum has been found.

Up to this point, the affine transformation matrix $\mathbf{A}$ was set to the identity matrix $\mathbf{I}$. This assumption can be made since the values of the affine matrix are very similar to the values of the identity matrix. So far, the calibration process proposed by Scaramuzza estimates the calibration parameters without modifying the affine matrix $\mathbf{A}$. The values of the matrix $\mathbf{A}$ can be determined in a second step by using a non-linear refinement stage that is based on a maximum likelihood estimation. Therefore, the following mathematical expression functional is minimized

$$E = \sum_{i=1}^{K} \sum_{j=1}^{L} |u_j^i - \tilde{u}(R^i, T^i, A, t, a_0, ...a_N, X_j^i|^2 \tag{2.35}$$

where u_j^i represents the coordinates of extracted chessboard corners and $\tilde{u}$ represents the reprojections of scene points X_j^i on the i-th chessboard pattern. A Levenberg-Marquardt [51, 52] algorithm is used to minimize this function, whereas the previously determined calibration parameters serve as an initial guess for minimization of Eq. 2.35. After minimization, the camera-parameters are estimated and can be used for image rectification and other applications.

2.4 Image rectification

Original images captured from omnidirectional cameras are distorted, not simply interpretable and are not easy to apply for normal image processing routines. For example, straight borders of rectangular objects or quadric surfaces might be represented as curves or as rhombuses in original images. For this reason, conventional image processing algorithms like Hough-transformation are no longer suitable. Methods and procedures exist that process original images of uncalibrated omnidirectional cameras, but no useful mathematical relation between real object properties like size and width can be found. Hence, there is a need to transform original images into panoramic images for conventional image processing routines. In this section, an image rectification algorithm is proposed to efficiently transform original images into panoramic images. First, different projections are presented that are capable of transforming original images into panoramic images. Based on the projection parameters and on the (calibrated) camera model (see Section 2.2.4), the algorithm computes the target projection area in 3D-world coordinates. Thereafter, corresponding 2D-image coordinates of the projection area on the sensor plane are determined and intensity values from the 2D-image coordinates are mapped onto the panoramic image using bicubic interpolation. Figure 2.20 provides a high level overview of the image rectification procedure.

2.4.1 Projections used for image rectification

In the image processing domain, two common techniques exist to transform images from one image space into another. These methods are called *Source-to-Target Mapping* and *Target-to-Source Mapping* and are suitable for geometric and perspective image transformations [53]. They also allow transforming original images, which are captured by omnidirectional cameras, into rectified, panoramic images for manifold applications such as surveillance, person extraction and ambiance monitoring.

Source-to-target mapping

Source-to-Target Mapping is a method to map pixel from a source image I onto a target image I'. This technique provides a function R to directly determine the pixel positions $(m',\ n')$ in a target image I' from the pixel positions $(u,\ v)$ in a source image I (see Eq. 2.36).

$$(m', n') = R(u, v) \tag{2.36}$$

Such computed pixel positions $(m',\ n')$ do not necessarily have to be coincident with the matrix points of target images. In other words, the positions may differ from the image matrix and may be located between four matrix points (see Figure 2.21(a)). Thus, the difficulty of Source-to-Target Mapping is to determine the pixel position in a target image I' where the algorithm maps the intensity $I(u,\ v)$. Furthermore, intensity information of many pixel positions $(m',\ n')$ in target images can not be directly determined from the pixel positions $(u,\ v)$ of an input image. Due to stretching effects caused by image rectification, the obtained image may contain

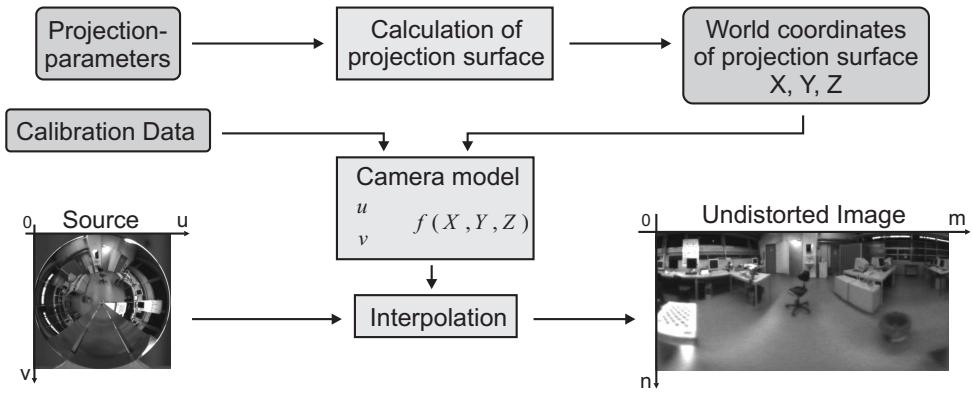


Figure 2.20: This diagram illustrates the proposed rectification process: The projection area is computed based on the calibration data and is stored in a look up table. The look up table is used to transform original images into rectified panoramic images [14].

vacant pixel positions where intensity information is not available (see Figure 2.22(a)). This is particularly the case for pixels in original images located close to the image center. Interpolation methods as proposed in [54, 55, 56] can be used to overcome this problem. By contrast, intensity values of pixels that are located at the boundary of original images may be mapped onto the same pixel position on target images.

Target-to-source mapping

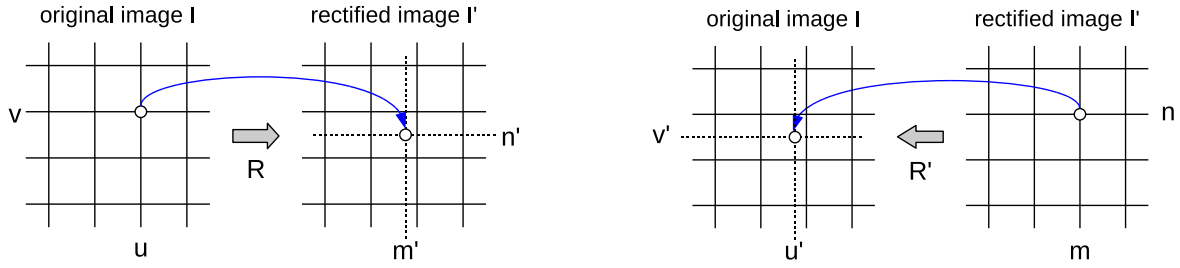
A different technique is the *Target-to-Source Mapping*. The main idea of Target-to-Source Mapping is to provide an inverse function R' to estimate the position of image pixels $(u',\ v')$ in source images I based on the matrix positions $(m,\ n)$ in target images I' (see Eq. 2.37).

$$(u', v') = R'(m, n) \tag{2.37}$$

In contrast to Source-to-Target Mapping, the main advantage of Target-to-Source Mapping is determining a corresponding pixel position $(u',\ v')$ for each target pixel position $(m,\ n)$. This overcomes the problem of vacant pixel positions in target images as each pixel in a rectified image is assigned to an intensity value from the original image (see Figure 2.22(b)). However, such computed pixel positions $(u',\ v')$ are also not coincident with the grid points of source images (see Figure 2.21(a)), but the required intensity values $I(u',\ v')$ for a pixel $(m,\ n)$ on a target image I can be easily estimated using interpolation methods.

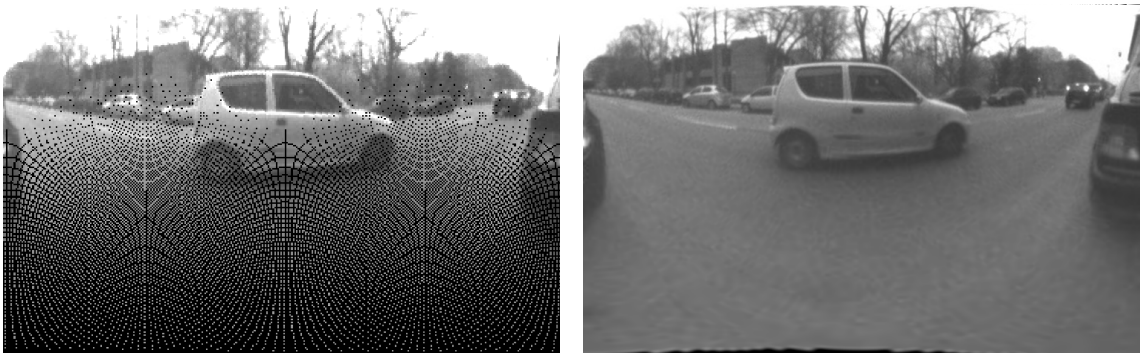
Cylindric projection

Cylindric projection is commonly used in robotics and is able to transform original images into panoramic images. Figure 2.23(a) illustrates a 3D-view of the cylindrical projection area that may relate to a cylindric cutout or to an entire cylinder. The size of the target projection area may be chosen by the following parameters. The parameters are the distance d of the projection



(a) Source-to-Target Mapping (b) Target-to-Source Mapping

Figure 2.21: Projection methods that are commonly used to transform images from one image space into another [53].



(a) (b)

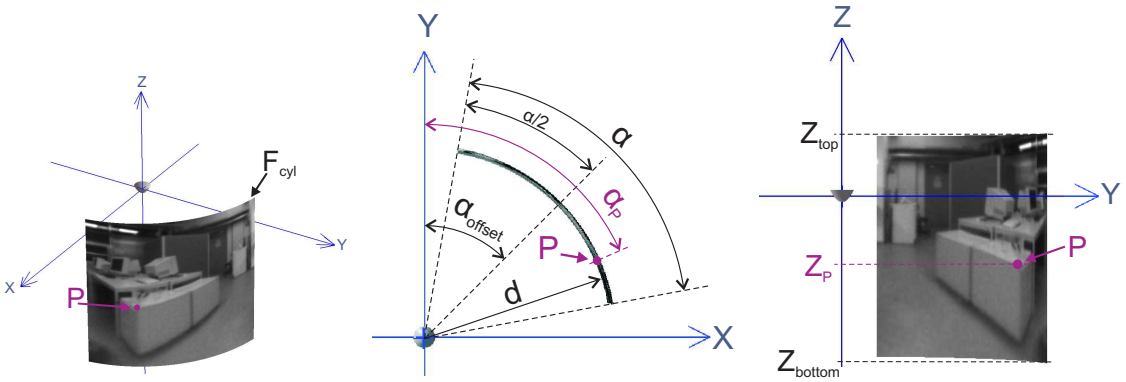
Figure 2.22: This figure illustrates rectified images using source-to-target (a) and target-to-source (b) mapping without interpolation to fill vacant pixel positions within target images.

area to the camera projection center, the rotation width α and the rotation offset α_{offset}, and the height of the projection area defined by the parameters Z_{top} and Z_{bottom} (see Figure 2.23(b) and Figure 2.23(c)). The cylindric coordinates $(\alpha_P, Z_P)^T$ of each image point P_F in the target area at position $[m, n]^T$ are calculated with the projection parameters and the target image size $(M, N)^T$ as follows:

$$\vec{P}_{F\,m,n} = \begin{bmatrix} d \\ \alpha_P \\ Z_P \end{bmatrix}_{m,n} = \begin{bmatrix} d \\ \left(\alpha_{offset} - \frac{\alpha}{2}\right) + \left(\frac{\alpha}{M} \cdot m\right) \\ Z_{top} - \left(\frac{Z_{top} - Z_{bottom}}{N} \cdot n\right) \end{bmatrix} \tag{2.38}$$

The cylindric coordinates are transformed into Cartesian coordinates assuming a clockwise rotation α_P beginning at the y - axis (see Eq. 2.39).

$$\vec{P}_F = \begin{bmatrix} X_P \\ Y_P \\ Z_P \end{bmatrix} = \begin{bmatrix} d \cdot \sin(\alpha_P) \\ d \cdot \cos(\alpha_P) \\ Z_P \end{bmatrix} \tag{2.39}$$



(a) 3D-View (b) Top-View (c) Side-View

Figure 2.23: Different views of the projection area for cylindric projection. (a): 3D-view of the projection area and its parameters (b,c) [14].

Conic projection

The *conic projection* is an alternative to the cylindric projection and is also able to map intensity values of pixels from original images onto a conic projection area (see Figure 2.24(a)). The projection area relates to an entire conic or a conic cutout. Its parameters are the distances from the projection center to the upper boundary d_{top} and to the lower boundary d_{bottom} of the target area, rotation α and the rotation offset α_{offset}, and the projection height that is defined by Z_{top} and Z_{bottom} (see Figure 2.24(b) and Figure 2.24(c)). Following Eq. 2.40, the cylindric coordinates $(\alpha_P, Z_P)^T$ of an image point P_F in the target area at position $[m, n]^T$ is computed using the projection parameters and the size of the target image $(M, N)^T$ as a first step.

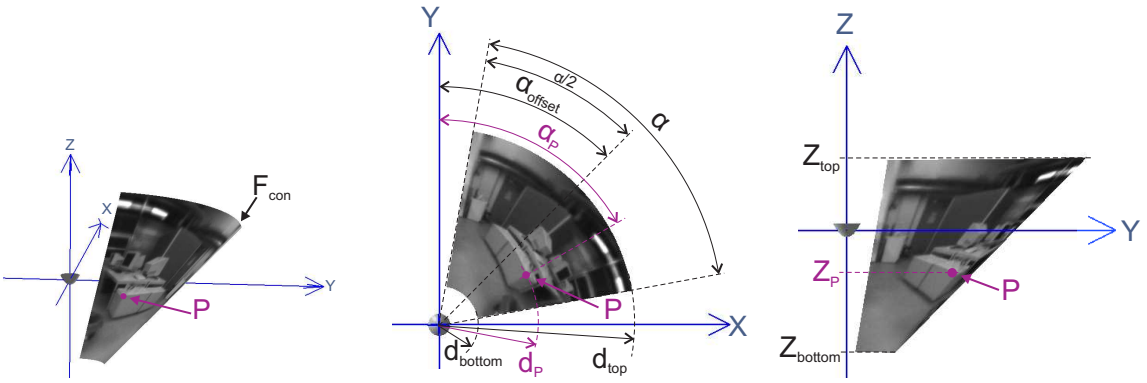
$$\vec{P}_{F\,m,n} = \begin{bmatrix} d \\ \alpha_P \\ Z_P \end{bmatrix}_{m,n} = \begin{bmatrix} d_{top} - \left(\frac{d_{top}-d_{bottom}}{N} \cdot n\right) \\ \left(\alpha_{offset} - \frac{\alpha}{2}\right) + \left(\frac{\alpha}{M} \cdot m\right) \\ Z_{top} - \left(\frac{Z_{top}-Z_{bottom}}{N} \cdot n\right) \end{bmatrix} \tag{2.40}$$

Similarly to cylindric projection, the cylindric coordinates is transformed into Cartesian coordinates assuming a clockwise rotation α_P beginning at the y - axis. Eq. 2.41 describes this transformation.

$$\vec{P}_F = \begin{bmatrix} X_P \\ Y_P \\ Z_P \end{bmatrix} = \begin{bmatrix} d \cdot \sin(\alpha_P) \\ d \cdot \cos(\alpha_P) \\ Z_P \end{bmatrix} \tag{2.41}$$

Spherical projection

A further projection is the *spherical projection* and is also commonly used in robotics. The spherical projection maps intensity values of pixels from original images onto a spheric projection area used for the target image. Figure 2.25(a) illustrates a 3D-view of the cylindrical



(a) 3D-View (b) Top-View (c) Side-View

Figure 2.24: Different views of the projection area for the conic projection. The 3D-view of the projection area (a) and its parameters (b,c) [14].

projection area. The size of the projection area can be defined using the following transformation parameters. These parameters are the elevation width β, elevation offset β_{offset} as well as the rotation width α and the rotation offset α_{offset} (see Figure 2.25(b) and Figure 2.25(c)). The center of the projection area is assumed to be coincident with the projection center of the mirror. Consequently, each image point P on the projection plane has the same solid angle regardless of its distance to the projection center. Each image point in the projection area is also projected onto the same image point on the camera sensor for each distance d. Hence, the distance r between the projection area and the projection center is set to unity for all image points P on the target area.

The spherical coordinates $(\alpha_P, Z_P)^T$ of an image point P in the projection area at position $[m, n]^T$ within the target image are computed based on the projection parameters and on the target image size $(M, N)^T$ (see Eq. 2.42).

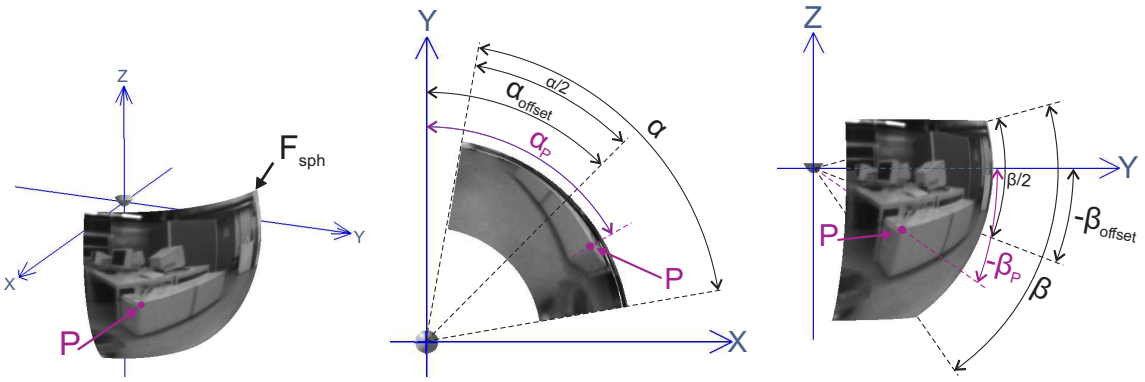
$$\vec{P}_{F\,m,n} = \begin{bmatrix} r \\ \alpha_P \\ \beta_P \end{bmatrix}_{m,n} = \begin{bmatrix} r \\ \left(\alpha_{offset} - \frac{\alpha}{2}\right) + \left(\frac{\alpha}{M} \cdot m\right) \\ \left(\beta_{offset} + \frac{\beta}{2}\right) - \left(\frac{\beta}{N} \cdot n\right) \end{bmatrix} \quad \text{with} \quad r = 1 \tag{2.42}$$

Finally, the spherical coordinates are transformed into Cartesian coordinates assuming a clockwise rotation α_P beginning at the y - axis and an anti-clockwise elevation β_P beginning at the x - axis (see Eq. 2.43).

$$\vec{P}_F = \begin{bmatrix} X_P \\ Y_P \\ Z_P \end{bmatrix} = \begin{bmatrix} 1 \cdot \sin(\alpha_P) \cdot \cos(\beta_P) \\ 1 \cdot \cos(\alpha_P) \cdot \cos(\beta_P) \\ 1 \cdot \sin(\beta_P) \end{bmatrix} \tag{2.43}$$

Plane projection

The presented cylindric, conic and spherical projections can generate panoramic images that have a horizontal view up to $360°$. Contrary to the these projections, the *plane projection* gener-



(a) 3D-View (b) Top-View (c) Side-View

Figure 2.25: Different views of the spherical projection area. The 3D-view of the projection area (a) and its parameters (b,c) [14].

ates images as if taken directly by a perspective camera. In other words, the plane projection is suitable for treating an omnidirectional camera as a perspective camera and may be interesting for applications that use only a limited field of view of the camera. Plane projection maps pixels from original images onto a rectangular image plane. Figure 2.26(a) illustrates a 3D-view of the plane target projection area. Parameters for this projection are the elevation offset β_{offset}, the rotation offset α_{offset}, the distances d between the projection area and projection center and the size of the target plane b and h (see Figure 2.26(b) and Figure 2.26(c)).

Following Eq. 2.44, the distance of an image point P_F in the target image at position $[m, n]^T$ to the left border b_P and the distance of this point to the lowest border h_p of the projection area are calculated using the size of the target image $(M, N)^T$, the target image width b and the target image height h.

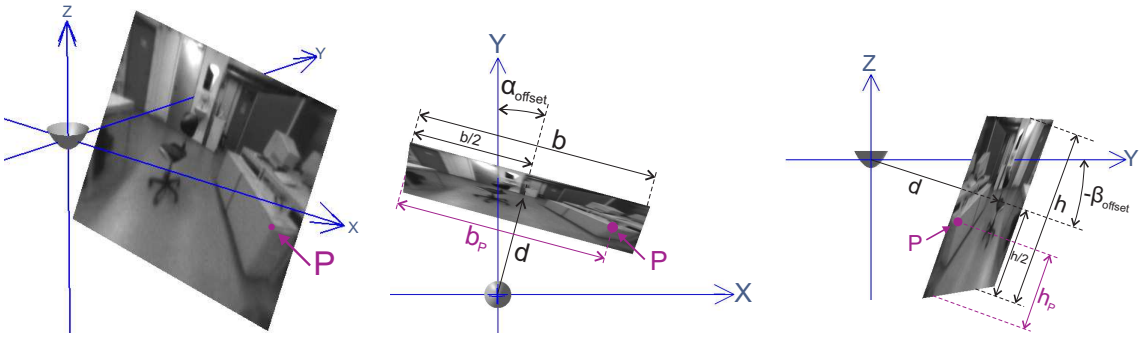
$$\begin{bmatrix} b_P \\ h_P \end{bmatrix}_{(m,n)} = \begin{bmatrix} \frac{b}{M} \cdot m \\ \frac{h}{N} \cdot n \end{bmatrix} \tag{2.44}$$

The normalized vector $\vec{F}'_C$ with its direction towards the image center of the target plane is computed with the distance between the target area and the projection center, the elevation β_{offset} and the rotation α_{offset} (see Eq. 2.45).

$$\vec{F}_{C'} = \begin{bmatrix} \frac{X_{F_C}}{d} \\ \frac{Y_{F_C}}{d} \\ \frac{Z_{F_C}}{d} \end{bmatrix} \quad \text{with} \quad \vec{F}_C = \begin{bmatrix} X_{F_C} \\ Y_{F_C} \\ Z_{F_C} \end{bmatrix} = \begin{bmatrix} d \cdot \sin(\alpha_{offset}) \cdot \cos(\beta_{offset}) \\ d \cdot \cos(\alpha_{offset}) \cdot \cos(\beta_{offset}) \\ d \cdot \sin(\beta_{offset}) \end{bmatrix} \tag{2.45}$$

After that, another normalized vector $\vec{F}_m$ is generated that is perpendicular to vector $\vec{F}'_C$ and whose direction is coincident with the direction of the target image coordinate m (see Eq. 2.46).

$$\vec{F}_n = \vec{F}_{C'} \times -\vec{F}_m \quad \text{with} \quad \vec{F}_m = \begin{bmatrix} Y_{F_{C'}} \\ -X_{F_{C'}} \\ 0 \end{bmatrix} \tag{2.46}$$



(a) 3D-View (b) Top-View (c) Side-View

Figure 2.26: Different views of the plane projection area. The 3D-view of the projection area (a) and its parameters (b,c) [14].

The normalized vector $\vec{F}_n$ may now be determined using the cross product $\vec{F}_n = \vec{F}'_C \times -\vec{F}_m$. Similarly to $\vec{F}_m$, $\vec{F}_n$ is perpendicular to $\vec{F}'_C$ and its direction is coincident with the direction of the target image coordinate n. Finally, the world coordinates of each image point P_F on the target area is computed using the vectors $\vec{F}_m$, $\vec{F}_n$, $\vec{F}_C$ and the parameters h_p and b_p (see Eq. 2.47).

$$\vec{P}_F = \vec{F}_C + \vec{F}_m \cdot (b_p - \frac{b}{2}) + \vec{F}_n \cdot (h_p - \frac{h}{2}) \tag{2.47}$$

2.4.2 Transformation from world to sensor coordinates

The determination of the projection area and the computation of its image coordinates in 3D-world coordinates (see Section 2.4.1-2.4.1) is the first step for image rectification. The world coordinates of each pixel position in the projection area are available and can be transformed to find their corresponding 2D-sensor coordinates. This transformation is required to map the intensity values of sensor pixels onto the corresponding pixel positions in the target projection area.

The transformation from world to sensor coordinates is realized with the camera model presented in Section 2.2.4. Therefore, the inverse camera model has to be determined and requires the extrinsic and intrinsic calibration parameters.

$$\vec{p} = \begin{bmatrix} x_{\vec{p}} \\ y_{\vec{p}} \\ z_{\vec{p}} \end{bmatrix} = \begin{bmatrix} x_{\vec{p}} \\ y_{\vec{p}} \\ a_0 + a_2\rho^2 + \ldots + a_N\rho^N \end{bmatrix} = \lambda \cdot \begin{bmatrix} u_{P''} \\ v_{P''} \\ a_0 + a_2\rho^2 + \ldots + a_N\rho^N \end{bmatrix} \tag{2.48}$$

Eq. 2.48 illustrates the inverse camera model (calibration function) for an image point P'' on the target projection area in world-coordinates. Thus, ρ is defined as the length of the vector $\vec{v} = [u_{P''}\ v_{P''}]^T$ to an image point on the sensor plane with $\rho = ||\vec{v}|| = \sqrt{u_{P''}^2 + v_{P''}^2}$.

In contrast to the calibration function, the scaling-factor λ, which has been presented in Section 2.3.2, is important to determine the inverse camera model. The world coordinates of each

scene point have to be computed using the scaling factor λ. In other words, λ is the scaling factor that describes all 3D-points located on a light ray that are projected into the same image point on the sensor plane. To facilitate the computation, a new scaling factor λ' is introduced that combines both λ and ρ to a new scaling factor so that the total length of vector $\vec{p}$ depends only on λ' (see Eq. 2.49).

$$\vec{p} = \begin{bmatrix} x_{\vec{p}} \\ y_{\vec{p}} \\ z_{\vec{p}} \end{bmatrix} = \lambda' \cdot \begin{bmatrix} \frac{u_{P''}}{\rho} \\ \frac{v_{P''}}{\rho} \\ \frac{(a_0+a_2\rho^2+\ldots+a_N\rho^N)}{\rho} \end{bmatrix} \quad \text{with} \quad \left| \begin{bmatrix} \frac{u_{P''}}{\rho} \\ \frac{v_{P''}}{\rho} \end{bmatrix} \right| = 1 \quad \text{and} \quad \lambda' = \lambda \cdot \rho \tag{2.49}$$

Following Eq. 2.49, λ' represents the length of vector $\vec{p}$ and may be calculated using the X, Y components for any pixel position on the projection plane (see Eq. 2.50).

$$\lambda' = \sqrt{x_{\vec{p}}^2 + y_{\vec{p}}^2} \tag{2.50}$$

The Z component of vector $z_{\vec{p}}$ can now be determined using Eq. 2.49 and Eq. 2.50:

$$\frac{z_{\vec{p}}}{\lambda'}\rho = a_0 + a_2\rho^2 + \ldots + a_N\rho^N \tag{2.51}$$

After reorganizing Eq. 2.51, ρ can be computed by solving the n-polynomial function that describes the characteristic of the mirror shape (see Eq. 2.52).

$$0 = a_0 - \frac{z_{\vec{p}}}{\lambda'}\rho + a_2\rho^2 + \ldots + a_N\rho^N \tag{2.52}$$

This equation is numerically solved and may provide several solutions. Complex and negative numbers as well as double roots are neglected since ρ represents a real positive length of a vector $\vec{v}$ on the sensor plane. For this reason, the solution for ρ must be the smallest, real and positive root and serves as an input to Eq. 2.53 along with the scaling-factor λ'.

$$\begin{bmatrix} u_{P''} \\ v_{P''} \end{bmatrix} = \frac{\rho}{\lambda'} \cdot \begin{bmatrix} x_{\vec{p}} \\ y_{\vec{p}} \end{bmatrix} \quad \text{with} \quad \lambda' = \sqrt{x_{\vec{p}}^2 + y_{\vec{p}}^2} \tag{2.53}$$

Eq. 2.53 represents the inverse camera model of the extrinsic calibration and is used to transform image points in the projection area with world coordinates into sensor coordinates on the virtual sensor plane E'' (see Section 2.2.4). The extrinsic calibration can be easily inverted following Eq. 2.54:

$$\vec{P} = \begin{bmatrix} u_P \\ v_P \end{bmatrix} = \mathbf{A}^{-1} \cdot (P'' - \vec{t}) \quad \text{with} \quad \vec{t} = \begin{bmatrix} u_{center} \\ v_{center} \end{bmatrix} \quad \text{and} \quad \mathbf{A} = \begin{bmatrix} c & d \\ d & 1 \end{bmatrix} \tag{2.54}$$

Using the inverse extrinsic calibration (see Eq. 2.53) and the inverse intrinsic calibration (see Eq. 2.54), the corresponding sensor coordinates $(u,\ v)$ for each image pixel $(m,\ n)$ in a panoramic image can be computed. This inverse camera model relates to the inverse projection function R' for the *Target-to-Source Mapping* presented in the last section.

2.4.3 Interpolation

Once the corresponding sensor coordinates of image pixels in a panoramic image have been computed, these coordinates are available for further rectification process and do not need to be recomputed for the same omnidirectional camera. Theses sensor coordinates $(u,\ v)$ are stored in a matrix $\mathbf{F}$ that functions as a *Look-Up-Table* (LUT) to quickly transform original images into panoramic images in online rectification processes. Eq. 2.55 presents the structure of this matrix: $\mathbf{F}' = M \times N \times 2$ with M = Number of Columns, N = Number of Rows (2.55) Following Eq. 2.55, the size of $\mathbf{F}$ relates to the size of the panoramic image $M,\ N$ whereas the values at the entries $(m,\ n)$ of matrix $\mathbf{F}$ contain the required sensor coordinates instead of intensity values.

$$P_{\mathbf{F}} = \begin{bmatrix} u \\ v \end{bmatrix}_{m,n} = \begin{bmatrix} \mathbf{F}_{m,n,1} \\ \mathbf{F}_{m,n,2} \end{bmatrix} \quad \text{with} \quad m = 1 \ldots M \quad \text{and} \quad n = 1 \ldots M \tag{2.56}$$

The computed sensor coordinates $(u,\ v)$, however, are not coincident with the grid points of the camera sensor. In other words, the pixel positions $(u,\ v)$ differ from the grid points of the camera sensor and could be located between several matrix elements (see Figure 2.27). Thus, the difficulty is to determine the sensor pixel (or pixels) whose intensity value $I(u, v)$ can be mapped to the target area. Interpolation can be used to overcome this problem and to determine the correct intensity value of a pixel in a target image.

In general, interpolation is a technique to determine values of certain positions in a discrete function when the required positions are not coincident with the sampling points. Common examples in the image processing domain using interpolation include geometric image transformations such as image scaling or image rotation. The main focus of interpolation here is to estimate intensity values for pixels in a target image whose reprojected locations are not coincident with grid points of the camera sensor. The most common interpolation methods and the interpolation kernels are presented below.

Nearest-Neighbor interpolation

Nearest-Neighbor Interpolation is an interpolation method in one or more dimensions and is also known as proximal interpolation or point sampling interpolation [53]. Nearest-neighbor interpolation approximates values for points on plane using the intensity values of points in the neighborhood of theses points. This interpolation simply selects the intensity value of the nearest sensor pixel $(u, v)_{Interpolation}$ in a neighborhood around a computed sensor pixel (u', v'). To achieve this, it is sufficient to round the computed sensor coordinates (u', v') to integer values. The interpolation kernel following Eq. 2.57 is presented in Figure 2.28(a).

$$K_{nn}(x, y) = \begin{cases} 1 & , \quad 0 \leq |x|, |y| < 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{2.57}$$

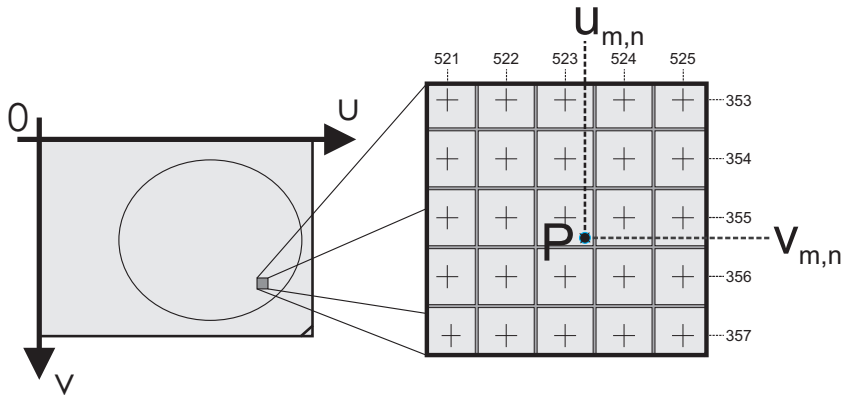


Figure 2.27: The position of matrix points in panoramic images differs from the position of grid points on the camera sensor. Intensity values of camera pixels must be interpolated to obtain valid intensity values for pixels in a panoramic image [14].

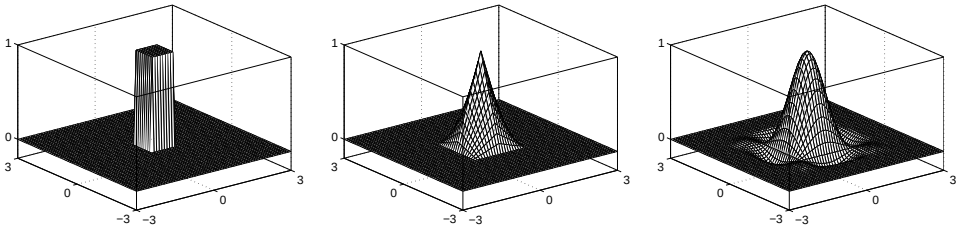
An intensity value determined in the manner for a sensor position (u', v') represents the intensity value at the corresponding target image pixel (m, n). Nearest-neighbor interpolation is fast and easy to implement, but it does not consider intensity values of other, neighboring pixels at all. As a result, there are staircase-shaped image artifacts in rectified images. These artifacts may lead to several problems in object detection as properties of objects may be changed. Bilinear and bicubic interpolation overcome this limitation and provide better rectification results.

Bilinear interpolation

In contrast to nearest-neighbor interpolation, *bilinear interpolation* considers intensity values of neighboring pixels. Bilinear interpolation is an extension of the one dimensional linear interpolation [53] and can be used to interpolate functions with two dimensions, e.g. two variables (sensor coordinates) on a regular image grid in one domain that differs from the sensor grid of a camera. Bilinear interpolation first performs a linear interpolation in one direction and another interpolation in the other direction.

In image processing, bilinear interpolation is realized with a 2D-kernel function and allows an interpolation of intensity values for every computed sensor coordinate (u', v'). Bilinear interpolation uses intensity values of the four nearest pixels to find an appropriate intensity value for a specific pixel position. Therefore, the distances to the four nearest sensor pixels (u_i, v_i) are calculated and the intensity values of the four nearest pixels are weighted depending on their distance to the target sensor coordinate (u', v') (see Eq. 2.58). That will lead to a smoother, panoramic image that contains fewer artifacts compared to rectified images using nearest-neighbor interpolation.

$$K_{bl}(x, y) = \begin{cases} 1 - x - y - xy & , \quad 0 \leq |x|, |y| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.58}$$



(a) Nearest Neighbor (b) Bilinear Interpolation (c) Bicubic Interpolation with $a = -1$

Figure 2.28: Different kernels that are used for interpolation.

Bicubic interpolation

The best kernel function to be used for interpolation is the $sinc$-function where $sinc(x) = \frac{sin(\pi x)}{\pi x}$. However, Burger *et al.* [53] mentioned that the $sinc$-function is not suitable for practical applications due to its infinite kernel size. Instead, approximating the $sinc$-function with a locally defined $cubic$-function $k_{bc}(u)$ helps to make this kernel applicable for technical purposes. Eq. 2.59 describes the approximated $sinc$-function where parameter a defines its gradient as presented in [53].

$$k_{bc}(u) = \begin{cases} (a+2)\cdot|u|^3 - (a+3)\cdot|u|^2 + 1 & , \quad 0 \leq |u| < 1 \\ a\cdot|u|^3 - 5a\cdot|u|^2 + 8a\cdot|u| - 4a & , \quad 1 \leq |u| < 2 \\ 0 & , \quad 0|u| \geq 2 \end{cases} \tag{2.59}$$

Similarly to bilinear interpolation, *bicubic interpolation* is an extension of the one-dimensional cubic-function $k_{bc}(u)$ and is used to interpolate functions with two dimensions. The kernel-function is defined by $K_{bc}(u,v) = k_{bc}(u)\cdot k_{bc}(v)$ for the sensor coordinates (u,v). Figure 2.28(c) illustrates the kernel for parameter $a = -1$. Panoramic images are smoother and have fewer interpolation artifacts than other images obtained with other interpolations: Therefore, bicubic interpolation is often chosen over nearest-neighbor or bilinear interpolation. However, the implementation of bicubic interpolation is more complex and leads to longer computation times.

2.5 Pixel density

In previous sections, various projections have been presented that can be used to transform original images from omnidirectional cameras into panoramic images, but which projection is best for a specific application? In this section, a novel method is presented to evaluate the different projections in terms of best utilization of sensor pixels in panoramic images for specific applications. Therefore, Florian Böhm [14, 3] proposed the *pixel density* as a novel tool to determine the resolution and, hence, the quality of panoramic images depending on the chosen projection. The pixel density is also suitable for comparing various projections for image transformation and camera/mirror configurations in terms of best utilization of sensor pixels in panoramic images.

Baker [31] introduced a formalism to determine the resolution of omnidirectional cameras that depends on the mirror characteristic k and on the distance c between the projection center and the pinhole of the perspective camera. However, this camera resolution does not indicate anything about the pixel distribution in rectified images when choosing a specific projection such as the spherical or cylindrical projection. For this reason, the pixel density σ is proposed as a new value to evaluate different projections and to compare different configurations of omnidirectional cameras in terms of best utilization of sensor pixels in panoramic images.

The pixel density is also called the *position-dependent resolution* of panoramic images and is determined using the distances of neighboring pixels in rectified images reprojected onto the sensor plane. In other words, the positions of neighboring pixels in panoramic images are mapped onto their corresponding positions on the sensor plane, and their distances to each other on the sensor plane are computed. Thus, the horizontal pixel density $\sigma_{h(m,n)}$ and the vertical pixel density $\sigma_{v(m,n)}$ are required to compute the pixel density σ. Figure 2.29 gives an overview of the basic concept for determining the pixel density. The horizontal pixel density $\sigma_{h(m,n)}$ at position $[m,n]^T$ is computed following Eq. 2.60:

$$\sigma_{h\,(m,n)} = \frac{1}{2} \cdot d_{h\,(m,n)} \quad \text{with} \tag{2.60}$$

$$d_{h\,(m,n)} = \sqrt{(u_{(m+1,n)} - u_{(m-1,n)})^2 + (v_{(m+1,n)} - v_{(m-1,n)})^2}$$

Analogically, the vertical pixel density $\sigma_{v(m,n)}$ at position $(m,n)^T$ is defined as

$$\sigma_{v\,(m,n)} = \frac{1}{2} \cdot d_{v\,(m,n)} \quad \text{with} \tag{2.61}$$

$$d_{v\,(m,n)} = \sqrt{(u_{(m,n+1)} - u_{(m,n-1)})^2 + (v_{(m,n+1)} - v_{(m,n-1)})^2}$$

The pixel density $\sigma_{g(m,n)}$ at position (m,n) is defined as the geometric mean of the horizontal and vertical pixel densities following Eq. 2.62:

$$\sigma_{g\,(m,n)} = \sqrt{\sigma_{h\,(m,n)} \cdot \sigma_{v\,(m,n)}} \tag{2.62}$$

The pixel density σ describes the utilization of sensor pixels in panoramic images whose intensity values are used to compute the intensities of image pixels in panoramic images. Consequently, the pixel density is also a measurement value to indicate the resolution of image regions in rectified, panoramic images. It is shown, that best utilization of sensor pixels and therefore best image quality of panoramic images is obtained for images having a pixel density close to unity. This can be understood as intensity information of one sensor pixel that is directly mapped to the corresponding pixel position on a target image. Moreover, a pixel density less than unity denotes poor resolution as the information of one sensor pixel is used for several pixels in panoramic images. Values of the pixel density larger than unity denote good resolution but also indicate a wast of sensor pixels since intensity information of several sensor pixels is mapped onto the same pixel in a panoramic image. As discussed later in Section 2.6, the characteristic of the pixel density depends on the chosen camera configuration and on the chosen projection.

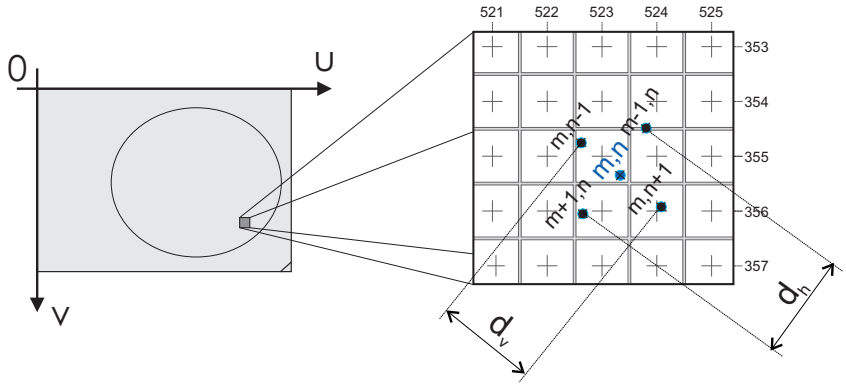


Figure 2.29: Positions of panoramic image pixels $(m,\ n)$ and their distances to each other on the sensor plane. These distances are used to compute the pixel density [14].

2.6 Results

In this section, the performance of the proposed chessboard corner extraction algorithm (see Section 2.3.1) is evaluated in terms of robustness against differently illuminated calibration images and in terms of its relevance for the quality of calibration results. Additionally, this evaluation has been carried out in terms of accuracy, reprojection error and number of iterations required for error minimization by means of several experimental results on real and simulated image data.

For this purpose, variously illuminated calibration images with different numbers of chessboard squares were captured both by real, low resolution and high resolution omnidirectional cameras. These images served as input to evaluate the performance of the proposed algorithm and are presented in the following. Further detailed experiments and evaluations of the calibration procedure (see Section 2.3.2) were presented by Scaramuzza and may be found in [30]. In [30], the calibration algorithm was evaluated in terms of the influence of noisy input data due to manual selection of chessboard corners, the degree of the polynomial mirror function f (see Section 2.2.3 and Section 2.2.4) and the influence of the number of calibration images on the results. Camera calibration and chessboard corner detection all run in a Matlab environment so that the times for execution measured in the experiments relate to the execution times needed using the Matlab environment.

2.6.1 Chessboard detection and camera calibration

Chessboard corner extraction

In a first setup, experiments were conducted to evaluate the robustness of the proposed chessboard-corner extraction algorithm. Therefore, variously illuminated calibration images containing different numbers of chessboard squares ($05x07$, $07x09$, $09x11$) were captured by both a low resolution (640×480 pixel) and a high resolution (1024×768 pixel) omnidirectional

(a) pI (b) dhD (c) dsN (d) bhD (e) bsN

Figure 2.30: Differently illuminated calibration images used for performance evaluation of the proposed chessboard corner extraction algorithm. Legend: *pI*: perfectly illuminated, *dhD/bhD*: dark/bright with high dynamic, *dsN/bsN*: dark/bright with strong noise

camera. The lighting scenarios were categorized in five illumination types. These types are perfectly illuminated calibration images ***pI*** and dark and bright images both with strong noise and high dynamic (***dhD/bhD***: dark/bright with high dynamic, ***dsN/bsN***: dark/bright with strong noise). Figure 2.30 illustrates sample images for each scenario captured by a real omnidirectional camera in the laboratory. In the following, the shortcuts introduced for each scenario are utilized for figures and tables.

Table 2.1 illustrates chessboard corner extraction results using sets of 40 calibration images obtained for each scenario. The evaluation is performed for omnidirectional cameras with a hyperbolic mirror with a mirror constant $k = 7.55$. Simulations demonstrated that the mirror type (parabolic, conic, hyperbolic mirror) has no influence on chessboard corner extraction. The upper part of Table 2.1 shows the percentage detection rates of chessboard corners in low resolution images (640×480 pixels). A detection rate of 92.5% means that the chessboard corners are sucessfully extracted in 92.5% of all test images. The lower part of Table 2.1 presents the percentage detection rates of chessboard corners in high resolution images (1024×768 pixels). Chessboard corner extraction is successfully completed when the chessboard, the markers and the corners are extracted and when the corner order is correct.

It can been seen, that perfectly illuminated calibration images lead to good detection results for all tested scenarios. Difficult lighting scenarios also lead to satisfying results in chessboard corner extraction, whereas the detection rates significantly decrease for very dark and noisy calibration images. Moreover, the detection rate for $09x11$ squared chessboards in low resolution calibration images is very small due to bad extraction of very small squares in low resolution images. In particular, very small chessboard squares close to the image center are difficult to extract and lead to a failure of chessboard detection. On the other hand, less squared chessboards in high resolution images (f.ex. $05x07$) may also result in worse detection results. In the algorithm, the chessboard region is assumed to be the one that contains the most rectangles: Consequently, images regions that contain more rectangles than the chessboard pattern are wrongly classified valid chessboard regions. Increasing the number of chessboard squares in high resolution calibration images easily overcomes this limitation and also leads to a better approximation of the mirror function.

Resolution	Squares	dhD	dsN	**pI**	bhD	bsN
640×480	05x07	92.5%	90.0%	**97.5**%	92.5%	85.0%
	07x09	87.5%	82.5%	**95.0**%	90.0%	87.5%
	09x11	72.5%	65.0%	**80.0**%	75.5%	67.5%
1024×768	05x07	87.5%	75.0%	**90.0**%	85.0%	77.5%
	07x09	92.5%	85.0%	**97.5**%	90.0%	87.5%
	09x11	90.0%	82.5%	**95.0**%	92.0%	85.0%

Table 2.1: Detection rates [%] for various illumination scenarios using low-resolution (640×480 pixels) (top) and high-resolution (1024×768 pixels) images (bottom).

Chessboard Extraction	Marker	Corner	Corner Order
$\approx 59\%$	$\approx 33\%$	$\approx 7\%$	$\approx 1\%$

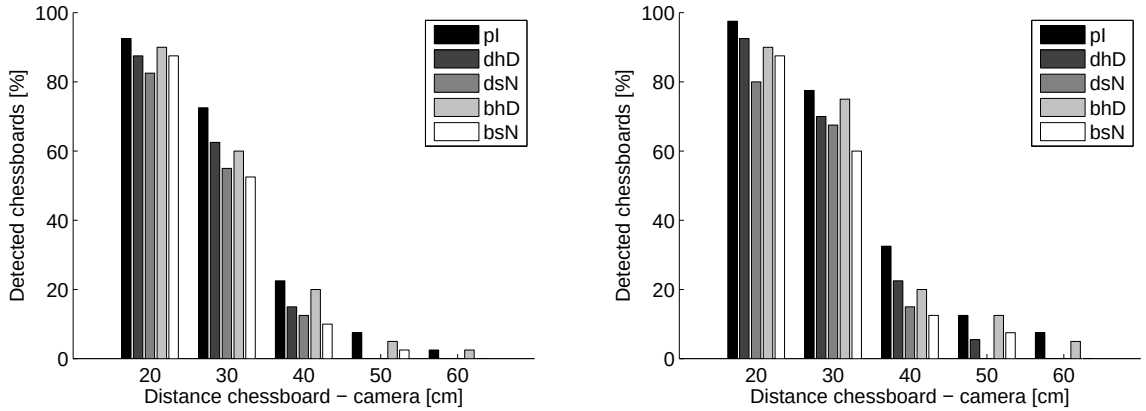
Table 2.2: Failure rate for intermediate extraction steps.

When the overall chessboard corner extraction fails, the algorithm fails mostly in one of its intermediate extraction steps. Table 2.2 gives an overview of failure rates for the proposed extraction steps. For example, a failure rate of 59% for chessboard extraction indicates that, when chessboard corner extraction breaks off, chessboard extraction fails in 59% of all cases. The main difficulty in chessboard corner extraction is the robust detection of the chessboard regions. This can be seen in the high failure rate for chessboard detection. Once the chessboard region is identified, marker as well as corner detection can be easily performed.

As a next step, the detection rate for chessboard corner detection for different distances between camera and chessboard is determined. Similarly to previous experiments, chessboard corner extraction is complete when chessboard, markers and corners are successfully extracted and when the corner order is correct. Figure 2.31 gives an overview of the detection rates both for low-resolution images (see Figure 2.31(a)) and for high-resolution images (see Figure 2.31(b)) captured in variously illuminated scenarios. Good detection rates are obtained for chessboards located close to the camera ($< 20cm$), whereas chessboards positioned far away from the camera cannot be sufficiently detect. In 87% of all cases when chessboard-corner extraction failed, the chessboard region cannot be sufficiently detected.

Iterative calibration refinement

The calibration algorithm proposed by Scaramuzza (see Section 2.3.2) allows estimating the calibration error. The calibration error is suitable for determining the quality of camera calibration using the reprojection error. With the help of the determined calibration parameters, the known 3D-world coordinates of the chessboard corners are reprojected into image coordinates. This reprojection is compared to the location of the extracted corners in calibration images: The average of all differences between the image coordinates recomputed for all chessboard corners and the actual positions of the extracted corners in calibration images is called the reprojection error. Consequently, a low reprojection error means a good calibration result and vice versa. The calibration process is refined by an iterative minimizing of this reprojection error. The



(a) Low-resolution images

(b) High-resolution images

Figure 2.31: Detection rates [%] for differently illuminated calibration patterns captured at different distances to the camera.

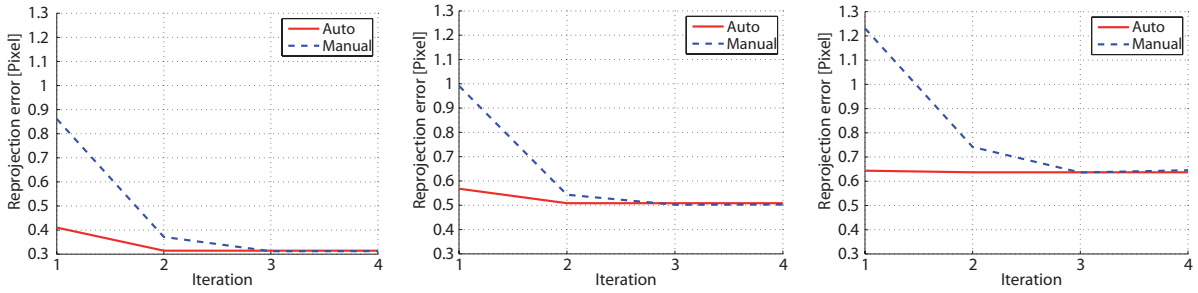
iteration process stops when there are no significant changes of the actual reprojection error compared to the reprojection errors of previous iteration stages.

In further experiments, the number of iterations that is required to refine the calibration for both manually and automatically extracted chessboards is measured and compared. Figure 2.32 presents the number of iterations that are required to obtain sufficient calibration results for an omnidirectional camera using different calibration patterns. The camera used in the experiment has a hyperbolic mirror with fixed mirror constant $k = 7.55$ (see Figure 2.32(a), Figure 2.32(b), Figure 2.32(c)). Figure 2.32 also shows that manual selection of chessboard corners leads to larger initial reprojection errors compared to camera calibration with automatic chessboard corner extraction. The higher reprojection error results from inaccuracies caused by manual selection of chessboard corners in calibration images. Consequently, more iteration stages are necessary to refine the calibration when using manually extracted corners. It can be seen, that at least four iteration stages are necessary to refine camera calibration based on manually selected chessboard corners and to obtain similar accuracies compared to refinement stages based on automatic chessboard corner extraction.

However, it is noteworthy that the reprojection error increases with an increasing number of chessboard corners. The calibration is optimized to approximate a mirror function that matches to all chessboard corners. In general, functions that are approximated with many supporting points have a higher overall error than functions that are approximated with only a few (see Figure 2.32(c)). However, the quality of the mirror function increases with an increasing number of chessboard corners. This is presented in the next section.

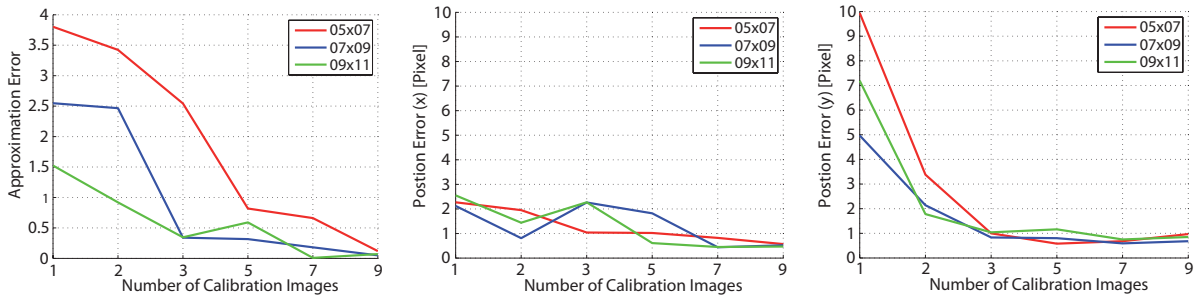
Number of calibration images

The number of calibration images significantly influences the quality of the approximated mirror function f and also influences the accuracy of the position of the determined distortion



(a) $k = 7.55$, Sqs: 05x07 (b) $k = 7.55$, Sqs: 07x09 (c) $k = 7.55$, Sqs: 09x11

Figure 2.32: Number of iteration stages required for minimizing the re-projection error.



(a) (b) (c)

Figure 2.33: Approximation error of the estimated mirror function (a) and the positioning error of the distortion center (b, x-position), (c, y-position) over the number of calibration images.

center. In general, the reprojection error depends on the approximation error of the mirror function f and on the positioning error of the distortion center $(u,\ v)$. Large approximation errors and a large position error of the distortion center result in a high reprojection error. In this section, experiments are conducted to study the approximation error of the mirror function f (polynomial degree = 4) and the positioning error of the distortion center depending on the number of calibration images. Therefore, calibration images are captured by an omnidirectional camera that has a hyperbolic mirror and a mirror constant ($k = 7.55$).

Figure 2.33(a) illustrates the approximation error of the estimated mirror function f for different numbers of calibration images. The approximation error has been determined by computing the size of the area that is spanned by the known ground truth mirror function and by the estimated mirror function. This area becomes zero when both functions overlap and increases to the expense that differences between both functions grow. Figure 2.33(b) and Figure 2.33(c) illustrate the absolute position errors of the estimated distortion centers in original images over the number of calibration images. This error is determined by computing the differences between the known and the estimated distortion center. It is shown that a growing number of calibration images and chessboards containing many chessboard squares lead to good approximation results.

Illumination	pI	dhD	dsN	bhD	bsN
Chessboard detection	6.3853	7.3108	8.9477	7.2849	7.3001
Marker detection	0.1797	0.2211	0.1890	0.2330	0.2599
Corner extraction	5.1161	5.1274	5.3459	5.1679	5.4109
Corner order	0.0056	0.0056	0.0056	0.0056	0.0056
Total time	11.6867	12.6649	14.4882	12.6914	12.9765

Table 2.3: Execution times for chessboard corner extraction on a 2.2GHz AMD-Phenom processor for variously illuminated calibration images (ODVS with $k = 7.55$, $07x09$ squares) using Matlab. Unit: [sec].

Execution time

Finally, experiments were conducted to determine the execution time required to extract chessboard corners in a single calibration image. The chessboard corner extraction algorithm is implemented in Matlab so that the times for execution relate to execution times in Matlab. In a first setup, the execution times for chessboard corner extraction are measured for variously illuminated calibration images. Therefore, calibration images were captured from chessboard patterns containing $07x09$ chessboard-squares using a low-resolution omnidirectional camera with 640×480 pixels. Table 2.3 gives an overview of the measured execution times over a set of 40 calibration images on a 2.2GHz AMD-Phenom quad-core processor. It might be seen, that the execution time increases for chessboard detection in dark and noisy calibration images. But once the chessboard region is extracted, the execution times for marker detection, corner extraction and corner order remain approximately constant.

In a second setup, the execution times are determined for chessboard extraction using well-illuminated calibration images captured by both a low and a high resolution camera. The calibration images are also captured from chessboard patterns that have different numbers of chessboard squares. In this experiment, the influence of image size and the number of chessboard squares on the extraction time is analyzed. Table 2.4 illustrates the execution times for a single calibration image on a 2.2GHz AMD-Phenom quad-core processor using a Matlab environment. The left part of Table 2.4 represents the execution times for low resolution calibration images and the right part illustrates the execution times for high resolution calibration images. Table 2.4 illustrates that the execution time for chessboard, marker and corner detection increases when increasing the image size and the number of chessboard squares in calibration images. By contrast, the execution time for determining the corner order is independent of the image size and depends only on the number of chessboard squares in calibration images.

2.6.2 Pixel density

In this section, different projections presented in Section 2.4 are evaluated and the properties of the pixel density introduced in Section 2.5 are analyzed. The characteristic of the pixel density depends on the chosen region of interest, on the mirror and camera configuration and on

Resolution	640x480			1024x768		
Num. of squares	05x07	07x09	09x11	05x07	07x09	09x11
Chessboard detection	6.4187	6.6266	6.7187	20.5800	21.4655	22.4871
Marker detection	0.2178	0.2223	0.2294	0.5382	0.5579	0.5536
Corner extraction	4.5492	5.2517	6.2363	5.2460	6.9984	8.7275
Corner order	0.0040	0.0056	0.0110	0.0042	0.0056	0.0098
Total time	11.1897	12.1062	13.1954	26.3684	29.0274	31.7780

Table 2.4: Execution times (single image) for perfectly illuminated calibration images captured by low and high-resolution omnidirectional cameras. Different numbers of chessboard squares are used in the calibration images, and execution is performed on a 2.2GHz AMD-Phenom processor using Matlab. Unit: [sec].

the chosen projection. Images from simulated omnidirectional cameras in synthesized environments and images taken from real omnidirectional cameras in the laboratory are used to analyze and to discuss the pixel density. For both synthesized and real images, various omnidirectional cameras are designed that are based on the single point of view theorem of Baker *et al.* (see Section 2.2.2). The cameras have different mirror characteristics (k), varying distances (c) between the projection center and the pinhole of the perspective camera, and use diverse fields of view (FOV) (α) for the perspective camera.

Table 2.5 gives an overview of the tested parameters for the simulated omnidirectional camera. Secondly, particular regions of interest (ROI) were defined that monitor areas below the camera and 3D-scenes with short and large distances to the camera system. All these regions provide the same vertical and horizontal number of pixels for all camera configurations in order to easily analyze and evaluate the properties of the pixel density. Original images are transformed into panoramic images for each camera configuration, and the pixel density is computed for each region of interest with different projections using real and synthesized images. However, the difference in the characteristics of the pixel densities between real and synthesized images was less than 0.3 pixels, so that corresponding curves overlap. Therefore and for a better understanding, only synthesized images are presented and discussed in this section.

Rotationally symmetric projections

The projection areas such as conical, cylindrical and spherical projection are chosen in such a way that the projection area is rotationally symmetric with respect to the z-axis of the camera coordinate system (see Figure 2.6(a)). Consequently, the pixel density σ remains constant for all numbers of columns and varies only along one row in rectified images. The characteristic of the pixel density is analyzed for several camera configurations having a constant mirror curvature k and different distances c between the projection center of the mirror and the pinhole of the perspective camera (see Table 2.5, scenario 2 and 3). In scenario 2, the field of view of the perspective camera is chosen to remain constant for all tested distances c between the camera and the mirror. For each test case in scenario 3, the field of view of the perspective camera is adjusted to the mirror size to obtain best utilization of sensor pixels for original images. The

Scenario	c (mm)	k	α (degree)
1	50	5,50	43,00
	50	7,55	39,50
	50	10,00	36,75
2	50	7,55	39,50
	75	7,55	39,50
	100	7,55	39,50
3	50	7,55	39,50
	75	7,55	28,50
	100	7,55	22,25

Table 2.5: Overview of the tested parameters for an omnidirectional camera.

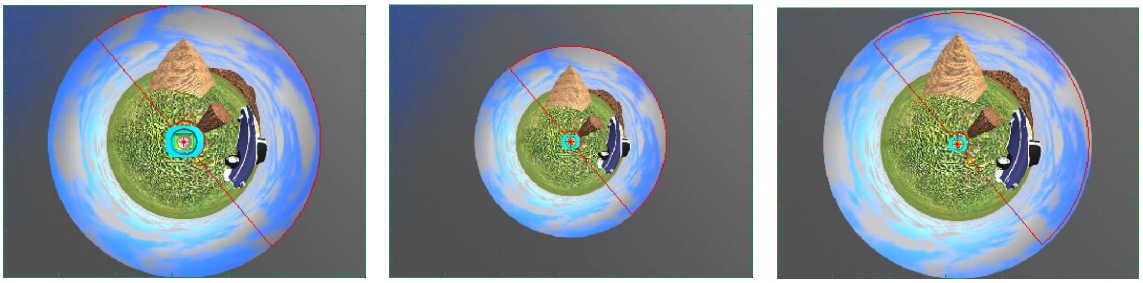
property of the pixel density is then studied for omnidirectional cameras with single point of view and constant distances c between the camera and the mirror (scenario 1).

Properties of the pixel density for cameras with fixed mirror constant and varying distances

In a first setup, the field of view of the perspective camera remains constant whereas the distance c between the projection center of the mirror and the camera pinhole varies. Figure 2.34 illustrates images captured by a synthesized omnidirectional camera. The omnidirectional cameras used in this setup are characterized by a mirror with a fixed characteristic $k = 7.55$, by a constant field of view and different distances ($c = 50mm, c = 100mm$) between the pinhole of the perspective camera and the projection center (see Figure 2.34(a) and Figure 2.34(b)). In a second setup, the field of view of the perspective camera is adjusted to the distances c and to the border of the mirror to obtain best pixel utilization of sensor pixels in original images. Figure 2.34(c) illustrates an image captured by an omnidirectional camera with the mirror constant $k = 7.55$ and the distance $c = 100mm$ between the mirror and the projection center. In contrast to previous camera configurations, the field of view is adapted to the mirror size. The regions of interest (displayed in red) are chosen for image rectification and are, hence, suitable for computing the pixel density for analysis. Figure 2.35 illustrates the panoramic images obtained for the second setup.

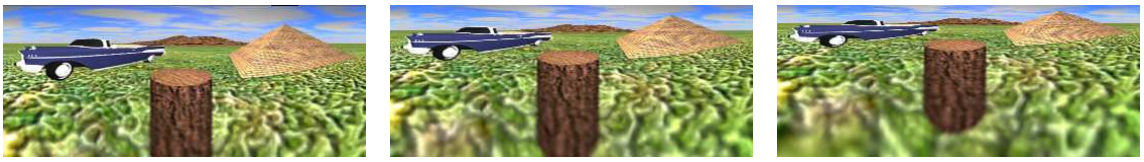
Figure 2.36 illustrates the characteristics of the pixel densities of panoramic images for the tested camera configurations for scenario 2 and scenario 3. Thus, the charts in the first row of Figure 2.36(a), Figure 2.36(b) and Figure 2.36(c) illustrate the characteristics of the pixel densities in panoramic images were obtained by omnidirectional cameras. These use constant fields of view and different distances d simultaneously. The charts in the second row illustrate the characteristics of the pixel densities in panoramic images obtained by cameras with varying fields of view.

It is shown that, for any increasing distance c between the mirror and the pinhole of the perspective camera, the values of the pixel density decrease for all projections when using perspective cameras with a constant field of view (see Figures 2.36(a), 2.36(b), 2.36(c)). In other words, an



(a) $c = 50mm$, $FOV = 39,5°$ (b) $c = 100mm$, $FOV = 39,5°$ (c) $c = 100mm$, $FOV = 22.25°$

Figure 2.34: Synthesized images captured by an ODVS with constant mirror $k = 7.55$ and various distances c using constant and adapted FOV's.



(a) $c = 50mm, k = 7.55$ (b) $c = 75mm, k = 7,55$ (c) $c = 100mm, k = 7,55$

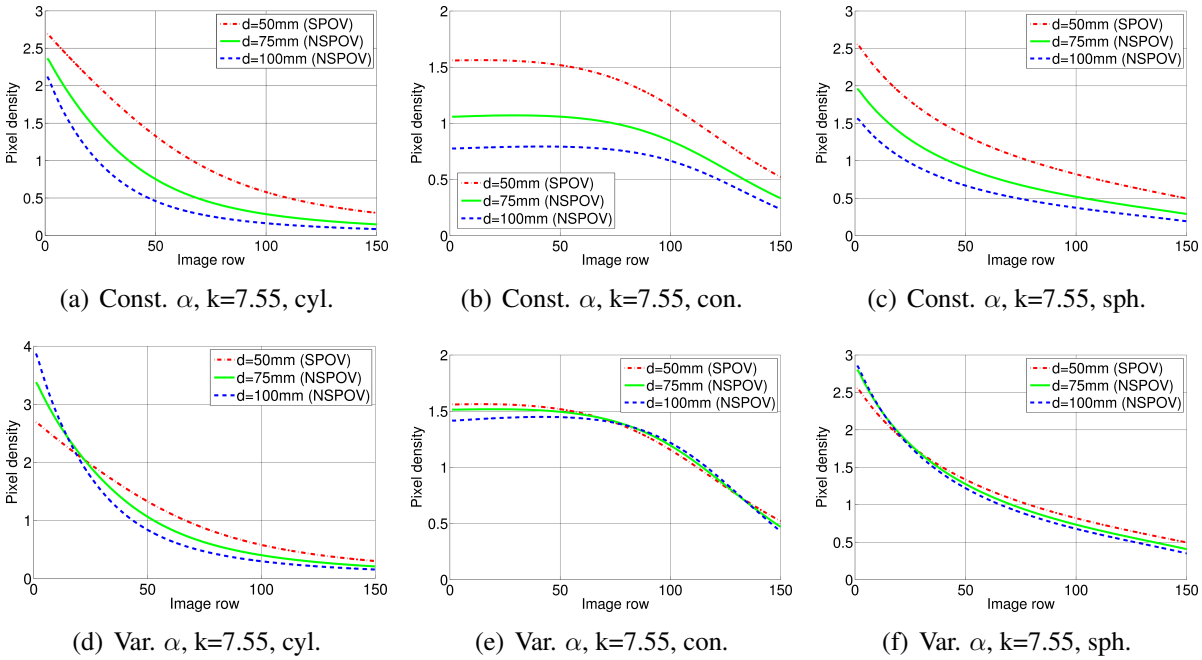
Figure 2.35: Panoramic images (cylindric projection) from an omnidirectional camera with fixed mirror constant k, different distances c between the pinhole of the camera and the projection center of the mirror and the adjusted field of view.

increasing distance c and a constant field of view of the camera lead to a reduction of sensor pixels available for original images projected by the mirror. This phenomenon is independent of the chosen projection, whereas a decreasing pixel density is more significant for the outer image regions than for inner image regions of original images.

When the field of view of the camera is adapted to distance c in order to obtain good utilization of sensor pixels, the pixel density in panoramic images is nearly identical for identical projections. The small differences, which can be seen in Figure 2.36(d), Figure 2.36(e) and Figure 2.36(f), result from the distance-based variance of the vertical field of view of the omnidirectional camera. In other words, the vertical field of view of an omnidirectional camera increases when the distance c between the camera and the mirror is enlarged. Figure 2.35 illustrates the differences of the pixel densities in panoramic images captured by omnidirectional cameras with short and with large distances c.

Omnidirectional cameras with single point of view and different mirror constants

In a third setup, experiments were conducted to analyze the influence of the projection on the pixel density. Therefore, images were captured by both synthesized and real omnidirectional cameras. These cameras have different mirror configurations ($k = 5,50$, $k = 7,55$, $k = 10,00$) and fulfill the single point of view theorem (see Section 2.2.1). Figure 2.37 illustrates original



(a) Const. α, k=7.55, cyl. (b) Const. α, k=7.55, con. (c) Const. α, k=7.55, sph.

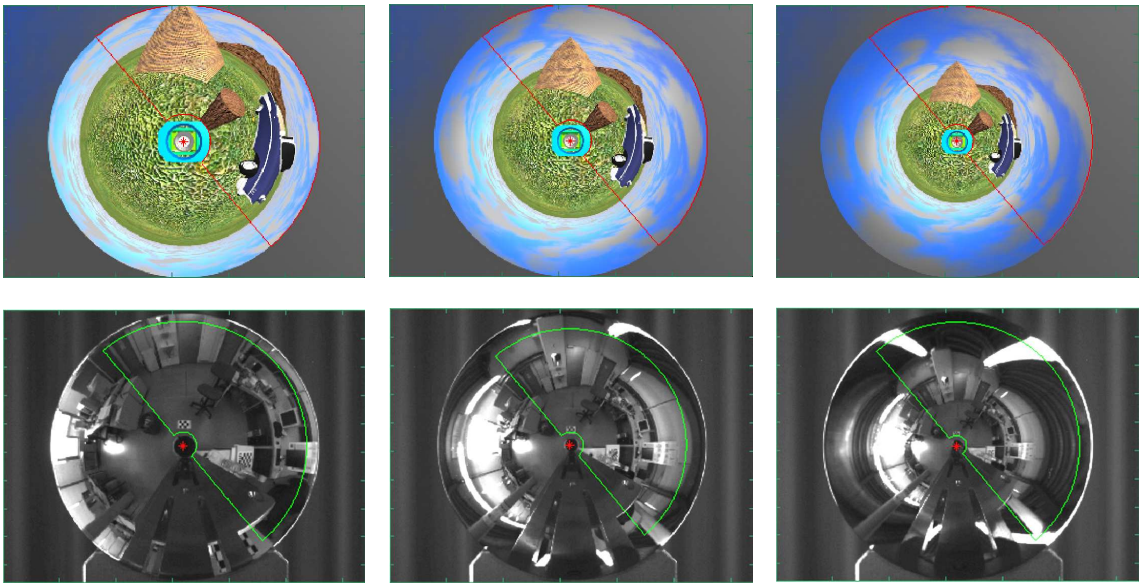
(d) Var. α, k=7.55, cyl. (e) Var. α, k=7.55, con. (f) Var. α, k=7.55, sph.

Figure 2.36: Pixel densities for panoramic images obtained from cameras with fixed fields of view (top) and adapted fields of view (bottom) are computed for all cylindrical, conical and spherical projections. For all projections, the cameras have different distances $c = 50mm$, $c = 75mm$ and $c = 100mm$ between the camera and the mirror projection center.

images captured by the cameras with different mirrors. For both synthesized and real images, a region of interest is specified and the images are transformed into panoramic images using cylindric, conic and spheric projection. In a next step, the pixel density for all camera configurations is calculated. Figure 2.38 illustrates the resulting characteristic of the pixel density.

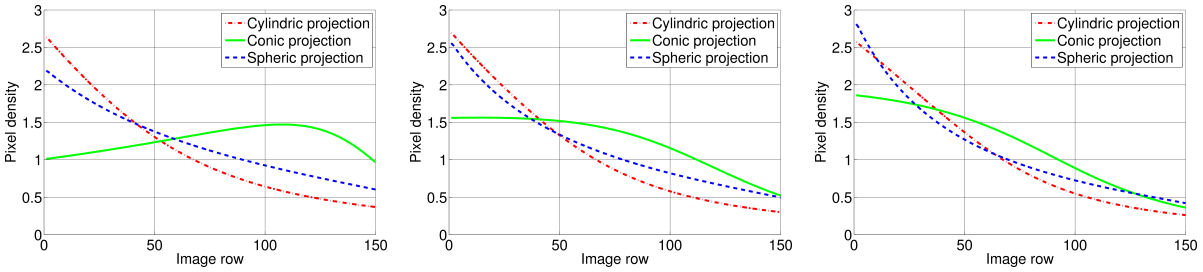
For all camera configurations, the pixel density for the cylindrical projection varies in a large range ($\sigma \in [0.3, 2.8]$) compared to other projections. This results in a high resolution in the upper regions of rectified images, but in a very poor resolution in the lower parts of panoramic images. In contrast to the cylindrical projection, the pixel density for the conic projection varies the least for all tested camera configurations. Consequently, the resolution in rectified images is more uniform and the sensor pixels are better distributed in panoramic images. However, the conic projection leads to strong distortions in panoramic images. A good compromise is the spherical projection with few distortions and a nearly homogeneous utilization of sensor pixels in rectified images. For this reason, the characteristic of the pixel density also demonstrates that the commonly used cylindrical projection is not the best projection for image rectification due to the large variations in its characteristic compared to other projections (see Figure 2.38).

Figure 2.39 illustrates panoramic images obtained from original images captured by real omnidirectional cameras. The characteristics of the pixel densities σ are determined for the omnidirectional cameras having different mirror configurations and fulfilling the single point of view theorem. The resulting pixel densities are presented in Figure 2.38.



(d) $k = 5.50$ (e) $k = 7.55$ (f) $k = 10.00$

Figure 2.37: This figure illustrates both synthesized and real images from ODVS with SPOV for different mirror configurations. The marked regions of interest are used to transform original images into panoramic images and to compute the pixel density for these images.

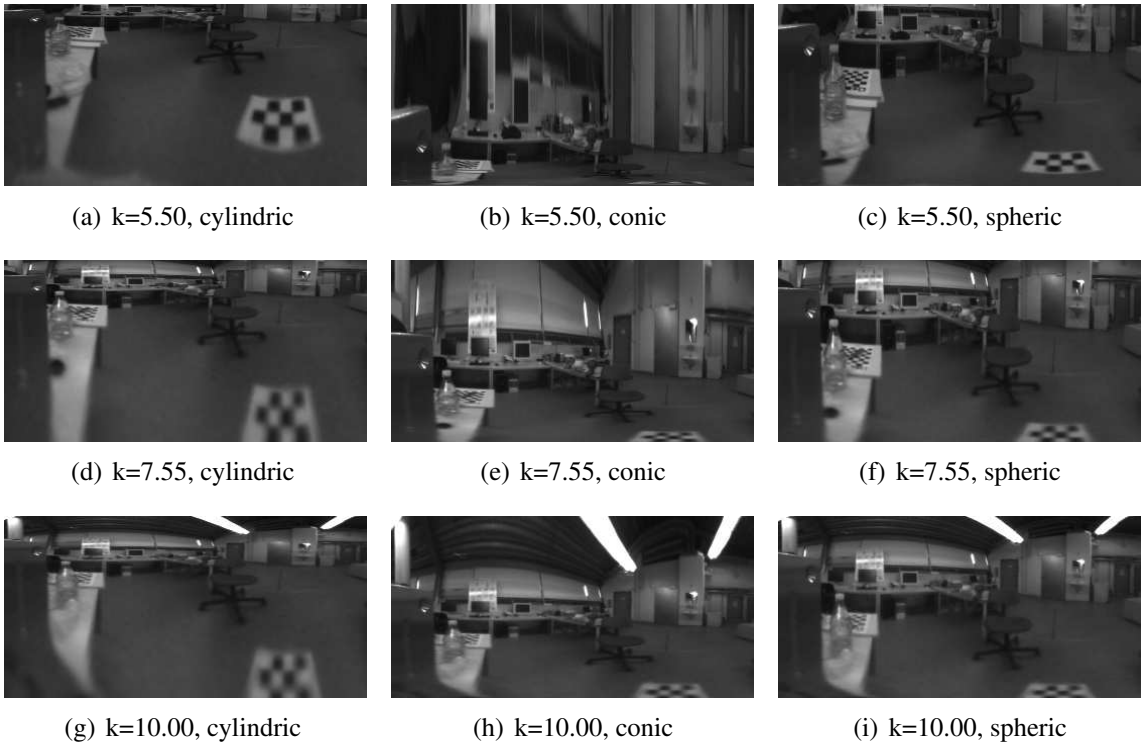


(a) k=5.50, SPOV (b) k=7.55, SPOV (c) k=10.00, SPOV

Figure 2.38: Pixel densities of panoramic images obtained from omnidirectional cameras having different mirrors and fulfilling the single point of view theorem.

Plane projection

In comparison to the conic, cylindric and spheric projections, the plane projection is not rotationally symmetric. Consequently, the pixel density is not constant over all number of columns in a rectified image. Figure 2.40 illustrates an original image captured by an omnidirectional camera with the mirror constant $k = 7.55$ and the transformed image using plane projection. It can be seen, that the plane projection can produce images as if taken directly by a perspective camera (see Figure 2.40(b)).

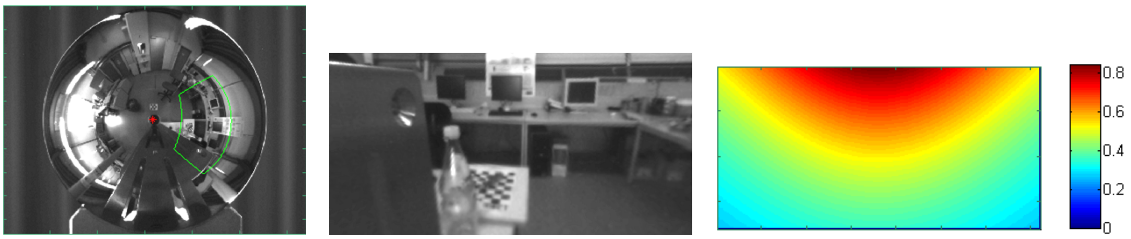


(a) k=5.50, cylindric (b) k=5.50, conic (c) k=5.50, spheric

(d) k=7.55, cylindric (e) k=7.55, conic (f) k=7.55, spheric

(g) k=10.00, cylindric (h) k=10.00, conic (i) k=10.00, spheric

Figure 2.39: Panoramic images obtained by omnidirectional cameras with different mirror configurations $k = 5.50$, $k = 7.55$ and $k = 10.00$ using cylindric, conic and spheric projection.



(a) Original image (b) Rectified image (c) Pixel Density

Figure 2.40: Original image with highlighted region of interest (a). Rectified image using plane projection (b) and its characteristic of the pixel density (c).

Execution time for image rectification

The proposed rectification algorithm (see Section 2.4) can transform original images captured by omnidirectional cameras into panoramic images. These images are a prerequisite for extracting drivers and estimating their body heights. Due to this, image rectification must be suitable to be performed online. The image rectification process is based on a Look Up Table to enable an efficient implementation in C or C++ and, hence, to obtain fast execution times. Table 2.6 gives an overview of the execution time for image transformation required by an optimized C-implementation running on a $2.2\ GHz$ AMD 64 X2 4200+ processor.

Rectified Image Size	480×204	720×306	960×408
Nearest Neighbor Interpolation	3.25 ms	4.84 ms	6.61 ms
Bilinear Interpolation	4.53 ms	6.81 ms	9.05 ms
Bicubic Interpolation	6.35 ms	9.62 ms	13.28 ms

Table 2.6: Execution times required to transform original images into panoramic image by means of an optimized, C-implemented rectification algorithm (Image size: 640×480 pixels).

2.7 Discussion

In Section 2.3.1, an algorithm is proposed to automatically extract chessboard corners in calibration images. This algorithm is highly robust against variously illuminated calibration images and is an extension to the calibration process that has been proposed by Scaramuzza. In that algorithm, chessboard corners have to be selected manually during the calibration process. But manual selection of chessboard corners is very time consuming and cannot calibrate omnidirectional cameras in the automotive domain. At the same time first results have been presented in [16] and [18], another extraction algorithm has been presented by Rufli *et al.* in [57] that automatically extracts chessboard corners in calibration images captured by omnidirectional cameras. A comparison to those techniques demonstrated a similar detection rate for perfectly illuminated images. Chessboard corner detection in calibration images captured under different illumination conditions, however, led to a detection rate two times lower than in our algorithm proposed in Section 2.3.1. An advantage of the algorithm proposed by Rufli *et al.* is the use of pre-compiled C-based code that drastically reduces the time for execution. But code optimization and, hence, the speeding up of the processing time for chessboard corner extraction using c-functions are potential tasks for future work.

In Section 2.5, the pixel density was introduced as a measurement parameter to compare different projections such as conical, cylindrical and spherical projection and to evaluate mirror camera configurations of omnidirectional cameras. In this manner, the pixel density supports users in selecting the optimal projection and projection parameters in terms of best utilization of sensor pixels in panoramic images. Best utilization of sensor pixels in panoramic images and, therefore, best resolution can be obtained when the pixel density is close to unity for all pixels in panoramic images. Values of the pixel density close to unity signify a direct mapping of intensity information from one sensor pixel position to the corresponding pixel position on the target image. A pixel density less than unity denotes poor resolution since the information of one sensor pixel is mapped to several pixel positions in panoramic images. Values larger than unity denote good resolution but also mean a waste of sensor pixels since intensity information of several sensor pixel positions is projected onto the same pixel position in the target image. This may also lead to a loss of resolution (for source to target mapping) due to an overlap of intensity information at one pixel position.

Furthermore, the projection with least variance in the pixel density for different mirror constants k and for various distances c between the camera pinhole and the projection center is recommended for rectification. In general, the conic projection seems to be the best projection due to

fewer variances in the characteristic of the pixel density and due to values close to unity. However, the conic projection leads to highly distorted panoramic images and might be a problem for image processing routines. In this case, the spherical projection can be a good alternative to the conic projection. The cylindrical projection may also be a good projection to transform original images into panoramic images if only small regions in original images are required for image transformation. Thus, the pixel density is suitable for choosing the projection with best utilization of sensor pixels in panoramic images.

Similarly to rotationally symmetric projections, the pixel density is also suitable for finding optimal projection parameters for plane projections. The parameters of the target projection planes can be carefully chosen to obtain only a small variation of the pixel density across the whole image.

2.8 Conclusion

In this chapter, the geometry and properties of omnidirectional cameras are presented including the mathematical representation of the camera. In particular, central projection cameras, i. e. omnidirectional cameras having a single point of view, are highly desirable as they can generate perspectively correct panoramic images. The known epipolar geometry of perspective cameras can be adapted to omnidirectional cameras in order to generate 3D-ambiance information of the area surrounding the car door. This chapter also presents the camera calibration process [30] especially targeting omnidirectional cameras to obtain the parameters of the camera model. The camera model is a prerequisite for transforming original images taken by the camera into panoramic images. In Section 2.3.1, a novel extension to the calibration scheme is proposed to enable robust, automatic extraction of chessboard corners in calibration images to perform camera calibration for applications in the automotive domain. The locations of chessboard corners in calibration images serve as an input to the calibration procedure to estimate the camera model. The proposed extraction algorithm is also extended to gain the robustness of chessboard corner extraction for variously illuminated calibration images.

This thesis also surveys commonly used projections and proposes an algorithm to transform original images captured by omnidirectional cameras into panoramic images (see Section 2.4). In Section 2.5, a new value – the pixel density – is proposed as a new tool to compare various projections for image transformation and to evaluate camera/mirror configurations in terms of best utilization of sensor pixels in panoramic images. In this manner, best utilization of sensor pixels in panoramic images can be obtained for any camera configuration. It is also shown that the commonly applied cylindrical projection is not suitable for some omnidirectional cameras due to its large variances of the pixel density compared to other projections. In this thesis, the pixel density as a tool for comparing different projections and camera configurations has first been proposed in the field of camera calibration and image transformation.

In the remainder of this thesis, the camera model presented in Section 2.2.4 and the image rectification method proposed in Section 2.4 are used to estimate the body-heights of approaching drivers and to generate 3D-ambiance information using motion stereo algorithms.

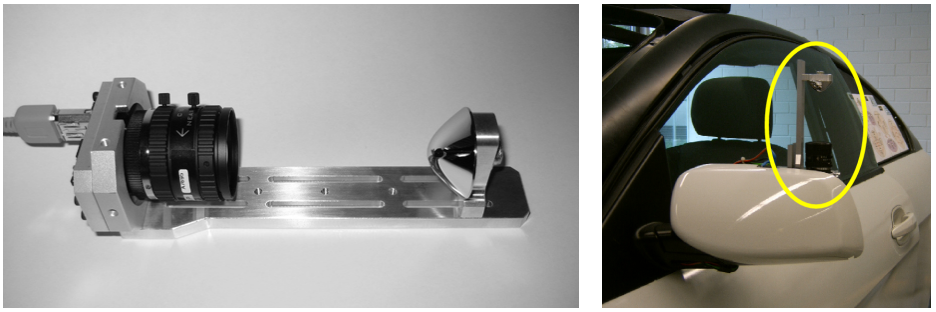
3 Driver body height estimation

3.1 Introduction

Nowadays, passenger comfort related issues are active research topics in the area of automotive ergonomics. maximizing passengers comfort in today's cars is gaining importance in the domain of automotive systems engineering, in particular for ingress/egress to/from a car in narrow parking situations. Studies in automotive ergonomics illustrate a strongly increasing level of comfort during ingress when there is an automatic adjustment of seat position according to driver height. For this purpose, automatic passenger seat adjustment has recently attracted a lot of attention [58, 59, 60, 61].

However, one drawback of known solutions lies in storing individual driver height in the car system or in a personal key. This results in a number of problems: Storing the driver's height is not feasible for rental cars. Furthermore, accidents may happen if a tall person mistakenly uses the key of a shorter one and the system adjusts the seat according to the height of the shorter person. To overcome these limitations and to provide individually adjusted seat positions according to driver height, this chapter introduces a new method of estimating the height of approaching car drivers based on image data of a single omnidirectional camera. The camera is integrated with the side-view mirror of a car and can monitor the surroundings next to the car in theirs entirety and, hence, to extract drivers approaching from anywhere due to its very large field of view.

Figure 3.1 illustrates such an omnidirectional camera (see Figure 3.1(a)) that is integrated with the side-view mirror of a car (see Figure 3.1(b)). Height estimation is subdivided into two pro-



(a) (b)

Figure 3.1: Omnidirectional camera (a) that is integrated with the side-view mirror of a car (b) to estimate the height of approaching drivers.

Figure 3.2: General framework for people extraction and height estimation.

cessing stages: First, the proposed algorithm extracts approaching drivers in panoramic images captured from the environment next to the car by separating persons from the background. Sets of foot and head points are determined for approaching drivers on which further processing stages are based. Second, the algorithm initially estimates the tilt of inclined parked vehicles based on gathered samples of head and foot points. Tilt of inclined parking vehicles strongly influences body height estimation and has to be considered when computing body heights of approaching drivers. Thereafter, an iterative optimization process removes outliers in the input data, refines the tilt of inclined parked cars and computes the body heights of approaching drivers. Then, the estimated height is used to ergonomically pre-adjust the seat according to driver body height to ease ingress. This method enables absolute body height estimation for a wide range of parking scenarios without knowledge of passengers or geometrical information of the surroundings. In particular, body height estimation is suitable for rental cars and may avoid accidents by mistakenly chosen keys.

3.2 Related work and contributions

An effective algorithm to automatically extract drivers and to estimate their body heights using a single camera consists of the following processing stages: Modeling of dynamic environments, object segmentation, object classification and tracking and body height estimation based on gathered input data from the camera. Figure 3.2 illustrates a high level overview of the general framework to extract people and to estimate their body heights. Object and motion detection is essential for nearly every surveillance tasks and people extraction system and aims to separate regions that contain objects like humans or cars from the background in an image. This process commonly includes environment modeling, object and motion detection, object clustering and classification. The result of the subsequent absolute body height estimation depends to a large extent on the quality of the regions in which humans are extracted. Incomplete regions may lead body height estimation to fail in particular when the regions for foot and head point determination have not been extracted. The first part of this chapter focuses on precise region extraction of the image regions that is a prerequisite to precisely determine body heights of approaching drivers.

3.2.1 Environment modeling and object detection

Each object extraction algorithm requires constructing and updating of an environment model to permit object detection and classification. Hu *et al.* [62] provide a very interesting survey about current techniques for environment model-based surveillance. In general, one can distinguish between two common environment models. 2D-models that are based on image coordinates

	Initia-lisation	Prior Knowl.	Static Objects	Obj. Size	Pers-pec-tive	Paral-leliza-bility	Com-plete-ness
Template Matching	no	needed	ind.	bad	bad	middle	middle
HMM	no	needed	ind.	bad	bad	middle	high
Features	no	needed	ind.	bad	bad	good	middle
Optical Flow	no	no	bad	ind	ind.	good	low
Background Estimation	needed	no	ind.	ind	ind.	good	high

Table 3.1: Overview of object extraction methods used for people detection. *Background estimation* seems to yield the best results for this application (*ind: independent*).

and 3D-models that are based on real world-coordinates. Current work of 3D-models is limited to indoor scenarios [63, 64] due to difficulties in precise 3D-reconstructions of outdoor scenarios [62]. Moreover, 2D-models are easier to implement and are, hence, more common in state-of-the-art applications such as surveillance or traffic monitoring. There are three types of 2D-models that are especially targeted to the following three scenarios. One 2D-model for static cameras, one that is feasible for pure translation cameras and one for mobile cameras. In this thesis, the vehicle and the car door are stationary during the time of height estimation. Therefore, a background detection algorithm is sufficient that is based on a 2D-environment model under the assumption of a static camera. However, when dealing with moving cameras, the work of Friederich [13] may be used to estimate the ego-motion of the camera and to compensate the shift in the background.

A number of techniques for object detection and extraction exist in the image processing literature, viz., Hidden Markov Models (HMM) [65], Template Matching [66], feature-based detection methods, optical flow-based methods [67],[68] and background separating methods. Table 3.1 gives a brief overview of potential methods that can extract people in images and compares them in terms of computation time, their abilities for parallelization in order to realize fast people extraction, their abilities to handle perspective changes and to detect human regions completely (completeness). Background estimation and optical flow are not sensitive to changes of perspective and to changes of object size. Furthermore, they do not need prior knowledge of the object's property e.g. color, shape and geometry. One disadvantage of optical flow is that it can not detect static or very slowly moving objects. Although background estimation solves this problem, it needs a small time interval for initialization to learn the background. Fortunately, in the presented application such an interval is available, viz., the time interval between activating body height estimation and the first door operation. Moreover, background estimation leads to very precise detection results regarding completeness of extracted humans. Hence, one focus of this chapter is on real-time people extraction algorithms using background estimation-based techniques. Though initialization is required, the advantage of highest completeness of regions compared to other algorithms prevails.

3.2.2 Foreground detection

The problem of extracting objects from a video sequence has been widely studied in surveillance [69], traffic monitoring [70] and vehicle guidance. In most applications, separating the foreground from the background is the first step in object tracking. Background subtraction and foreground modeling are powerful methods whose advantages are feature-independent segmentation (e.g., textures, direction of move, speed). Other common techniques for background subtraction include Kalman filtering [71], kernel density estimation [72], hidden Markov models [73], mixture of Gaussians [74], [75] and the use of color-based intensity independent features [76]. Most of these algorithms represent each background pixel using a probability density function (PDF) and classify pixels from new images as background depending on the description of pixels by means of their density functions. As an alternative, Bhaskar *et al.* [77] developed a foreground detection algorithm using cluster density estimation based on a Gaussian mixture model. This algorithm is suitable for handling illumination changes as well as dynamic backgrounds. Similar work was done in [78] using Kalman filtering to iteratively estimate the dynamic background texture and the regions of foreground objects. Kalman filtering was also used by Karmann *et al.* [71] to model the background dynamics of each pixel by choosing two different gains, thereby allowing fast adaptation of background changes and slow adaptation of foreground pixels. Ridder *et al.* [79] improved this approach and presented a shadow detection method assuming small differences between overshadowed and non-overshadowed backgrounds. However, strong shadows caused by direct sunlight cannot be detected. Although many background subtraction techniques have been proposed, the majority of the algorithms address shadow detection and illumination compensation by exploiting color information (see [72, 78]).

In scenarios where monochromatic video cameras are used – such as the presented scenario – the existing methods are no longer suitable. The camera system used for people extraction and height estimation consists of a monochromatic VGA camera that is designed for (cost sensitive) applications in the automotive domain. Costly, high resolution color video cameras may be useful for research, but impractical for real applications. For applications, where only monochromatic cameras are available, a common method to increase the robustness of image processing algorithms is the transformation of intensity based images into lighting invariant frames. This transformation, e.g. based on Census or Rank filtering [80, 81] is widely used, but intensity information of homogeneous regions will be lost. In other words, it may be no longer possible to distinguish between homogeneous foreground (cars, trucks) and background regions (walls). Therefore, the use of intensity-based images is highly desirable for the presented body height estimation application.

But intensity based monochromatic images lead to several challenges in object detection. For example, it is difficult to differentiate between small illumination changes caused by shadows or by small, valid foreground objects in gray scale images. Another challenge is to detect small objects in low resolution images captured by an omnidirectional vision system. These problems, along with the accuracy of background subtraction, the handling of sudden illumination changes and the possibility of parallelizing algorithms are the underlying motivations for developing an extended people extracting algorithm as presented in this thesis. It also describes extensions

to background estimation (e.g., illumination compensation and shadow elimination for gray-scale images) that are specifically tuned to the setting of the smart car door equipped with an omnidirectional camera. Figure 3.3 gives a high level overview of the proposed system, and the details of the algorithms for driver extraction are described in what follows.

Inspired by the background estimator of Ridder *et al.* [79] and by the shadow detector proposed by Jacques Jr. *et al.* [82], robust background estimation and foreground detection algorithms for gray scale images have been developed in this thesis. Ridder *et al.* proposed an extension to their algorithm that takes weak shadows from stationary or moving objects into account. They assume that weak shadows have the same characteristic as illumination changes and may hence be adapted by the background. To avoid adaption of illumination changes into the background, their algorithm automatically increases the threshold for foreground detection using the variance of estimated background values over time. The threshold is high if the variance of the estimated background values (e.g. caused by shadows) is high. However, pixels from small foreground objects – such as humans which are far away from car – also cause a high variance and may be suppressed. Moreover, strong shadows cannot be identified in [79], as they are detected as valid foreground. In other words, once detected as foreground, it is impossible to differentiate between shadow and foreground. A good shadow detector for gray scale images was introduced by Jacques *et al.* [82] using normalized cross correlation (NCC). The detector assumes shadow pixels as scaled versions (darker) of the corresponding background pixels, so that the NCC in a neighboring region is close to unity. On the other hand, the shadow detector misclassifies valid foreground pixels with small differences as shadow pixels.

To overcome these limitations, this thesis combines and modifies the background estimation method proposed by Rider *et al.* [79] and the shadow detection algorithm presented by Jacques *et al.* [82] to design a powerful background subtractor. This thesis also extends the shadow detector with the zero means cross correlation (ZNCC) in order to distinguish between shadows and valid foreground pixels. Furthermore, the proposed algorithm detects illumination changes using local search windows and updates the background to compensate for slow or sudden illumination changes. The proposed algorithm is also evaluated and compared with a background estimator based on Gaussian Mixtures, with the approach of Jacques Jr. *et al.* [82], and with the approach of Ridder *et al.* [79]. Experiments in complex outdoor and indoor environments under various lighting conditions demonstrated good foreground detection and high robustness of the proposed algorithm against shadows and illumination changes. The algorithms are also evaluated for their ability for parallelization and are compared in terms of sequential and parallel implementations (on an AMD Quad-Core CPU). The results show that the proposed algorithm runs in real-time and is thus suitable for implementation on embedded platforms such as an FPGA.

3.2.3 Driver detection and body height estimation

Tracking of people using vision sensors is an important requirement for a growing variety of applications such as activity recognition [83], pedestrian detection in the vehicle surroundings [84, 85], gait analysis [86] and estimation of anthropometric data like height and size of athletes or passengers [87, 88, 89]. Cupillard *et al.* [90] propose a three-stage algorithm to track groups

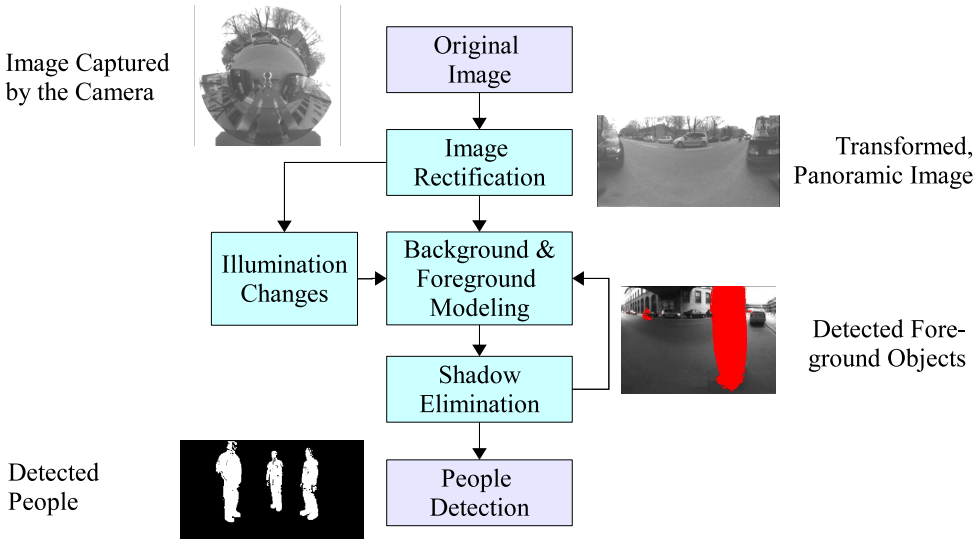


Figure 3.3: Block diagram for object detection using background estimation and foreground detection, shadow compensation and handling of illumination changes.

of people in a metro scene in order to prevent violence or vandalism. Their algorithm is tuned to recognize abnormal behavior. Detection is based on the extraction of moving regions using background subtraction, feature extraction such as position, center of gravity, height and width of an object, and on region classification. An interpretation module merges all features and detects violence and vandalism on the basis of abnormal behavior.

Another interesting approach is an image-based identification of humans based on their shape and gait. For this purpose, Collins *et al.* [91] established a multi-baseline-based method to identify humans based on their body shape and gait. Identification is performed by a viewpoint-dependent technique that requires up to 8 cameras. To achieve this, key frames of walking humans are compared with frames obtained in training sequences. The training sequences include biometric shape cues such as body height, proportions of the body, stride length and amount of arm swing. Additional thereto, an increasing effectiveness of personal identification has been proven by BenAbdelkader *et al.* [92], when height and the quantification of the biometric parameters of gait are also included in the identification process. In both approaches, height estimation is performed using a camera network.

Another common approach for estimating body heights is the use of a stereoscopic, camera-based surveillance system mounted on the ceiling of a room or on the high position of a building. In this manner, Izumi *et al.* [93] proposed an algorithm to estimate the height of a person by estimating the distance between the head of a person and the stereo camera system.

Height prediction using only a single camera was introduced by Bovyrin *et al.* [94]. They present a robust method to detect 3D-road maps and human heights using a single perspective camera. One limitation of this method is the prerequisite of a top-down view to a distant scene. However, a top-down view of a distant scene is difficult to realize in automotive applications. Moreover, the road map and the body height can be estimated only up to an unknown scale factor. To overcome the limitation of unknown scale factors in body height estimation, Danilo

et al. [88] propose a method to estimate the absolute body height for applications in the forensic anthropology by using a single camera and fixed reference points in a medical room. Similar work was done by Criminisi *et al.* [95]: Using at least one reference height in an unknown scene and a minimal calibration of the camera (the vanishing point(s) in images must be known), all other heights in a scene can be derived from this reference height.

Much research has been done for foreground classification, people detection in vehicle surroundings, human identification from body shape and gait and detection of abnormal behavior. To the best of our knowledge, little work has been done for absolute human body height prediction using a single omnidirectional camera. In particular, little work has been done to estimate the body height of approaching drivers in the automotive domain in order to ergonomically preadjust the seat position to improve ingress/egress in tight parking lots. State of the art algorithms for absolute body height estimation have several limitations that make them impractical for automotive applications. One disadvantage is the use of multiple cameras to obtain the absolute body height of persons. Due to space and cost constraints, an approach is pursued for this application that uses only a *single* omnidirectional camera attached to each the side-view mirror of the car. Furthermore, recording training sequences to identify approaching drivers and, hence, to adjust the seat according to height data stored in the car system is not suitable for automotive applications. Moreover, it is difficult to position a camera on a commercial vehicle to obtain a top-down view of a distance scene. Body height estimation up to an unknown scale factor is feasible to identify abnormal behavior, but is impractical to ergonomically optimize the seat position for better ingress. Reference points or objects with known height in a scenario overcome the unknown scale factor problem, but this cannot be realized for any parking situation.

This chapter presents a new method for estimating absolute body heights of approaching drivers with a single omnidirectional camera. The proposed method overcomes the limitation of unknown scale factors and avoids the presence of reference points in parking scenes. For this purpose, a *car ground plane* is introduced spanned by the four wheels of a car. The distance of the camera to the car ground plane can be determined during camera calibration and is assumed to be known. Additionally, a *ground plane* is introduced on which drivers walk straight ahead towards the car. The absolute height of approaching drivers can be determined, when the orientation between the camera and the ground plane is available. Unfortunately, this information is only partially available due to missing position sensors and due to unknown parking situations. For this reason, this thesis proposes an efficient algorithm to estimate the orientation and position of the ground plane relative to the camera using sets of head and foot points from approaching drivers. With the known orientation of the ground plane and its distance to the camera, the absolute body heights of approaching drivers are computed.

The rest of this chapter is organized as follows. Background initialization, the background estimator, the shadow detector and an algorithm to compensate illumination changes and to identify approaching drivers are presented in Section 3.3. In Section 3.4, the mathematical representation for the most common parking scenarios and a generic, mathematical model are presented. The generic model is feasible for a wide range of parking scenarios. The results obtained for driver extraction and body height estimation are presented and discussed in Section 3.5. Finally, this chapter concludes by briefly outlining some possibilities for future work in Section 3.6.

3.3 Driver extraction

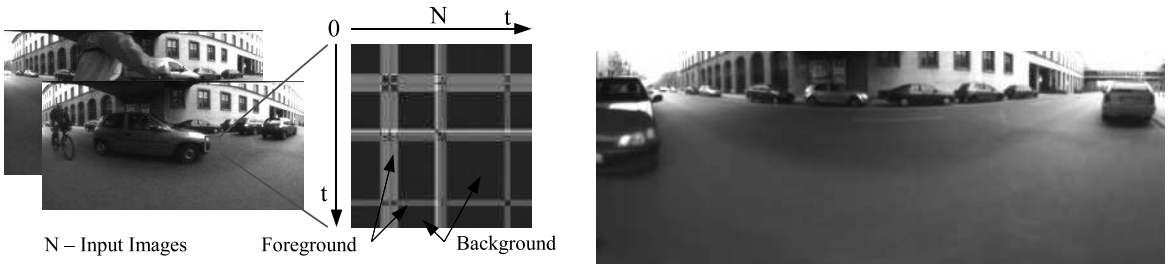
As discussed in Section 3.2.1 and Section 3.2.2, background estimation was proposed to extract drivers in panoramic images. The background model used in this thesis is based on the approach of Karman *et al.* [71] and Ridder *et al.* [79]. It is extended to provide better shadow detection and to be more robust against illumination changes for applications in the automotive domain. The mathematical details of the background model along with the shadow detector and a method to account for illumination changes are presented. Furthermore, background initialization is described in Section 3.3.1, and Section 3.3.2 presents the background model that is based on Kalman filtering [71, 79] to model the dynamics of each background pixel. Pixels are identified as background or foreground pixels using thresholding. Potential foreground pixels are classified as valid foreground or as shadow pixels using the NCC and the ZNCC (see Section 3.3.3 and 3.3.4). Finally, a method is proposed that accounts for global illumination changes in Section 3.3.5.

3.3.1 Background initialization

Many approaches addressing background estimation require the recording of a separate image from an empty scene to initialize the background model. But this is nearly impossible for applications in the automotive domain: An example besides many others is parking on a highly frequented street with many road participants. In the approach of Ridder *et al.*, each background pixel is initialized with a fixed value that is adapted during a training period using large numbers of frames. Jacques *et al.* use median-based background initialization over a large number of frames. Median-based initialization allows recording of background images in busy-street scenarios, but it also assumes that pixels contain background content for at least half of the initialization frames.

Real life experiments demonstrate that this assumption is not necessarily valid for several traffic scenarios. For this purpose, Farin *et al.* [96] propose a powerful method to solve the problem of extracting background images in highly frequented scenarios. The principal idea of their algorithm is to roughly segment each pixel from input frames into foreground and into background. The segmentation is carried out on small blocks for each pixel position from the input frames. Farin classifies background content by searching for the subsets of frames that show stable content within the blocks. In other words, the content of blocks with background varies less than the content of blocks with foreground. To find the blocks that contain background pixels, the similarity of block contents over a fixed training period is computed and stored into a *Similarity Matrix* for each pixel in an image. This matrix contains the differences (realized with the Sum of Absolute Differences, SAD) between image content at the block positions for each pair of frames. Low values in the matrix relate to background regions, whereas high values correspond to foreground regions.

Each similarity matrix is decomposed into two parts, one that may contain background (low values) and one that may contain foreground (high values). Then, the background image is computed based on the pixels that contain background content using a median-based algorithm [97].



(a) Similarity matrix of one pixel position p_i (b) Initialized background

Figure 3.4: Similarity matrix obtained for one pixel position p_i in a panoramic image (a). Slight differences between the blocks for each pair of frame are shown as dark, big differences between the blocks as bright areas. Estimated background image as an initialization for the background model (b).

For this application, such segmented background images are used to initialize the Kalman-based background model (see Section 3.3.2). Further training periods are not required. Although the method proposed by Farin *et al.* [96] is very robust for background initialization, it is very time-consuming due to the block difference calculation using SAD. Following the definition of SAD, the absolute, pixel-wise differences of the intensities $I_1(x, y)$ and $I_2(x, y)$ for pixels (x, y) in a pair of frames (1,2) are computed for a block B_i with size $m \times m$. These differences have to be computed for each image pair in a set of n frames. This leads to long computation times for background initialization depending on the number of frames in a training sequence.

For this reason, SAD for block difference calculation is replaced by block averaging. Absolute differences and the similarity of two blocks in consecutive frames are computed based on the block averages for each corresponding block. In other words, the average $\mu_n(p_i)$ is computed for each block and is stored for further computations. Previously computed values $\mu_n(p_i)$ of blocks can be reused so that only one block difference computation is necessary for each image pair in a training sequence. Clearly, nine difference computations for blocks with size of 3x3 pixels would be required using SAD. In Eq. 3.1, an efficient algorithm based on the block averages $\mu_n(p_i)$ is proposed for computing block differences and entries of the similarity matrices.

$$
\begin{aligned}
t_1 \quad &: \quad \mu_1(p_i) = mean(B_i(p_i)) \\
t_2 \quad &: \quad \mu_2(p_i) = mean(B_i(p_i)) \;,\; d_{1,2}(p_i) = |\mu_1(p_i) - \mu_2(p_i)| \\
t_3 \quad &: \quad \mu_3(p_i) = mean(B_i(p_i)) \;,\; d_{1,3}(p_i) = |\mu_1(p_i) - \mu_3(p_i)| \;,\; d_{2,3}(p_i) = |\mu_2(p_i) - \mu_3(p_i)| \\
t_n \quad &: \quad \mu_n(p_i) = mean(B_i(p_i)) \;,\; d_{j,n}(p_i) = |\mu_j(p_i) - \mu_n(p_i)| \;\; j \in [1, n-1]
\end{aligned}
\tag{3.1}
$$

For each pixel position p_i in an image of n training frames, the average $\mu_n(p_i)$ of a block B_i in the neighborhood of a pixel p_i is determined. Thereafter, the absolute differences of blocks for each pair of frames are computed using previously computed block-averages $\mu_n(p_i)$.

Clearly, the block average $\mu_1(p_i)$ for each pixel in the first frame 1 is computed at time step t_1. At time step t_2, the block averages $\mu_2(p_i)$ for each pixel in frame 2 are computed, and the absolute differences $d_{1,2}(p_i)$ are determined using stored block averages $\mu_1(p_i)$. Then, the results are stored in the similarity matrix with the elements $d_{j,n}(p_i) = d_{n,j}(p_i)$. At time step t_3, the block averages $\mu_3(p_i)$ are computed, and the absolute differences $d_{1,3}(p_i)$ and $d_{2,3}(p_i)$ are determined. Similar to time step t_2, the previously determined block averages $\mu_1(p_i)$ and

$\mu_2(p_i)$ from time step t_1 and t_2 can be re-used for further processing. However, background initialization using block averaging is less robust compared to background initialization with SAD (see Section 3.5.1), but it allows to reuse previously determined block average values $\mu_{(n-1)}(p_i) \cdots \mu_1(p_i)$. This way, the computation time of this algorithm is 10 times faster.

Such determined background images serve as an initialization for the Kalman-based background estimation and foreground detection. The number of frames required for background initialization strongly depends on the number of foreground objects in the training sequence. Ten frames would be sufficient to learn the background for empty scenarios and at least 40 frames are required to initialize the background image in highly frequented road scenarios.

3.3.2 Kalman-based background estimation

As discussed in Section 3.2, the thesis used background estimation for extracting objects of interest from the captured images. The background model is based on the approach of Karman *et al.* [71] and has been extended by Ridder *et al.* [79]. In this section, the Kalman-based background model proposed by Ridder *et al.* [79] is presented. The background model considers the dynamics of the background, e.g. slow illumination changes, and can detect foreground objects in panoramic images. The underlying Kalman filter theory is well described, e.g. by Anderson *et al.* [98] so that the Kalman filter theory is only briefly presented below. Best information of a system state is obtained by an estimation that explicitly considers noise in the measurement. Following Eq. 3.2, the estimation of the system state $\hat{s}(t_i)$ at time t_i is

$$\hat{s}(t_i) = \tilde{s}(t_i) + \mathbf{K}(t_i)[z(t_i) - \mathbf{H}(t_i)\tilde{s}(t_i)] \tag{3.2}$$

and the prediction $\tilde{s}(t_i)$ at time t_i

$$\tilde{s}(t_i) = \mathbf{A}(t_i)\hat{s}(t_{i-1}) \tag{3.3}$$

Hereby, $\mathbf{A}(t_i)$ represents the system matrix and $\mathbf{H}(t_i)$ the measurement matrix. $z(t_i)$ is called the system input that is required to estimate the unknown system state along with the Kalman gain $\mathbf{K}(t_i)$.

In this application, the system input $z(t_i)$ represents new intensity values provided by the camera system. The Kalman gain $\mathbf{K}(t_i)$ depends on the predicted error covariance and on the system input and serves as a weighting factor for the system input. In other words, if the measurement noise is high then the Kalman gain is low due to a high error covariance and vice versa. The Kalman filter runs with two filtering states that are prediction of the new system state and correction (estimation) of the prediction using the system input. At the correction state, the predicted system state is compared with the actually measured system input. The estimation of the system state is then computed by weighting the difference between the measured system state and the prediction using the Kalman gain. Thus, measured values get a lower weighting when their error covariance is high and vice versa [79, 71].

Karmann and Ridder use Kalman filtering to extract the background of images captured by a stationary camera and model the background dynamics of each pixel as follows: The intensity

of a pixel at position (x,y) at time t is given by $I_{x,y}(t)$. The estimated system state of the background is denoted as $\hat{I}_{x,y}(t)$ and its derivative is denoted as $\hat{\dot{I}}_{x,y}(t)$. The system state represents the background for each pixel in extracted input images and is estimated as follows:

$$\begin{bmatrix} \hat{I}_{x,y}(t) \\ \hat{\dot{I}}_{x,y}(t) \end{bmatrix} = \begin{bmatrix} \tilde{I}_{x,y}(t) \\ \tilde{\dot{I}}_{x,y}(t) \end{bmatrix} + \mathbf{K}_{x,y}(t) \cdot \left(I_{x,y}(t) - \mathbf{H} \cdot \begin{bmatrix} \tilde{I}_{x,y}(t) \\ \tilde{\dot{I}}_{x,y}(t) \end{bmatrix} \right) \tag{3.4}$$

Following Eq. 3.3, the prediction $\tilde{I}_{x,y}(t)$ of the system state $\hat{I}_{x,y}(t)$ and its derivative $\tilde{\dot{I}}_{x,y}(t)$ at time t is given by:

$$\begin{bmatrix} \tilde{I}_{x,y}(t) \\ \tilde{\dot{I}}_{x,y}(t) \end{bmatrix} = \mathbf{S} \cdot \begin{bmatrix} \hat{I}_{x,y}(t-1) \\ \hat{\dot{I}}_{x,y}(t-1) \end{bmatrix} \tag{3.5}$$

With the system matrix S, the measurement matrix H and the Kalman gain K is:

$$\mathbf{S} = \begin{bmatrix} 1 & s_{1,2} \\ 0 & s_{2,2} \end{bmatrix}, \mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{K}_{x,y}(t) = \begin{bmatrix} k1_{x,y}(t) \\ k2_{x,y}(t) \end{bmatrix} \tag{3.6}$$

In [71], $s_{1,2} = s_{2,2} = 0.7$ was used to model the background dynamics. Since the camera returns only intensity values $I_{x,y}(t)$, the measurement matrix H is a constant. Following Eq. 3.7 and Eq. 3.8, the Kalman gain is chosen depending on detected foreground $m_{x,y}(t) = 1$ or detected background $m_{x,y}(t) = 0$. To achieve a classification, a pre-estimation of the next system state is performed following Eq. 3.7:

$$m_{x,y}(t) = \begin{cases} 1, & if \quad \left[d'_{x,y}(t) \geq th_{bg}\right] \vee \\ & \left[(d'_{x,y}(t) < th_{bg}) \wedge \right. \\ & \left. (d''_{x,y}(t) \geq th_{bg})\right] \\ 0, & if \quad \left[d'_{x,y}(t) < th_{bg}\right] \wedge \\ & \left[d''_{x,y}(t) < th_{bg}\right] \end{cases} \tag{3.7}$$

$$\begin{aligned} d'_{x,y}(t) &= |I_{x,y}(t) - \tilde{I}_{x,y}(t)| \\ d''_{x,y}(t) &= |I_{x,y}(t) - \hat{I'}_{x,y}(t)| \\ \text{with} \quad \hat{I'}_{x,y}(t) &= \tilde{I}_{x,y}(t) + \beta \cdot \left[I_{x,y}(t) - \tilde{I}_{x,y}(t)\right] \end{aligned} \tag{3.8}$$

Background and foreground pixels are determined using simple thresholding: However, pixels whose differences of intensity to the system state are smaller than a fixed threshold $(d' < th_{bg})$ do not necessarily indicate background. For example, shadow pixels may cause a very small difference of intensity to the system state that might be smaller than the threshold. However, these pixels must not be classified as background and belongs to foreground. To identify such pixels, Ridder *et al.* propose a pre-estimation $\hat{I'}_{x,y}(t)$ of the next system state under the assumption that pixels with small differences in their intensities to the background belong to background. However, if the pre-estimated value d'' is greater than th_{bg} this pixel nevertheless belongs to foreground. A binary map $m_{x,y}(t)$ represents the segmentation of pixels (1 for foreground and 0 for background), and the Kalman gain $k1,2_{x,y}(t) = \alpha$ or $k1,2_{x,y}(t) = \beta$ is chosen depending on the binary map $m_{x,y}(t)$ (see Eq. 3.7).

$$k1,2_{x,y}(t) = \begin{cases} \alpha, & \text{if} \quad m_{x,y}(t) = 1 \\ \beta, & \text{if} \quad m_{x,y}(t) = 0 \end{cases} \tag{3.9}$$

3.3.3 Shadow detection

Body height prediction is based on precise extraction of head and foot points of approaching drivers. Therefore, it is highly recommended to carefully determine the regions of persons in consecutive images in order to estimate their body heights. Shadow pixels, however, may falsify the lower body regions and could lead to inaccuracies in body heights when head and foot points are wrongly estimated. Hence, shadow pixels must be detected and suppressed. The normalized cross correlation (NCC, [82]) is used as an initial step to detect shadow pixels, and the result is then refined by the zero mean normalized cross correlation (ZNCC) to handle foreground pixels with small differences to the background.

Let $\tilde{I}_{x,y}(t)$ be the estimated background image, and let $I_{x,y}(t)$ be an image captured from a scene. For each foreground pixel, a template $T_{xy}(n,m)$ is generated in such a way that $T_{xy}(n,m) = I_{x+n,y+m}(t)$ for $-N \leq (n,m) < N$. $\bar{t}$ denotes the arithmetical mean of template $T_{xy}(n,m)$. Furthermore, let $B_{xy}(n,m)$ be a corresponding template of the background in such a way that $B_{xy}(n,m) = \hat{I}_{x+n,y+m}(t)$. Also, $\bar{b}$ is the arithmetical mean of template $B_{xy}(n,m)$. Then, the similarity between the image template $T_{xy}(n,m)$ and the background template $B_{xy}(n,m)$ at pixel (x,y) is computed with ZNCC as well as NCC ($\bar{t} = 0, \bar{b} = 0$) following Eq. 3.10:

$$ZNCC_{x,y} = \frac{EZR_{x,y}}{EZB_{x,y} \cdot EZT_{x,y}} \tag{3.10}$$

with

$$\begin{aligned} EZR_{x,y} &= \sum_{n=-N}^{N} \sum_{m=-N}^{N} |(B_{xy}(n,m) - \bar{b})||(T_{xy}(n,m) - \bar{t})| \\ EZB_{x,y} &= \sqrt{\sum_{n=-N}^{N} \sum_{m=-N}^{N} (B_{xy}(n,m) - \bar{b})^2} \text{ and} \\ EZT_{x,y} &= \sqrt{\sum_{n=-N}^{N} \sum_{m=-N}^{N} (T_{xy}(n,m) - \bar{t})^2} \end{aligned} \tag{3.11}$$

The term $EZT_{x,y}$ considers the energy of the image template and $EZB_{x,y}$ the energy of the background template. A pixel may potentially be classified as shadow if the NCC value between both templates is close to unity and if the energy of the image template $ET_{x,y}$ is smaller than the energy of the background template $EB_{x,y}$ (see Eq. 3.12). The energies $EB_{x,y}$ and $ET_{x,y}$ can be determined by calculating $EZB_{x,y}(\bar{b} = 0)$ and $EZT_{x,y}(\bar{t} = 0)$.

$$NCC_{x,y} \geq th_{NCC} \text{ and } EB_{x,y} > ET_{x,y} \tag{3.12}$$

3.3.4 Shadow refinement

Depending on the chosen threshold th_{NCC} with $(th_{NCC} < 1.0)$, many foreground pixels with small differences to background pixels may be misclassified as shadow pixels. To overcome

this limitation, the classification of shadow- and nonshadow-pixels has been refined using the ZNCC. The advantage of ZNCC is light invariance, so that only differences in texture cause significant changes in its value. Hence, the refinement stage verifies if there are significant changes caused by changes in textures instead of illumination. Although the ZNCC is light invariant, image noise (texture changes) influence the ZNCC and cause an offset. This offset θ can be determined when initializing the background model and must be considered by the threshold th_{ZNCC}. Similarly to the NCC, a pixel belongs to shadow if the ZNCC is close to the initial value and the energy $ET_{x,y}$ of the image template is smaller than the energy of the background template $EB_{x,y}$.

In contrast to the NCC, $EZT_{x,y}$ and $EZB_{x,y}$ represent the energies of the textures from the background template and from the image template. Thus, the energy of textures from valid foreground pixels might be lower than the energy of texture from background. This is the case for large homogeneous objects like trucks that cover a very detailed background such as brushwood. A pixel may belong to shadow if the energy $EZT_{x,y}$ of the image template is approximately the same as the energy $EZB_{x,y}$ of the background following Eq. 3.13.

$$\begin{aligned} |ZNCC_{x,y} - (1.0 - \theta)| &\leq th_{ZNCC}\ and \\ |EZB_{x,y} - EZT_{x,y}| &\leq th_{comp}\ and \\ ET_{x,y} &< EB_{x,y} \end{aligned} \tag{3.13}$$

3.3.5 Active light adaptation

Background models that are based on Kalman filtering can follow slow illumination changes in the background. However, when foreground objects cover the background, illumination changes in the background cannot be detected. Furthermore, the background model cannot consider sudden illumination changes as it cannot distinguish between sudden illumination changes or fast moving foreground objects. So there is a need to modify the background model to consider sudden illumination changes in input images. Therefore, every new image is subdivided into m sub-images fitting the whole image, and the mean gray value for each sub-image is calculated (see Eq. 3.14):

$$\mu(m,t) = \frac{1}{J \cdot I} \sum_{j=-J/2}^{J/2} \sum_{i=-I/2}^{I/2} I(p_{x(m)} + j, p_{y(m)} + i, t) \tag{3.14}$$

$p_{x(m)}$ and $p_{y(m)}$ are the center of each sub-image and J, I its image size. The global illumination change $\Delta(t)$ can be detected by calculating the median of all local illumination changes $\delta(m,t)$.

$$\Delta(t) = \underset{m}{median}\delta(m,t) \tag{3.15}$$

with

$$\delta(m,t) = \mu(m,t) - \mu(m,t-1) \tag{3.16}$$

(a) (b)

Figure 3.5: Indoor scenario with walking people captured by an omnidirectional camera (a). Extracted humans in panoramic images (b).

Since small illumination changes are automatically adapted by the background model, simple thresholding is introduced in order to avoid modifying the background model too frequently.

$$\Delta(t) = \begin{cases} 0, & \text{if } \Delta(t) < th_{\Delta} \\ \Delta(t), & \text{if } \Delta(t) \geq th_{\Delta} \end{cases} \tag{3.17}$$

Finally, Eq. 3.5 that predicts the system state for background estimation is modified to consider illumination changes as follows:

$$\begin{bmatrix} \tilde{I}_{x,y}(t) \\ \tilde{\hat{I}}_{x,y}(t) \end{bmatrix} = S \cdot \begin{bmatrix} \hat{I}_{x,y}(t-1) \\ \hat{\hat{I}}_{x,y}(t-1) \end{bmatrix} + \begin{bmatrix} \Delta(t) \\ 0 \end{bmatrix} \tag{3.18}$$

Using this approach, the background model can consider slow as well as sudden illumination changes. Figure 3.5(a) shows an image containing a group of people walking in a complex indoor environment. The difficulties of this scenario are weak and strong shadows and small differences between foreground and background. Figure 3.5(b) illustrates the extraction result on which driver identification and height estimation is based.

3.3.6 Parallelization

For all the proposed techniques, one can see that different pixels may be processed in parallel. In other words, image rectification, the background estimator, the shadow detector as well as the illumination compensation can all be run in parallel on a multi-core CPU. As mentioned before, the algorithm has to work in real-time and hence such parallelization is highly desirable. A parallelization scheme is useful to speed up the system for time-critical tasks such as detection of traffic participants or detection of approaching driver. Therefore, original images returned by the camera subsystem are divided into n sub-images and the same number of threads is generated to be run on a multi-core platform. After processing each image the results of all threads must be merged and interpolated (e.g., when an object being detected is split across two or more sub-images) for further object detection and foot and head point detection algorithms. Figure 3.6 illustrates the realized parallelization technique.

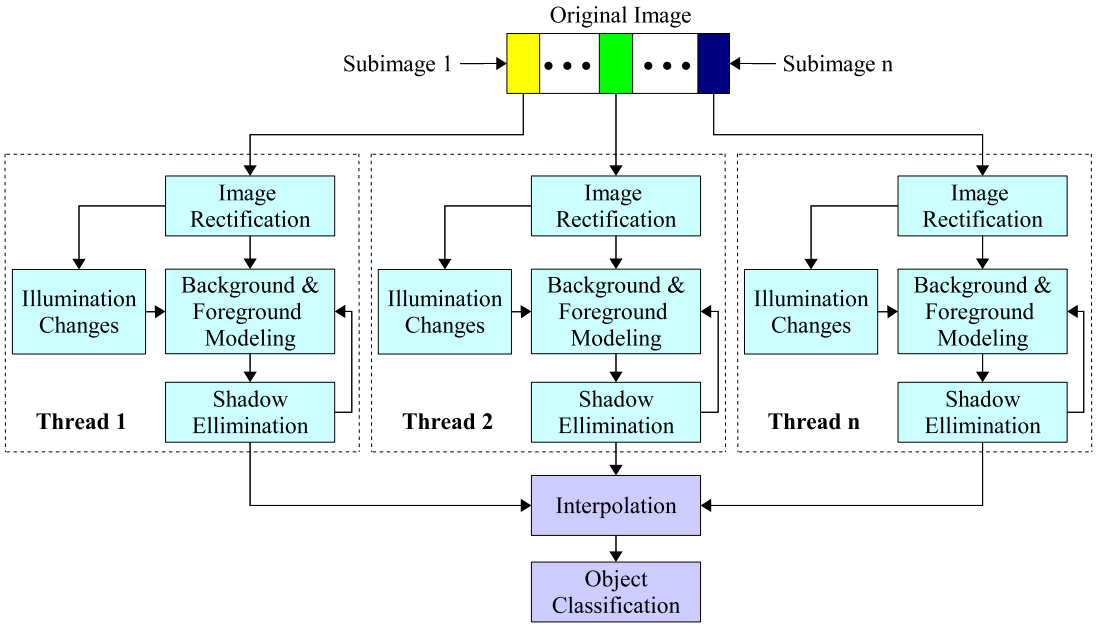


Figure 3.6: Parallelization concept of the object detection algorithm. Each input frame is subdivided into n-frames and the same number of concurrent threads is generated to extract road users. The results obtained from each thread are merged by interpolation.

3.3.7 Driver determination and foot/head point extraction

Precise human extraction and torso determination are prerequisites for obtaining highly accurate body heights of approaching drivers. Based on the torso, head and foot points are extracted and serve as input for the body height estimation algorithm. In previous sections, the proposed foreground detection algorithm has been presented that separates foreground objects such as approaching drivers or cars from background [5]. A classification factor f_c is introduced to distinguish between humans and other foreground objects. Following Eq. 3.19, an extracted region reg belongs to humans when the classification factor f_c is larger than a lower bound th_{low_class} and smaller than an upper bound th_{up_class}.

$$reg = \begin{cases} 1 & th_{low\_class} < f_c \leq th_{up\_class} \\ 0 & otherwise \end{cases} \quad \text{with } f_c = \frac{w_{reg}}{h_{reg}} \tag{3.19}$$

The classification factor f_c depends on both width w_{reg} and height h_{reg} of detected regions. Figure 3.7 illustrates an identified driver in a group of detected people walking close to the vehicle. Since drivers walk straight towards the car, they can easily be identified based on their distances to the vehicle and based on their trajectory in image coordinates. The distance of humans to the vehicle is closely related to the position of their regions in images, particularly to the location of the bottom edge of extracted regions. In other words, bottom edges of human regions close to the lower image parts mean a short distance to the vehicle and vice versa. Due to this, the region at the left side of Figure 3.7 with the shortest distance to the car is identified

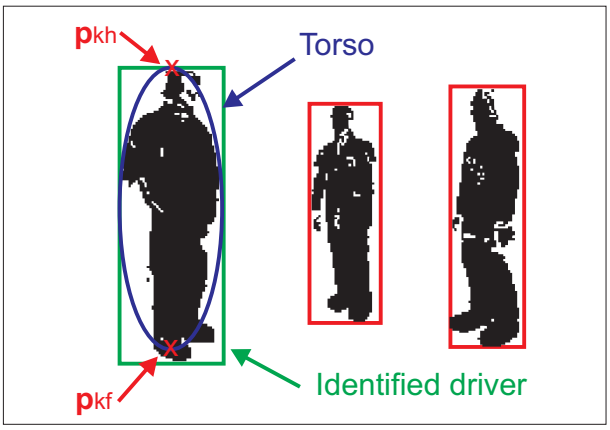


Figure 3.7: Identified driver in a group of detected humans. Head points $\mathbf{p}_{kh}$ and foot points $\mathbf{p}_{kf}$ are determined based on the torso of drivers.

as the region containing the potential car driver.

In a next step, the torso of potential drivers is estimated by means of a blob-detection algorithm [99, 100]. Torso estimation allows determining head and foot points of approaching drivers and overcomes the foot point determination problem of walking people. The problem of using the lowest point of extracted human regions in images is that physical location of this point strongly depends on the walking direction of humans. In other words, the lowest point of humans in images relates to the forward section of foots for approaching and to the heels for leaving humans. In addition, there is a large difference between these locations that is of up to 30 cm depending on the foot size. This may lead to errors in foot point determination and hence to errors in height estimation. The torso determination overcomes this problem by extracting foot ankles when estimating foot points. Based on the torso, head points and foot points in 2D-image coordinates are determined. The image coordinates of head and foot points are transformed into corresponding 3D-coordinates $\mathbf{p}_{kh}$, $\mathbf{p}_{kf}$ of the camera coordinate system K.

3.4 Driver body height estimation

In this section, a novel method is proposed to estimate the absolute body height of approaching car drivers to improve ingress in narrow parking lots. Section 3.4.1 introduces the coordinate systems and its relations to each other that are required for body height estimation. Section 3.4.2 presents a mathematical model that represents standard parking situations i.e., a car that is tilted with respect to the street surface due to its parking position on a curbstone. Section 3.4.3 introduces a model for a parking scenario where vehicles are parked in inclined positions. A general, mathematical framework is presented in Section 3.4.4 that combines both the curbstone and the inclined parking scenario into a general model. Each of these scenarios requires a particular mathematical solution to precisely determine the absolute body height from foot and head points of approaching drivers. These points are extracted in panoramic images obtained from an omnidirectional camera.

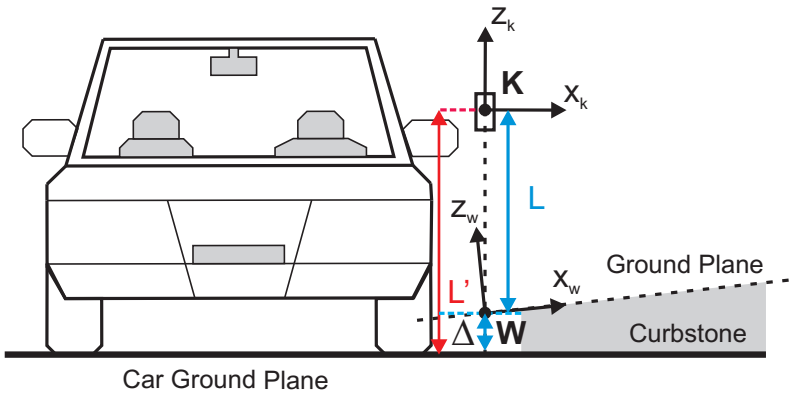


Figure 3.8: Orientation of both the camera and world coordinate system.

3.4.1 Definitions

The omnidirectional camera is integrated with the side-view mirror of the car and provides both a large vertical and horizontal field of view. The coordinate system $\mathbf{K}$ of the omnidirectional camera is located at the projection center of the mirror and has a fixed, well-known distance L' to the *car ground plane* (see Figure 3.8). The car ground plane is defined as the plane spanned by the four wheels of a car.

Distance L' is the key prerequisite for absolute height estimation as it overcomes the scale-factor problem of absolute body height estimation using a single camera only. It may be determined during extrinsic camera calibration. The z-axis of the camera coordinate system is assumed to be perpendicular to the car ground plane. Misalignments of its orientation can be detected and compensated by extrinsic camera calibration. A world coordinate system $\mathbf{W}$ is introduced at the intersection point between the z-axis of the camera coordinate system $\mathbf{K}$ and a *ground plane*. Moreover, the z-axis of the world coordinate system $\mathbf{W}$ is assumed to be perpendicular to the surface of the ground plane.

The ground plane is supposed to be the plane on which drivers walk straight toward the vehicle. This ground plane may be coincident with the car ground plane for inclined parking situations or may be tilted with respect to the car ground plane for curbstone parking scenarios. The reason why such a ground plane and the world coordinate system are so desirable is that body height estimation can drastically be simplified due to certain physical constraints.

A distance vector $\mathbf{d}_K = (0 \;\; 0 \;\; -L)^T$ is introduced that starts at the origin of the camera coordinate system $\mathbf{K}$ and ends at the origin of the world coordinate system $\mathbf{W}$. In other words, L is assumed to be the length of the distance vector $\mathbf{d}_K$ with respect to the z-axis of the camera coordinate system $\mathbf{K}$ (see Figure 3.8). The length L of vector $\mathbf{d}_K$ is called the *ground distance*. The ground distance depends on L' and on the tilt of the ground plane with respect to the car ground plane and is a prerequisite for absolute height estimation using only one camera. L can be determined by estimating the body heights of approaching drivers (see Section 3.5.3), and can also be determined by motion stereo-based algorithms (see Section 4). Stereo-based algorithms are especially required when a car is parked close to a curbstone meaning that there is an offset Δ between the car ground plane and the ground plane. This offset cannot be determined with image correspondences only.

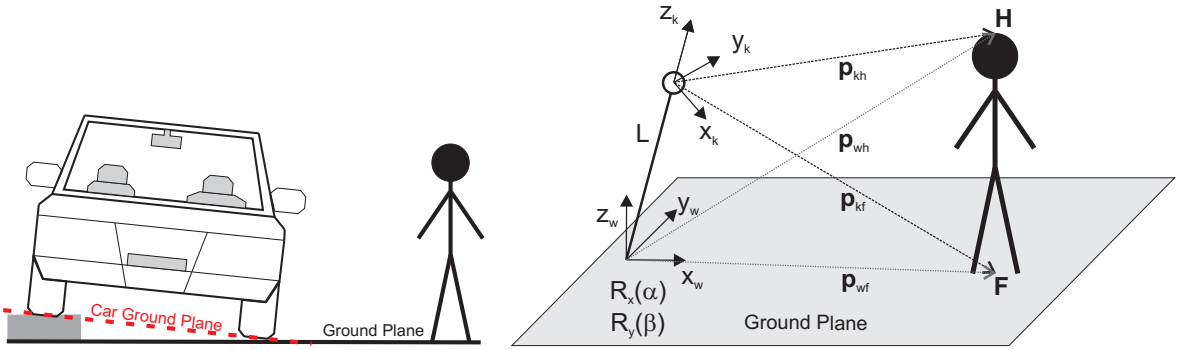


Figure 3.9: Model of a curbstone parking scenario, where a driver walks straight ahead towards a tilted vehicle.

3.4.2 Curbstone parking scenario – tilted vehicle

The first scenario describes parking situations where drivers walk straight ahead towards a vehicle that is parked on a curbstone. Due to the parking scenario, the car ground plane is tilted with respect to a horizontally oriented ground plane. Consequently, the camera system $\mathbf{K}$ is tilted along two dimensions with respect to the world coordinate system $\mathbf{W}$ located on the flat road surface. The relation between the camera coordinate system and the world coordinate system can be expressed by two rotations along the x-axis and y-axis of the world coordinate system $\mathbf{W}$. Figure 3.9 illustrates a scheme and a geometrical representation of this parking scenario.

Based on the silhouette of approaching drivers, sets of foot-points $\mathbf{p}_{kf}$ and head-points $\mathbf{p}_{kH}$ in camera coordinates are extracted. The lengths $\mathbf{p}_{kf}$ and $\mathbf{p}_{kH}$ are only known up to an unknown scale factor as a single camera provides the direction of vectors only. Hence, vectors to head- and foot-points can be described as $\mathbf{p}_{kf} = r \cdot \mathbf{p}_{N\_kf}$ and $\mathbf{p}_{kh} = t \cdot \mathbf{p}_{N\_kh}$ with $||\mathbf{p}_{N\_kf}|| = ||\mathbf{p}_{N\_kh}|| = 1$ and $r, t \in \mathbb{R}$. The vectors of head and foot points obtained in camera coordinates can then be transformed into world coordinates following Eq. 3.20

$$\mathbf{p}_{wf} = \mathbf{R}_w^k \cdot (\mathbf{p}_{kf} - \mathbf{p}_{kL}), \quad \mathbf{p}_{wh} = \mathbf{R}_w^k \cdot (\mathbf{p}_{kh} - \mathbf{p}_{kL}) \quad \text{with} \quad \mathbf{p}_{kH} = \begin{pmatrix} 0 \\ 0 \\ -L \end{pmatrix} \tag{3.20}$$

The relation between the camera and the world coordinate systems (tilting) is represented by the rotation matrix $\mathbf{R}_w^k$ (see Eq. 3.21).

$$\mathbf{R}_w^k = \mathbf{R}_y^T(\beta) \cdot \mathbf{R}_x^T(\alpha) = \begin{pmatrix} c\beta & s\alpha s\beta & -c\alpha s\beta \\ 0 & c\alpha & s\alpha \\ s\beta & -s\alpha c\beta & c\alpha c\beta \end{pmatrix} \tag{3.21}$$

For the rest of this section, the following notation for the normalized head- and foot-points as seen by the camera are used: $\mathbf{p}_{N\_kf} = (x_n^f \; y_n^f \; z_n^f)^T$ (foot points) and $\mathbf{p}_{N\_kf} = (x_n^h \; y_n^h \; z_n^h)^T$ (head points), where superscripts f and h indicate the coordinates of foot or head points and n the corresponding set number. Analogically to the normalized camera coordinates, the world

coordinates are represented by the following notation: $\mathbf{p}_{wf} = (X_n^f \; Y_n^f \; Z_n^f)^T$ and $\mathbf{p}_{wh} = (X_n^h \; Y_n^h \; Z_n^h)^T \; \forall n \in \mathbb{R}$. Following Eq. 3.22 and Eq. 3.23, Eq. 3.20 becomes:

$$\mathbf{p}_{wf} = \begin{pmatrix} X_n^f \\ Y_n^f \\ Z_n^f \end{pmatrix} = \begin{pmatrix} r \cdot (c\beta x_n^f + s\alpha s\beta y_n^f - c\alpha s\beta z_n^f) - c\alpha s\beta L \\ r \cdot (c\alpha y_n^f + s\alpha z_n^f) + s\alpha L \\ r \cdot (s\beta x_n^f - s\alpha c\beta y_n^f + c\alpha c\beta z_n^f) + c\alpha c\beta L \end{pmatrix} \tag{3.22}$$

$$\mathbf{p}_{wh} = \begin{pmatrix} X_n^h \\ Y_n^h \\ Z_n^h \end{pmatrix} = \begin{pmatrix} t \cdot (c\beta x_n^h + s\alpha s\beta y_n^h - c\alpha s\beta z_n^h) - c\alpha s\beta L \\ t \cdot (c\alpha y_n^h + s\alpha z_n^h) + s\alpha L \\ t \cdot (s\beta x_n^h - s\alpha c\beta y_n^h + c\alpha c\beta z_n^h) + c\alpha c\beta L \end{pmatrix} \tag{3.23}$$

Since drivers walk on a horizontally oriented ground plane, following assumptions can be made to derive an expression for the absolute driver height. Foot points are assumed to be the lowest points of approaching drivers, so that the z-component of $\mathbf{p}_{wf}$ can be set to zero. Additionally, the z-component of $\mathbf{p}_{wh}$ represents the height h of approaching drivers (see Eq. 3.24).

$$Z_n^f = 0, \quad Z_n^h = h \tag{3.24}$$

Another assumption is a nearly upright posture of drivers during walking: Hence, the x and y components of the vectors to foot and head points must be identical (see Eq. 3.25).

$$X_n^f = X_n^h, \quad Y_n^f = Y_n^h \tag{3.25}$$

Following Eq. 3.26, the unknown scale factors r and t can be computed as follows:

$$t = r \cdot \frac{c\alpha y_n^f + s\alpha z_n^f}{c\alpha y_n^h + s\alpha z_n^h} \quad \text{with} \quad r = \frac{-c\alpha c\beta L}{s\beta x_n^f - s\alpha c\beta y_n^f + c\alpha c\beta z_n^f} \tag{3.26}$$

and therefore the height h of approaching drivers following Eq. 3.27:

$$h = c\alpha c\beta L \cdot \left(1 - \frac{c\alpha y_n^f + s\alpha z_n^f}{c\alpha y_n^h + s\alpha z_n^h} \cdot \frac{s\beta x_n^h - s\alpha c\beta y_n^h + c\alpha c\beta z_n^h}{s\beta x_n^f - s\alpha c\beta y_n^f + c\alpha c\beta z_n^f}\right) \tag{3.27}$$

Determination of the camera tilt α, β from foot and head points

The height of approaching drivers can easily be determined when the camera tilt α and β is available. But this is not the case for real, unknown parking scenarios. However, the camera tilt α and β can be calculated when sets of vectors to foot and head points are available. Using Eq. 3.25 and replacing r and t with the expressions in Eq. 3.26, Eq. 3.28 describes a solution for β that depends only on the coordinates of foot points $\mathbf{p}_{N\_kf} = (x_n^f \; y_n^f \; z_n^f)^T$, on the coordinates of head points $\mathbf{p}_{N\_kf} = (x_n^h \; y_n^h \; z_n^h)^T$ and on the camera tilt α.

$$\frac{c\beta}{s\beta} = \frac{-(y_n^f z_n^h - y_n^h z_n^f)}{c\alpha \cdot (x_n^f y_n^h - x_n^h y_n^f) + s\alpha \cdot (x_n^f z_n^h - x_n^h z_n^f)} \tag{3.28}$$

Using at least two head and foot points, α can be determined by solving Eq. 3.29:

$$tan(\beta) = -\left(\frac{c\alpha B_n + s\alpha C_n}{A_n}\right) = \cdots = -\left(\frac{c\alpha B_n + s\alpha C_n}{A_n}\right) \text{ with } A_n \neq 0 \tag{3.29}$$

with $A_n = (y_n^f z_n^h - y_n^h z_n^f)$, $B_n = (x_n^f y_n^h - x_n^h y_n^f)$, $C_n = (x_n^f z_n^h - x_n^h z_n^f)$ and $n \in \mathbb{R}$. Eq. 3.30 illustrates a potential solution for camera tilt α based on two head- and foot-points.

$$tan(\alpha) = \frac{A_n B_{n+1} - A_{n+1} B_n}{A_{n+1} C_n - A_n C_{n+1}} \tag{3.30}$$

3.4.3 Inclined parking scenario – tilted road surface

The first scenario describes parking scenarios where drivers walk straight ahead towards a vehicle parked in an inclined position. For an inclined parking scenario, the car ground plane is collinear with respect to the ground plane. For this reason, the orientations of the camera system $\mathbf{K}$ and of the world coordinate system $\mathbf{W}$ are coincident. Similar to the first scenario, the assumption is made that drivers walk in an upright posture. But contrary to the first scenario, the upright posture of walking humans is not perpendicular to the ground plane. This tilt is called *driver tilt* and can also be expressed by a rotation along two dimensions. The tilt is described by two tilt angles γ and δ with respect to the orientation of the world coordinate system.

Figure 3.10 illustrates a scheme of the inclined parking scenario. Similarly to the first scenario, the height of approaching drivers can be derived from extracted sets of head points $\mathbf{p}_{N\_kf}$ and foot points $\mathbf{p}_{N\_kh}$ in normalized camera coordinates. Their corresponding world coordinates can be computed following Eq. 3.31:

$$\mathbf{p}_{wf} = r \cdot \mathbf{p}_{N\_kf} + \mathbf{p}_{wL}, \quad \mathbf{p}_{wh} = t \cdot \mathbf{p}_{N\_kh} + \mathbf{p}_{wL} \quad \text{with} \quad \mathbf{p}_{wL} = \begin{pmatrix} 0 \\ 0 \\ L \end{pmatrix} \tag{3.31}$$

where $\mathbf{p}_{wL}$ represents the distance between the camera and world coordinate system.

The following notation for the normalized head and foot points as seen by the camera are used. Foot points are represented by $\mathbf{p}_{N\_kf} = (x_n^f \; y_n^f \; z_n^f)^T$, whereas head points are represented by $\mathbf{p}_{N\_kf} = (x_n^h \; y_n^h \; z_n^h)^T$. Analogically to normalized camera coordinates, the world coordinates of foot and head points are represented by the following notation: $\mathbf{p}_{wf} = (X_n^f \; Y_n^f \; Z_n^f)^T$ and $\mathbf{p}_{wh} = (X_n^h \; Y_n^h \; Z_n^h)^T$. Eq. 3.32 illustrates the relation between foot and head points in world coordinates.

$$\mathbf{p}_{wh} = \mathbf{p}_{wf} + \mathbf{R}_{xy} \cdot \begin{pmatrix} 0 \\ 0 \\ h \end{pmatrix} \tag{3.32}$$

with

$$\mathbf{p}_{wf} = \begin{pmatrix} X_n^f \\ Y_n^f \\ Z_n^f \end{pmatrix} = r \cdot \begin{pmatrix} x_n^f \\ y_n^f \\ z_n^f \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ L \end{pmatrix} \text{ and } \mathbf{p}_{wh} = \begin{pmatrix} X_n^h \\ Y_n^h \\ Z_n^h \end{pmatrix} = t \cdot \begin{pmatrix} x_n^h \\ y_n^h \\ z_n^h \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ L \end{pmatrix} \tag{3.33}$$

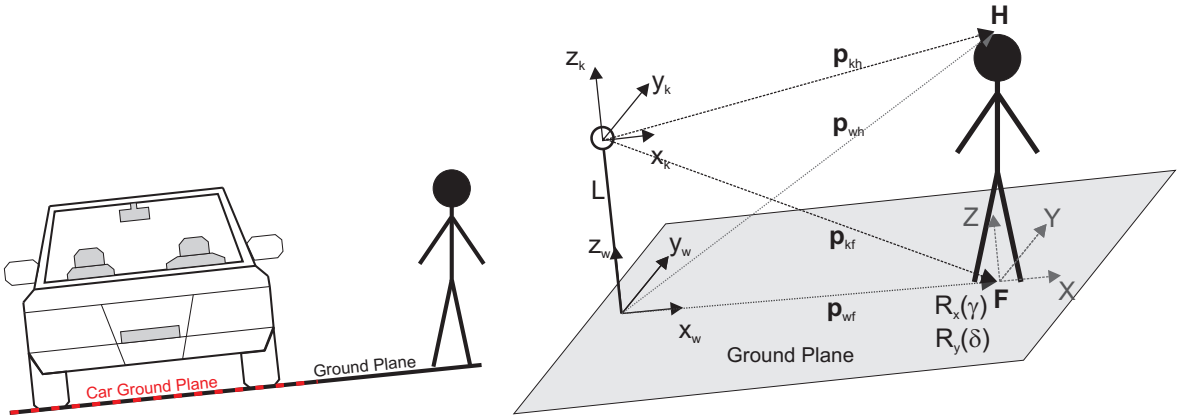


Figure 3.10: Parking situation where a driver walks towards an inclined parked vehicle.

The rotation matrix $\mathbf{R}_{xy}$ (see Eq. 3.34) represents the orientation (driver tilt) of the approaching driver with respect to the ground plane.

$$\mathbf{R}_{xy} = \mathbf{R}_x(\gamma) \cdot \mathbf{R}_y(\delta) = \begin{pmatrix} c\delta & 0 & s\delta \\ s\gamma s\delta & c\gamma & -s\gamma c\delta \\ -c\gamma s\delta & s\gamma & c\gamma c\delta \end{pmatrix} \tag{3.34}$$

Similarly to the curbstone parking scenario, the z-component Z_1 of foot-points can be set to zero. Due to this, scale factor r can be computed following Eq. 3.35.

$$r = -L/z_n^f \tag{3.35}$$

After replacing r and some reorganization of Eq. 3.32, the following equations can be used to determine the scale factor t and to estimate the absolute body height h.

$$-L \cdot x_n^f/z_n^f + h \cdot s\delta = t \cdot x_n^h \tag{3.36}$$
$$-L \cdot y_n^f/z_n^f - h \cdot c\delta s\gamma = t \cdot y_n^h \tag{3.37}$$
$$h \cdot c\delta c\gamma = t \cdot z_n^h + L \tag{3.38}$$

The second scale-factor t is obtained by reorganizing Eq. 3.38 and by replacing the scale-factors r and t in Eq. 3.37 (see Eq. 3.39).

$$t = \frac{h \cdot c\gamma c\delta - L}{z_n^h} \tag{3.39}$$

The height of approaching drivers for inclined parking scenarios is then determined following Eq. 3.40:

$$h = \frac{-L \cdot (y_n^f z_n^h - y_n^h z_n^f)}{z_n^f c\delta (s\gamma z_n^h + c\gamma y_n^h)} \tag{3.40}$$

Determination of driver tilt γ, δ from head and foot points

At this point, the absolute driver height h can be computed from the input data from the camera system when the driver tilt γ and δ is available – but this is not the case for parking scenarios

due to missing tilt information generated by inertial sensors. On the other hand, the input data provided by the camera system can also be used to estimate the tilt of approaching drivers. An arithmetic expression can be found by replacing t and h in Eq. 3.36 so that it depends only on the driver tilt γ (see Eq. 3.41)

$$\frac{c\delta}{s\delta} = \frac{-(y_n^f z_n^h - y_n^h z_n^f)}{c\gamma \cdot (x_n^f y_n^h - x_n^h y_n^f) + s\gamma \cdot (x_n^f z_n^h - x_n^h z_n^f)} \tag{3.41}$$

This equation may be used to find an expression for γ. Since γ should not vary when drivers approach the car, Eq. 3.41 must be valid for all head and foot points $\mathbf{p}_{N\_kh}$ and $\mathbf{p}_{N\_kh}$. Using at least two foot and head points, γ may be determined by solving Eq. 3.42

$$\tan\delta = -\frac{c\gamma B_1 + s\gamma C_1}{A_1} = \cdots = -\frac{c\gamma B_n + s\gamma C_n}{A_n} \quad \text{with} \quad A_n \neq 0 \tag{3.42}$$

with $A_n = y_n^f z_n^h - y_n^h z_n^f$, $B_n = x_n^f y_n^h - x_n^h y_n^f$ and $C_n = x_n^f z_n^h - x_n^h z_n^f$. Finally, Eq. 3.43 express a solution to compute the angle γ.

$$\tan\gamma = \frac{A_n B_{n+1} - A_{n+1} B_n}{A_{n+1} C_n - A_n C_{n+1}} \tag{3.43}$$

3.4.4 Generic height model combining curbstone and inclined parking scenario

In previous sections, algebraical models have been presented describing specific parking scenarios such as curbstone parking scenarios (see Section 3.4.2) or inclined parking scenarios (see Section 3.4.3). Their most important advantage is the analytical determination of the camera tilt (α, β) and the driver tilt (γ, δ). However, specific mathematical requirements are necessary to guarantee a valid solution for each set of input data. The input data are normalized vectors to the head points $\mathbf{p}_{N\_kh}$ and foot points $\mathbf{p}_{N\_kf}$ of approaching drivers obtained from the omnidirectional camera. Furthermore, real life parking scenarios usually combine both the inclined and the curbstone scenario; for instance when the vehicle is parked on a curbstone at an inclined road surface (see Figure 3.11). In general, one of the presented parking models dominates in each parking scenario, but neglecting the other one may lead to inaccuracies in height estimation (see Section 3.5.3) up to 20 cm. Therefore, a generic, arithmetic model considering both parking scenarios is highly desirable to obtain accurate body heights and is presented in this section.

Based on the extracted torso of identified drivers, sets of foot points $\mathbf{p}_{kf}$ and head points $\mathbf{p}_{kh}$ in camera coordinates are determined. The camera provides the direction of the vectors only and the lengths of $\mathbf{p}_{kf}$ and $\mathbf{p}_{kH}$ are only known up to an unknown scale factor. Hence, vectors to head and foot points can be represented as $\mathbf{p}_{kf} = r \cdot \mathbf{p}_{N\_kf}$ and $\mathbf{p}_{kh} = t \cdot \mathbf{p}_{N\_kh}$ with $||\mathbf{p}_{N\_kf}|| = ||\mathbf{p}_{N\_kh}|| = 1$ and $r, t \in \mathbb{R}$. The vectors to head and foot points are then transformed into world coordinates following Eq. 3.44

$$\mathbf{p}_{wf} = \mathbf{R}_{\mathbf{w}}^{\mathbf{k}} \cdot (\mathbf{p}_{kf} - \mathbf{p}_{kL}), \quad \mathbf{p}_{wh} = \mathbf{R}_{\mathbf{w}}^{\mathbf{k}} \cdot (\mathbf{p}_{kh} - \mathbf{p}_{kL}) \quad \text{with} \quad \mathbf{p}_{kH} = \begin{pmatrix} 0 \\ 0 \\ -L \end{pmatrix} \tag{3.44}$$

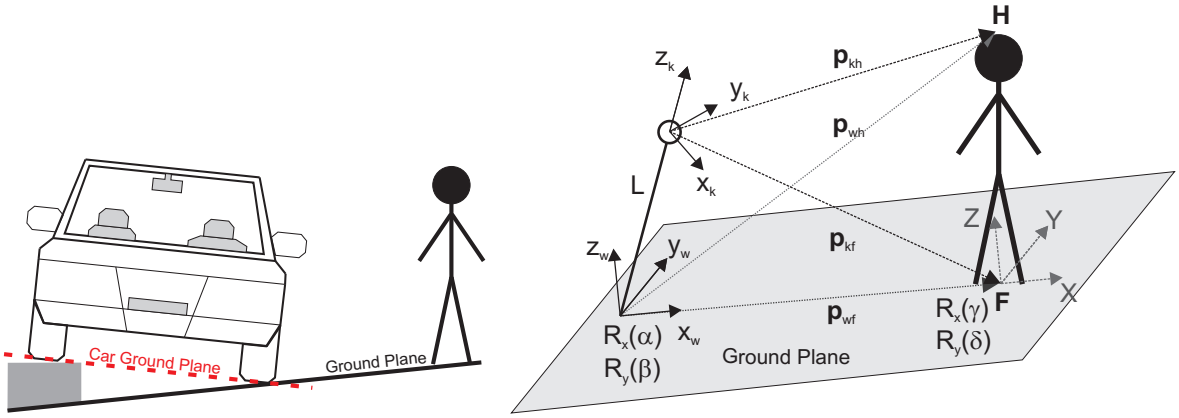


Figure 3.11: General model representing all potential parking situations. The model combines the tilted parking situation (α, β, see Figure 3.9) and the inclined parking situation (γ, δ, see Figure 3.10).

Hereby, the rotation matrix $\mathbf{R}_w^k$ represents the camera tilt of parked vehicles (see Eq. 3.45).

$$\mathbf{R}_w^k = \mathbf{R}_y^T(\beta) \cdot \mathbf{R}_x^T(\alpha) = \begin{pmatrix} c\beta & s\alpha s\beta & -c\alpha s\beta \\ 0 & c\alpha & s\alpha \\ s\beta & -s\alpha c\beta & c\alpha c\beta \end{pmatrix} \tag{3.45}$$

For the rest of this section, the following notations for the normalized head- and foot-points as seen by the camera are used: $\mathbf{p}_{N\_kf} = (x_n^f \; y_n^f \; z_n^f)^T$ and $\mathbf{p}_{N\_kf} = (x_n^h \; y_n^h \; z_n^h)^T$ where superscripts f and h indicate foot and head point coordinates and n the corresponding set number. Analogically to the normalized camera coordinates, the world coordinates are represented by the following notation: $\mathbf{p}_{wf} = (X_n^f \; Y_n^f \; Z_n^f)^T$ and $\mathbf{p}_{wh} = (X_n^h \; Y_n^h \; Z_n^h)^T \; \forall n \in \mathbb{N}$.

Analogically to the curbstone model and following Eq. 3.46 and Eq. 3.47, Eq. 3.44 becomes:

$$\mathbf{p}_{wf} = \begin{pmatrix} X_n^f \\ Y_n^f \\ Z_n^f \end{pmatrix} = \begin{pmatrix} r \cdot (c\beta x_n^f + s\alpha s\beta y_n^f - c\alpha s\beta z_n^f) - c\alpha s\beta L \\ r \cdot (c\alpha y_n^f + s\alpha z_n^f) + s\alpha L \\ r \cdot (s\beta x_n^f - s\alpha c\beta y_n^f + c\alpha c\beta z_n^f) + c\alpha c\beta L \end{pmatrix} \tag{3.46}$$

$$\mathbf{p}_{wh} = \begin{pmatrix} X_n^h \\ Y_n^h \\ Z_n^h \end{pmatrix} = \begin{pmatrix} t \cdot (c\beta x_n^h + s\alpha s\beta y_n^h - c\alpha s\beta z_n^h) - c\alpha s\beta L \\ t \cdot (c\alpha y_n^h + s\alpha z_n^h) + s\alpha L \\ t \cdot (s\beta x_n^h - s\alpha c\beta y_n^h + c\alpha c\beta z_n^h) + c\alpha c\beta L \end{pmatrix} \tag{3.47}$$

Eq. 3.48 describes a relation between foot and head points in world coordinates, whereas the z-component of the foot point vectors in world coordinates can be set to zero.

$$\mathbf{p}_{wh} = \mathbf{p}_{wf} + \mathbf{R}_{xy} \cdot \begin{pmatrix} 0 \\ 0 \\ h \end{pmatrix} = \begin{pmatrix} X_n^f \\ Y_n^f \\ 0 \end{pmatrix} + \mathbf{R}_{xy} \cdot \begin{pmatrix} 0 \\ 0 \\ h \end{pmatrix} = \begin{pmatrix} X_n^h \\ Y_n^h \\ Z_n^h \end{pmatrix}, \tag{3.48}$$

The rotation matrix $\mathbf{R}_{xy}$ represents the tilt of approaching drivers relative to the ground plane (see Eq. 3.49).

$$\mathbf{R}_{xy} = R_x(\gamma) \cdot R_y(\delta) = \begin{pmatrix} c\delta & 0 & s\delta \\ s\gamma s\delta & c\gamma & -s\gamma c\delta \\ -c\gamma s\delta & s\gamma & c\gamma c\delta \end{pmatrix} \tag{3.49}$$

By setting Z_n^f to zero, the scale-factor r can be computed following Eq. 3.46:

$$r = -\frac{L \cdot c\alpha c\beta}{s\beta x_n^f - s\alpha c\beta y_n^f + c\alpha c\beta z_n^f} \tag{3.50}$$

The unknown scale factor r in the X_n^f, Y_n^f components of Eq. 3.46 can be replaced by Eq. 3.50. The new expression for the X_n^f and Y_n^f components can be inserted into Eq. 3.48 to obtain three new equations (see Eq. 3.51, Eq. 3.52, Eq. 3.53). These equations can be used to determine for obtaining a new expression to the body height.

$$t \cdot (c\beta x_n^h + s\alpha s\beta y_n^h - c\alpha s\beta z_n^h) = -\frac{c\alpha c\beta L \cdot (c\beta x_n^f + s\alpha s\beta y_n^f - c\alpha s\beta z_n^f)}{s\beta x_n^f - s\alpha c\beta y_n^f + c\alpha c\beta z_n^f} + s\delta h \tag{3.51}$$

$$t \cdot (c\alpha y_n^h + s\alpha z_n^h) = -\frac{c\alpha c\beta L \cdot (c\alpha y_n^f + s\alpha z_n^f)}{s\beta x_n^f - s\alpha c\beta y_n^f + c\alpha c\beta z_n^f} - c\delta s\gamma h \tag{3.52}$$

$$t \cdot (s\beta x_n^h - s\alpha c\beta y_n^h + c\alpha c\beta z_n^h) + c\alpha c\beta L = c\delta c\gamma h \tag{3.53}$$

The second scale factor t can be determined by solving Eq. 3.53 as follows:

$$t = -\frac{-c\delta c\gamma h + c\alpha c\beta L}{s\beta x_n^h - s\alpha c\beta y_n^h + c\alpha c\beta z_n^h}, \tag{3.54}$$

The body height h is then determined by replacing the scale factor t in Eq. 3.51 with Eq. 3.54 and by solving Eq. 3.51. Following Eq. 3.55, the body height can be computed from a set of foot- and head-points obtained from the camera system if the camera tilt $\alpha,\ \beta$ and the driver tilt $\gamma,\ \delta$ are available.

$$\begin{aligned} h = & \frac{c\alpha c\beta L(c\alpha C_1 - s\alpha B_1)}{(s\beta x_n^f - s\alpha c\beta y_n^f + c\alpha c\beta z_n^f)} \cdot \ ... \\ & \cdot \frac{1}{[s\delta(s\beta x_n^h - s\alpha c\beta y_n^h + c\alpha c\beta z_n^h) - c\delta c\gamma(c\beta x_n^h + s\alpha s\beta y_n^h - c\alpha s\beta z_n^h)]} \end{aligned} \tag{3.55}$$

with

$$\begin{aligned} B_1 &= x_n^f y_n^h - x_n^h y_n^f \\ C_1 &= x_n^f z_n^h - x_n^h z_n^f \end{aligned}$$

3.4.5 Estimation of camera tilt $(\alpha,\ \beta)$ and driver tilt $(\gamma,\ \delta)$

In contrast to the curbstone and inclined parking scenario, it is difficult to analytically derive expressions to estimate the camera tilt α, β and the driver tilt γ, δ from a set of foot points $\mathbf{p}_{N\_kf}$ and head points $\mathbf{p}_{N\_kh}$. The reason for that is that camera and driver tilt mathematically depend on each other and that it is hardly possible to separate them into dedicated equations describing the camera tilt and the driver tilt only. For this reason, the strategy for overcoming this problem is to numerically minimize a model-based function $\mathbf{f}(\alpha, \beta, \gamma, \delta, \mathbf{p}_{kf}, \mathbf{p}_{kh})$ to obtain the best guess for $\alpha, \beta, \gamma, \delta$ from the input data. Ideally, the minimization function should

become zero for sets of head and foot points if the estimated camera and driver tilts best match with the real camera and driver tilts (see Eq. 3.56).

$$\min_{\alpha,\beta,\gamma,\delta} ||\mathbf{f}(\alpha,\beta,\gamma,\delta)|| \quad \Rightarrow \quad \mathbf{f}(\alpha,\beta,\gamma,\delta) \rightarrow 0 \tag{3.56}$$

In other words, the input data sets depend on the camera tilt and the driver tilt. Therefore, a minimization function $\mathbf{f}(\alpha,\beta,\gamma,\delta)$ can be found that has a global minimum when the numerically estimated values for α, β, γ and δ best match with the real tilt values.

One potential candidate for this minimization function $\mathbf{f}(\alpha,\beta,\gamma,\delta)$ is the *height difference function* computed for two or more input data sets. The height h of approaching drivers must be approximately constant for any set of input data. The differences between the estimated body heights computed for two or more input data sets become zero. Eq. 3.57 expresses this relation, whereas i represents the index of one input data set.

$$\mathbf{f}_{hd}(\alpha,\beta,\gamma,\delta) = ||h_i - h_{i+1}|| \quad \forall i \in \mathbb{N} \tag{3.57}$$

The required height h is computed based on the function expressed by Eq. 3.55. However, this minimization function is not recommended because of ambiguities of driver tilt γ. γ effects equation Eq. 3.55 within the term $cos(\gamma)$, so that it is not possible to differentiate between positive or negative solutions for γ. Furthermore, the trigonometrical terms related to the camera tilt α and β dominate and lead to good minimization results for the camera tilt, but to worse minimization results concerning the driver tilt.

A better minimization function $\mathbf{f}(\alpha,\beta,\gamma,\delta)$ has been derived by stacking the expressions for scale-factor t obtained in Eq. 3.54 for the height h obtained in Eq. 3.55 into Eq. 3.52. Eq. 3.58 expresses a new minimization function $\mathbf{f}_{hm} = \mathbf{f}_{hm}(\alpha,\beta,\gamma,\delta)$ to overcome ambiguities in the determination of γ. Furthermore, a more robust solution for α, β, γ and δ can be found during the minimization process due to a better balanced presence of trigonometrical terms related to camera and driver tilt. Another advantage of this expression is its capability to estimate a solution for camera tilt and driver tilt without any knowledge of the ground distance L between the camera and the ground plane. The input data sets $\mathbf{p}_{N\_kf}$ and $\mathbf{p}_{N\_kh}$ depend on the camera and driver tilt. Therefore, the minimization function becomes zero if the estimated camera tilt and driver tilt best match with the real camera and driver tilt.

$$\mathbf{f}_{hm} = \left| \frac{A_i^1}{D_i^1} + \frac{c\delta s\gamma(c\alpha C_i^1 - s\alpha B_i^1)}{D_i^1(s\delta D_i^2 - c\delta c\gamma E_i)} + \left(\frac{c\delta c\gamma(c\alpha C_i^1 - s\alpha B_i^1)}{D_i^1(s\delta D_i^2 - c\delta c\gamma E_i)} - 1 \right) \cdot \frac{A_i^2}{D_i^2} \right| \stackrel{!}{=} 0 \tag{3.58}$$

with:

$$\begin{aligned}
A_i^1 &= c\alpha y_i^f + s\alpha z y_i^f \\
A_i^2 &= c\alpha y_i^h + s\alpha z_i^h \\
B_i^1 &= x_i^f y_i^h - x_i^h y_i^f \\
C_i^1 &= x_i^f z_i^h - x_i^h z_i^f \\
D_i^1 &= s\beta x_i^f - s\alpha c\beta y_i^f + c\alpha c\beta z_i^f, D_i^1 \neq 0 \\
D_i^2 &= s\beta x_i^h - s\alpha c\beta y_i^h + c\alpha c\beta z_i^h, D_i^2 \neq 0 \\
E_i &= c\beta x_i^h + s\alpha s\beta y_i^h - c\alpha s\beta z_i^h
\end{aligned}$$

and $(s\delta D_i^2 - c\delta c\gamma E_i) \neq 0$. Subscript i represents the actual index of an input data set that consists of the foot point coordinates $(x_i^f\, y_i^f\, z_i^f)^T$ and head point coordinates $(x_i^h y_i^h z_i^h)^T$. Another advantage of Eq. 3.58 is that one set of input data would theoretically be sufficient to determine camera and driver tilt by minimizing Eq. 3.58. However, at least 32 input data sets are recommended to reduce the influence of noise.

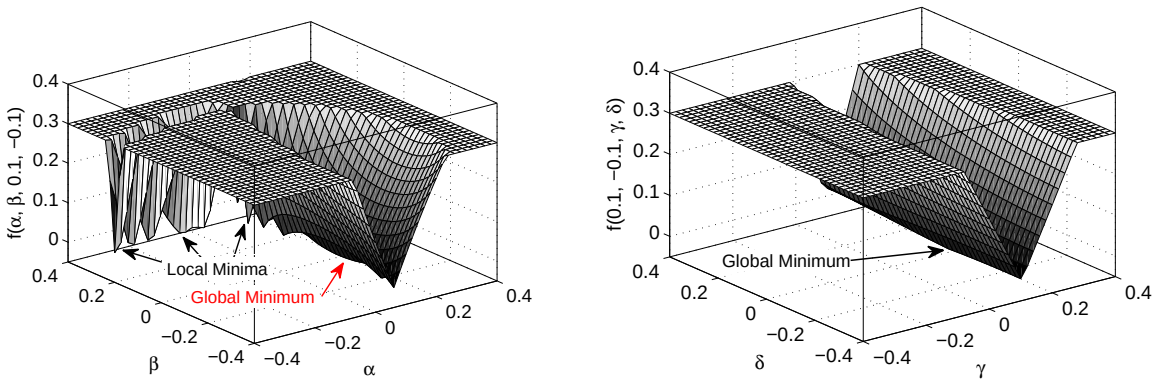
When dealing with linear equation systems, least-square algorithms may be used to obtain solutions for mathematical problems that are based on multiple input data sets. For that purpose, a general procedure is to split the minimization problem into the following form $\mathbf{Ax} \leq \mathbf{b}$ so that $\mathbf{x} = (\alpha\ \beta\ \gamma\ \delta)^T$. But this is not possible due to the highly nonlinear minimization term (see Eq. 3.58). As an alternative, solutions for nonlinear terms can be found by solving a *multi-objective minimization problem*. In this context, for this problem, the basic idea is to solve a set of objectives simultaneously by formulating this problem as a goal attainment problem introduced by Gembicki [101]). The goal attainment problem of Gembicki [101] is briefly presented below.

$$\underset{\mathbf{x},\theta}{\text{minimize}}\ \theta \text{ such that } \begin{cases} \mathbf{F}(\mathbf{x}) - \mathbf{weight} \cdot \theta \leq \mathbf{goal} \\ c_{eq}(\mathbf{x}) = 0 \\ lb \leq \mathbf{x} \leq ub \end{cases} \tag{3.59}$$

The variables $\alpha, \beta, \gamma, \delta$ are the entries of vector $\mathbf{x} = (\alpha, \beta, \gamma, \delta)^T$. The advantage of this minimization method is that objectives $f_{hm}(\mathbf{x_i})$ can be defined for each input data set $i \in \mathbb{N}$ and that an appropriate solution can be found minimizing all objectives simultaneously.

In other words, all objectives $\mathbf{F}(\mathbf{x}) = \{f_{hm}(\mathbf{x_1}), f_{hm}(\mathbf{x_2}), ..., f_{hm}(\mathbf{x_n})\}$ are simultaneously solved in least square sense. Furthermore, design goals $\mathbf{goal} = (goal_1, goal_2, ..., goal_n)$ are introduced that relate to the number of objectives $\mathbf{F}(\mathbf{x})$. $\mathbf{goal}$ can also be understood as a set of values that objectives $\mathbf{F}(\mathbf{x})$ should attain during minimization. Vector $\mathbf{weights}$ allows the objectives to be over- or underachieved. Gembicki [101] proposed the weighting coefficients $\mathbf{weight} = (w_1, w_2, ..., w_n)$ to control the relative degree of the objectives' over or under attainment. This is important since it is sometimes not known whether the objectives reach the goals (under attainment) due to the strong noise of the input data provided by the camera. Additionally, constraints can be introduced to limit the search area – such as to define a lower bound lb or an upper bound ub – and to define auxiliary conditions such as $c_{eq}(\mathbf{x})$ in order to obtain better convergence, with $c_{eq}(\mathbf{x}) = |h(\mathbf{x}_i) - h(\mathbf{x}_{i+1})|$ (see Eq. 3.57). *Sequential quadratic programming* (SQP) [102, 103, 104] method is used for minimization. The minimization algorithm proposed by Gembicki [101] yields highly precise results, but the limitation of this concept is the estimation of local solutions for camera tilt α, β and driver tilt γ and δ only.

Moreover, initial points $\mathbf{x}_0 = (\alpha_0\ \beta_0\ \gamma_0\ \delta_0)^T$ have to be determined close to the global minimum to guarantee convergence of the minimization scheme into the global minimum of Eq. 3.58 for any input data set. Figure 3.12 illustrates a simulated characteristic of the 4-dimensional minimization function $\mathbf{f}_{hm}(\mathbf{x})$ for fictive camera and driver tilt $\mathbf{x}_{tilt} = (0.1\ -0.1\ 0.1\ -0.1)^T$. In particular, Figure 3.12(a) illustrates the characteristic of $\mathbf{f}_{hm}$ assuming $\gamma = 0.1$ and $\delta = -0.1$. It can be seen that $\mathbf{f}_{hm}$ has many local minima and only one global minimum. Choosing wrong initial points leads the function to converge into one of the local minima.



(a) (b)

Figure 3.12: Local and global minima of the minimization function for tested tilts a:) $\gamma = 0.1, \delta = -0.1$ and b:) $\alpha = 0.1, \beta = -0.1$ (Unit [rad]).

Similarly to Figure 3.12(a), Figure 3.12(b) shows the characteristic of $\mathbf{f}_{hm}$ that depends on γ and δ under the assumption of constant values for $\alpha = 0.1$ and $\beta = -0.1$. Additionally, in Figure 3.12(b), it can be seen that there is only one global minimum at $(0.1, -0.1)$ for $\mathbf{f}_{hm}$. However, the gradient in direction of δ is very low and may lead to a slow convergence into the global minimum. For these reasons, it is highly recommended to choose initial points $\mathbf{x}_0$ close to the global minimum.

The algorithm solving the multi-objective goal attainment problem is very time consuming. Therefore, it is highly recommended to test only a few initial points. Experiments show that good minimization results can be achieved for about 12 initial points that have been chosen close to the global minimum. Therefore, this thesis proposes an algorithm to find appropriate initial points $\mathbf{x}_0$ that are located close to the global minimum on the one hand and that reduce the search area for minimization on the other hand. Figure 3.13 illustrates a block diagram of the proposed algorithm to find appropriate starting points. Intermediate iteration stages of this algorithm are illustrated in Figure 3.14.

First, the maximum search area for both the camera tilt α, β and the driver tilt γ, δ is defined and a fixed number m of initial points is uniformly spread over the whole search area (see Figure 3.14(a), Figure 3.14(d)). Thereafter, each initial point $\mathbf{x}_{0i}$ is tested for being a potential starting point by stacking it into the minimization function $\mathbf{f}_{hm}$. Results are computed for each initial point and the best n initial points are selected by choosing the ones that result within values close to zero for $\mathbf{f}_{hm}(\mathbf{x}_{0i})$. As a next stage, the algorithm checks whether the search area has been modified or not. This might be the case if all best initial points are located within a small area. A new search area for both camera and driver tilt is defined and the same number m of initial points is uniformly spread over the whole search area to potentially find better starting points (see Figure 3.14(b), Figure 3.14(e)). The algorithm repeats this iteration stage (*Inner Loop*, see Figure 3.13) until the maximum number of iterations has been exceeded and there are no significant changes by modifying the search area. Most of n best initial points selected last

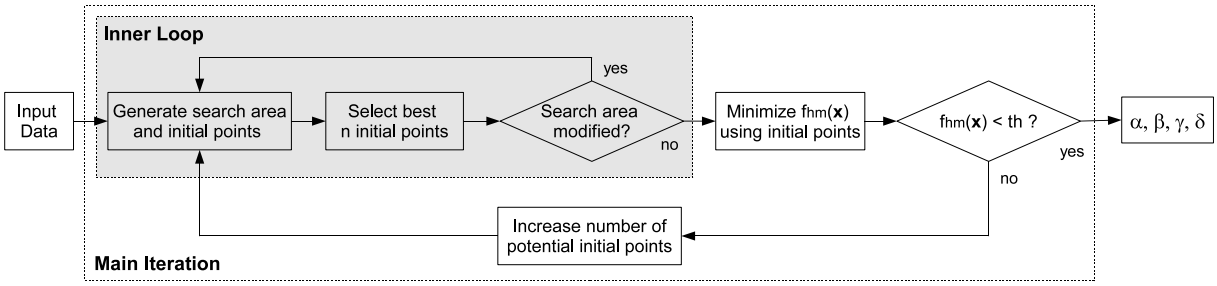


Figure 3.13: Block diagram of the algorithm to determine initial points.

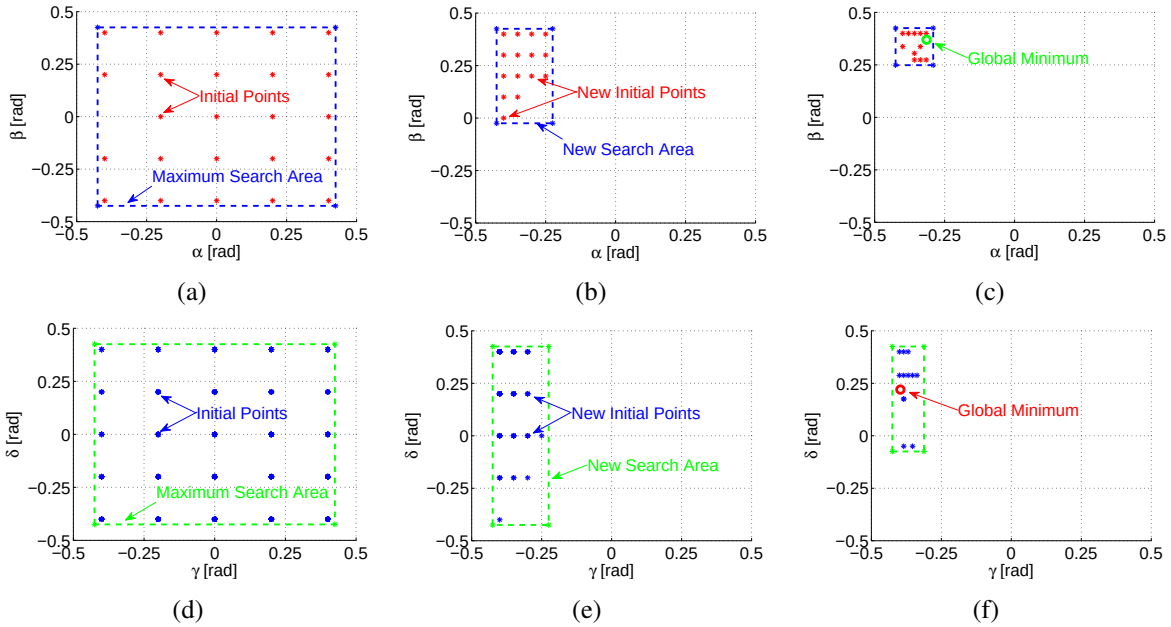


Figure 3.14: Starting point optimization to find appropriate initial points for estimating camera tilt α, β (a,b,c) and driver tilt γ, δ (g,d,e) (Unit [rad]).

are located close to the global minimum and serve as an input to solve the multi-objective goal attainment problem.

Figure 3.14(c) and Figure 3.14(f) illustrate the graphical results $\mathbf{x} = (\alpha, \beta, \gamma, \delta)^T$ for an exemplary minimum at $(-0.31\ 0.37\ -0.39\ 0.22)$. The algorithm is repeated with an increased number of initial points m to find better initial points to minimize $\mathbf{f}_{hm}$ if the global minimum has not been detected.

3.4.6 Fast estimation of camera tilt (α, β) and driver tilt (γ, δ)

Camera and driver tilt estimation by solving a multi-objective goal attainment minimization problem leads to very precise results but the time for execution is very long and may take up to 30 seconds. Studies illustrated a maximum time of 5 seconds that is available to estimate the height of approaching drivers. This is the time between activating the car door system by a remote key and the first contact with the outer door handle for normal ingress situations. Hence,

there is a need for an efficient implementation to provide fast estimation of the camera tilt and driver tilt. For this reason, a standard minimization problem can be solved for input data sets $i \in \mathbb{N}$ using a Levenberg-Marquardt minimization algorithm [51, 52].

$$\underset{\alpha,\beta,\gamma,\delta}{min} \sum_{i=1}^{n} \mathbf{f}_i^2(\alpha, \beta, \gamma, \delta) \tag{3.60}$$

Lourakis provided a fast C/C++ implementation of the Levenberg-Marquardt algorithm [105]. Camera and driver tilt estimation consists of two processing stages. The camera tilt is initially determined by minimizing the function $\mathbf{f}_{hd}(\alpha, \beta, \gamma, \delta) = ||h_i - h_{i+1}||$ with the height function h_i (see Eq. 3.55). In Eq. 3.55, the camera tilt dominates and is suitable for being estimated as an initial guess for further minimization stages. Thereafter, the algorithm estimates camera and driver tilt by iteratively minimizing function $\mathbf{f}_{hm}$ (see Eq. 3.58) using the initial guess for α and β. Results of the second stage are refined by an iteration process that stops when there are no significant changes in the current estimation of $(\alpha\ \beta\ \gamma\ \delta)$ compared to previous iteration stages.

3.4.7 Ground distance estimation

The camera and driver tilt can be estimated without any knowledge of the ground distance L. The ground distance is the length of the distance vector between the camera and the world coordinate system (see Section 3.4.1). This is an advantage, on one hand, but this is also a drawback, on the other hand, as L cannot be determined during the pose estimation process. Moreover, the ground distance L influences Eq. 3.56 as a scale factor: This scale factor is important to precisely estimate the body height of approaching drivers and overcomes the scale factor problem for height estimation using one camera only.

The ground distance L can be determined with the calibrated distance L' between the origin of the camera coordinate system and the car ground plane and the estimated camera tilt $(\alpha, \beta)^T$, and mainly depends on the actual parking scenario. A prerequisite for estimating L from L' and the camera tilt is that at least one of the car's wheels has contact to the *ground plane*. When the vehicle is correctly parked in the direction of travel, the assumption can always be made that at least one of the car's wheels – usually one on the driver side – has contact to the plane road surface. The road surface is identical with the *ground plane* if an approaching driver walks straight towards the car on the road (see Figure 3.11).

For such parking situations, L can be computed with the help of L' and the camera tilt (α, β). In another parking scenario, a vehicle is parked on the road surface close to the curbstone and co-drivers approach on a curbstone. In this case, none of the wheels has contact with the *ground plane* so that there is an unknown offset Δ between the wheels and the *ground plane*. This unknown offset can be considered in L but cannot be determined using a single omnidirectional camera only. Hence, two mathematical descriptions for two parking scenarios must be taken into account: One for scenarios where at least one wheel has contact with the ground plane and one for scenarios where drivers walk straight toward the car on an elevated ground plane (such as curbstone).

Wheel-based ground distance estimation

If at least one car wheel has contact with the ground plane, L can be computed using the camera tilt (α, β) and the calibrated distance L'. As mentioned in Section 3.4.1, L is the length of the distance vector between the origin $\mathbf{K}$ of the camera coordinate system and the origin of the world coordinate system $\mathbf{W}$. The origin $\mathbf{W}$ of the world coordinate system is defined as the intersection point of the z-axis of the camera coordinate system and the ground plane (see Figure 3.8 and Figure 3.15).

A new coordinates system $\mathbf{C}$ – called *car wheel coordinate system* is introduced whose origin is located at the boundary point of the wheel and the ground plane. The corresponding wheel that has contact with the ground plane can be determined with the estimated camera tilt. The orientation of the car wheel coordinate system $\mathbf{C}$ is assumed to be coincident with the world coordinate system $\mathbf{W}$. The position of the camera system $\mathbf{K}$ relative to one of the car wheels can precisely be determined during extrinsic camera calibration. Misalignments of the camera can also be detected by the calibration procedure and be considered in image rectification. The left side of Figure 3.15 illustrates the location of the omnidirectional camera relative to one of the car wheels. The location of the car wheel coordinate system $\mathbf{C}$ for a parked car (tilt $\beta \geq 0$, curbstone parking scenario) is illustrated on the right side of Figure 3.15. It depends on the camera tilt (α, β) and has to be placed into the car wheel that has contact to the ground plane.

Let $\mathbf{p}_{cK} = (0, 0, -L')^T$ be the distance vector beginning at the camera system $\mathbf{K}$ in direction of the world coordinate system $\mathbf{W}$. Furthermore, let $\mathbf{p}_{cC} = (k_x, k_y, L')^T$ be a vector in the direction of the camera coordinate system $\mathbf{K}$ beginning at the car wheel coordinate system $\mathbf{C}$. Both vectors are related to the car wheel coordinate system $\mathbf{C}$ and k_x and k_y are assumed to be known. Then, a straight line g can be computed with respect to the z-axis of the camera coordinate system $\mathbf{K}$ beginning at the center of the camera coordinate system (see Eq. 3.61).

$$g: \ \mathbf{r} = \mathbf{p}'_{cC} + \lambda \cdot \mathbf{p}'_{cH}, \ \lambda \in \mathbb{R} \tag{3.61}$$

with $\mathbf{p}'_{cC} = (cC_x, cC_y, cC_z)^T = \mathbf{R}_K^W \cdot \mathbf{p}_{cC}$, $\mathbf{p}'_{cK} = (cK_x, cK_y, cK_z)^T = \mathbf{R}_K^W \cdot \mathbf{p}_{cK}$ and

$$\mathbf{R}_K^W = \mathbf{R}_x(\alpha) \cdot \mathbf{R}_y(\beta) = \begin{pmatrix} c\beta & 0 & s\beta \\ s\alpha s\beta & c\alpha & -s\alpha c\beta \\ -c\alpha s\beta & s\alpha & c\alpha c\beta \end{pmatrix}. \tag{3.62}$$

Hereby, the straight line g is defined in car wheel coordinates. The origin $\mathbf{p}_{wO}$ of the world coordinate system $\mathbf{W}$ is defined as the intersection point of the straight line g with the ground plane beginning at the origin of the camera coordinate system with respect to its z-axis. Therefore, length L is the length of the straight line g and can be determined as follows.

$$\mathbf{p}_{wO} = \begin{pmatrix} O_x \\ O_y \\ 0 \end{pmatrix} = \begin{pmatrix} cC_x \\ cC_y \\ cC_z \end{pmatrix} + \lambda \cdot \begin{pmatrix} cK_x \\ cK_y \\ cK_z \end{pmatrix} \tag{3.63}$$

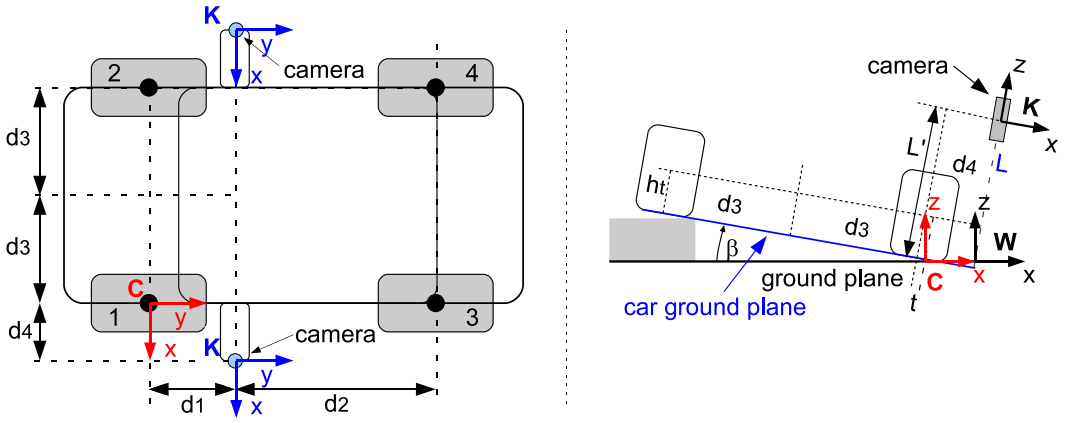


Figure 3.15: Location of the omnidirectional camera relative to one of the car wheels (left). Location of the coordinate systems for a parked car (tilt $\beta \geq 0$, curbstone parking scenario). The location of the car wheel coordinate system **C** depends on the camera tilt (α, β) has to be placed into the car wheel that has contact to the ground plane.

With $\lambda = -cC_z/cK_z$, the distance L can be computed following Eq. 3.64:

$$L = ||\mathbf{p}_{wO} - \mathbf{p}'_{cC}|| \tag{3.64}$$

Depending on the parking scenario and on the camera tilt (α, β), the components k_x and k_y of vector $\mathbf{p}_{cC}$ can be computed for cameras attached to the driver side of a car following Eq. 3.65 and for cameras attached to the co-driver side of a car following Eq. 3.66. Additionally, the orientation of the camera coordinate system at the co-driver side is assumed to be coincident with the camera coordinate system of the driver side (see Figure 3.15), whereas h_t represents the half diameter of the car wheel and $s(\alpha) = sin(\alpha)$.

$$\mathbf{p}_{cC\_driver} = \begin{cases} (d_4 - h_t s(\beta), d_1 + h_t s(\alpha), L')^T & \text{for } \alpha \geq 0\text{ , wheel 1} \\ (d_4 - h_t s(\beta), -d_2 + h_t s(\alpha), L')^T & \text{for } \alpha < 0\text{ , wheel 3} \end{cases} \tag{3.65}$$

$$\mathbf{p}_{cC\_co-driver} = \begin{cases} (-d_4 + h_t s(\beta), d_1 + h_t s(\alpha), L')^T & \text{for } \alpha \geq 0\text{ , wheel 2} \\ (-d_4 + h_t s(\beta), -d_2 + h_t s(\alpha), L')^T & \text{for } \alpha < 0\text{ , wheel 4} \end{cases} \tag{3.66}$$

Ground distance estimation using ambiance information

If there is an offset Δ between the car wheel coordinate system and the ground plane, L cannot be determined with the algorithm presented above. Then, 3D-ambiance information – in particular 3D-ambiance information of the ground plane – may be obtained to determine the distance L. For example, a motion stereo algorithm can generate distance information using a single camera algorithm.

Let $\mathbf{p}_i^k,\ i \in \mathbb{N}$ be vectors with known length to at least three points on the ground plane. These vectors are obtained from the omnidirectional camera and are noted in the camera coordinate

system K. These vectors describe a plane $\mathbf{E}$ on which each point can be determined with the scale factors r and t of the vectors following $\mathbf{E}: \ \mathbf{e^k} = \mathbf{p}_1^k + r \cdot (\mathbf{p}_1^k - \mathbf{p}_2^k) + t \cdot (\mathbf{p}_1^k - \mathbf{p}_3^k)$. The distance L is defined as the length of the vector $\mathbf{p}_D^k$ between the origin of the camera coordinate system and the intersection point in $\mathbf{E}$ with respect to the z-axis of the camera system. Then, $\mathbf{p}_D^k$ can be described as $\mathbf{p}_D^k = (0 \ 0 \ -L)^T$ and the length L can be determined by solving Eq. 3.67 as follows:

$$\mathbf{E}: \ \mathbf{p}_D^k = \mathbf{p}_1^k + r \cdot (\mathbf{p}_1^k - \mathbf{p}_2^k) + t \cdot (\mathbf{p}_1^k - \mathbf{p}_3^k) \stackrel{!}{=} \begin{pmatrix} 0 \\ 0 \\ -L \end{pmatrix} \tag{3.67}$$

Eq. 3.67 can be solved in least square sense if more than three vectors $\mathbf{p}_i^k$ are available. This way, the influence of noise in $\mathbf{p}_i^k$ can be reduced and leads to a more precise determination of L. With known camera tilt and driver tilt, the body height of approaching drivers can be computed from input data following Eq. 3.55. Figure 3.16 illustrates images of two real life parking scenarios where one car is parked on the road surface and one is parked close to a curbstone. For the later parking situation, the curbside is identified and indicates the need to estimate the height of the curbstone – the unknown offset Δ – using stereo techniques. In this application, the system returns a default value for body height if 3D-height information of the curbstone is not available.

3.4.8 Body height estimation and refinement

The input data sets obtained from real parking scenarios are noisy and may contain many outliers. Estimation of the body height that is based on noisy input data without any refinement stage may lead to imprecise results. Therefore, a refinement algorithm is proposed to remove outliers in the input data and, hence, to improve body height estimation. Figure 3.17 illustrates the block diagram of the proposed refinement algorithm.

The algorithm computes body heights h_i for all input data sets i using the camera tilt, the driver tilt and the ground distance that have all been estimated in previous processing stages. Based on single height values, the algorithm computes a mean height h_{mean} that is used to identify outliers.

Then, the algorithm removes input data sets whose estimated height values differ from the estimated mean height h_{mean} larger than a fixed difference $d_{hi} = |h_i - h_{mean}| > \delta_{thres} \cdot h_{mean}$. Hereby, a good value to identify outliers is $\delta_{thres} = 20\%$ of the mean height. The remaining input data sets are reused to refine camera tilt, driver tilt, the ground distance L and the body heights. The refinement process stops if there is no modification of the input data sets compared to previous iteration steps and if the maximum number of iterations has been exceeded.

Figure 3.16: Left: Parking on the ground plane. Right: Parking close to the sidewalk. The curbside of the sidewalk may be used to identify parking situations that have an offset Δ between the ground plane and the car ground plane.

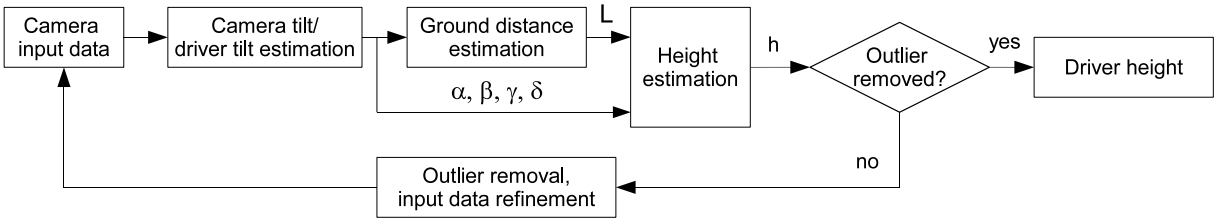


Figure 3.17: Block diagram of the proposed body height estimation and refinement algorithm.

3.5 Results

In this section, the performance of the proposed people extraction algorithm, the proposed camera tilt and driver tilt estimation and height estimation algorithm are evaluated. This evaluation has been run in terms of accuracy, quality of people extraction, quality of camera tilt and driver tilt estimation and number of iterations required for the minimization functions by means of several experimental results on real and simulated image data. Section 3.5.1 presents the results of the proposed people extraction algorithm. Results of camera tilt and driver tilt determination are presented in Section 3.5.2 and results of the body height estimation algorithm are presented in Section 3.5.3.

3.5.1 People extraction

To verify and to evaluate the Kalman-based people extracting algorithm with its extension for better shadow detection and illumination compensation (see Section 3.3), experiments have been conducted in complex environments containing weak and strong shadows as well as small differences between foreground and background using an omnidirectional vision system (ODVS). Image rectification was used to transform the captured images into panoramic images of size 480×204 pixels that are used to test the proposed algorithms under various conditions (dark and light regions, image noise and different resolutions due to image rectification and interpolation).

Quality of background initialization

In this section, the results on background initialization are presented. To compare different methods for background initialization, the background quality Q (see Eq. 3.68) is introduced: The background quality is defined as the ratio between the number of correctly extracted background pixels $N_{Background}$ and the number of ground truth pixels $N_{Groundtruth}$. A valid background pixel pb_i is a pixel that has a maximum difference $d = |pb_i - pbg_i| \leq THRES \quad \forall i \in [1, n]$ to the corresponding ground truth background pixel pbg_i, where $THRES = 5$ was chosen for the proposed people detection application.

$$Q_{background} = \frac{N_{Background}}{N_{Groundtruth}} \tag{3.68}$$

Training sequences from real life parking scenarios with many non stationary foreground objects were generated to evaluate the usability of the background initialization. Figure 3.18(a) illustrates the percentages of foreground pixels in frames from a chosen initialization sequence. Due to the large field of view of the camera, so many foreground objects in images are quite common for highly frequented road scenarios.

The background image is estimated using a block similarity matrix (SM) (see Section 3.3.1) whose entries are determined by calculating the similarity between the pixels (in a block) for each pair of frames. To speed up the execution time, averaging instead of SAD for block differencing was used. However, the quality of initialized background images using averaging is lower compared to the background image based on SAD (see Figure 3.18(b)), but experiments demonstrated a very fast adaptation of wrongly initialized background pixels.

Figure 3.18(c) and Figure 3.18(d) compare the method for background initialization based on averaging to the median-based [82] and to the Kalman-based [79] initialization. The initialization process started at different frames of the training sequence for a different number N of input images. The comparison demonstrates that at least 40 images are enough for SM-based background initialization compared to other methods where more images are needed for better initialization results. Thereby, the challenge for the other methods is the large number of pixels containing foreground content for more than half of input frames.

Experiments were also conducted to analyze the performance of background initialization in the presence of heavy snow. Fig. 3.19 illustrates training images of a snowy scenario (with highlighted snowflakes) and the resulting initialized background images. Simulations demonstrated the same performance for rainy scenarios. Thus, the proposed background initialization is highly robust against heavy snow and heavy rain.

Detection of shadow pixel candidates

To detect small differences in intensity between foreground and background objects, the threshold th_{bg} (see Eq. 3.8) must be low. An experimentally obtained value was $th_{bg} \geq 5$ that allows the detection of small intensity differences, but still many shadow pixels and noise are detected. Figure 3.20 illustrates the result of foreground detection (BG/FG) for one pixel over time using

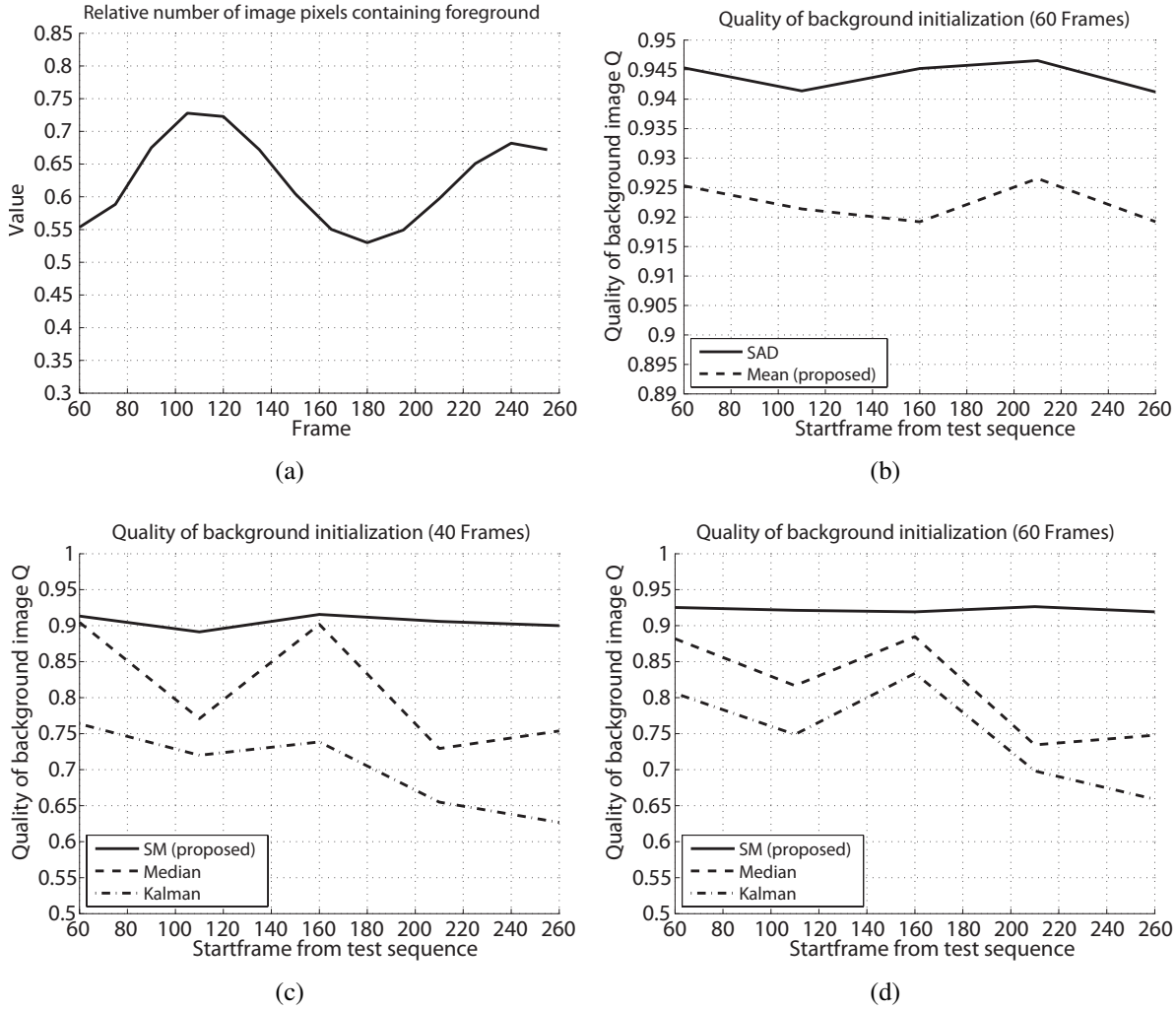


Figure 3.18: Number of image pixels containing foreground relative to the total number of image pixels (a). Differences in the quality using SAD and averaging (b). Quality of the obtained background images for a different number of training images started at different frames of the training sequence (c,d).

NCC and ZNCC. NCC was useful to pre-estimate shadow pixels, but valid foreground pixels are often misclassified as shadow pixels which can be seen on the noisy characteristic for the foreground (*BG/FG NCC*). ZNCC overcomes these limitations by taking textural changes into account, so that foreground pixels are not misclassified as background. The result is a smoother characteristic of the foreground/background values (*BG/FG ZNCC*).

Figure 3.21 compares the results of the proposed shadow detection to the shadow detection algorithm proposed by [79]. Ridder *et al.* assume that weak shadows have the same characteristic as illumination changes that may be adapted into the background. Therefore, their algorithm automatically increases the threshold for foreground detection using the variance (see Figure 3.21, variance) of the estimated background values over time. The threshold is high if the variance of the estimated background values (e.g. caused by shadows) is high. However, pixels from small

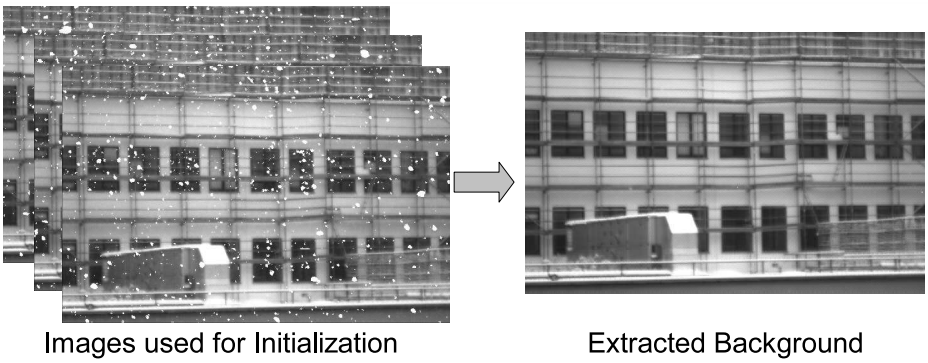


Figure 3.19: Image sequence to initialize the background for a snowy scenario (highlighted snowflakes) and the initialized background image.

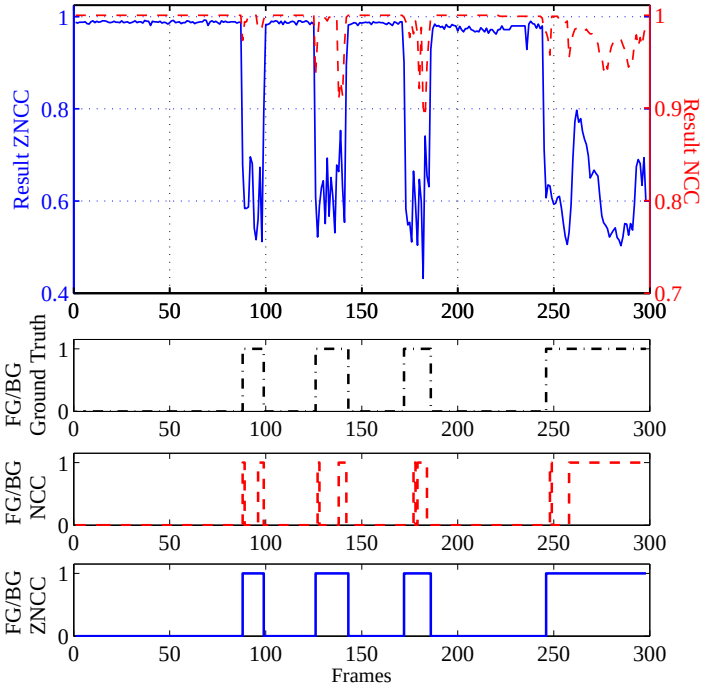


Figure 3.20: Characteristic of NCC and ZNCC from an image pixel (Top). Classification of foreground ($FG/BG = 1/0$) using NCC and the refinement using ZNCC (Middle and Bottom). ZNCC can better distinguish valid foreground from shadow.

foreground objects – such as short drivers who are far away from the car – also cause a high variance and could be suppressed (see frames 160-175).

Strong shadows cannot be identified as they are detected as foreground. Once detected as foreground it is impossible to differentiate between shadow and foreground (see frames 115-130). Increasing the threshold up to 15 may suppress strong shadows, but foreground objects with small differences to the background may be suppressed as well. Shadow detection based on NCC and shadow refinement based on ZNCC allow the use of a small threshold to extract fore-

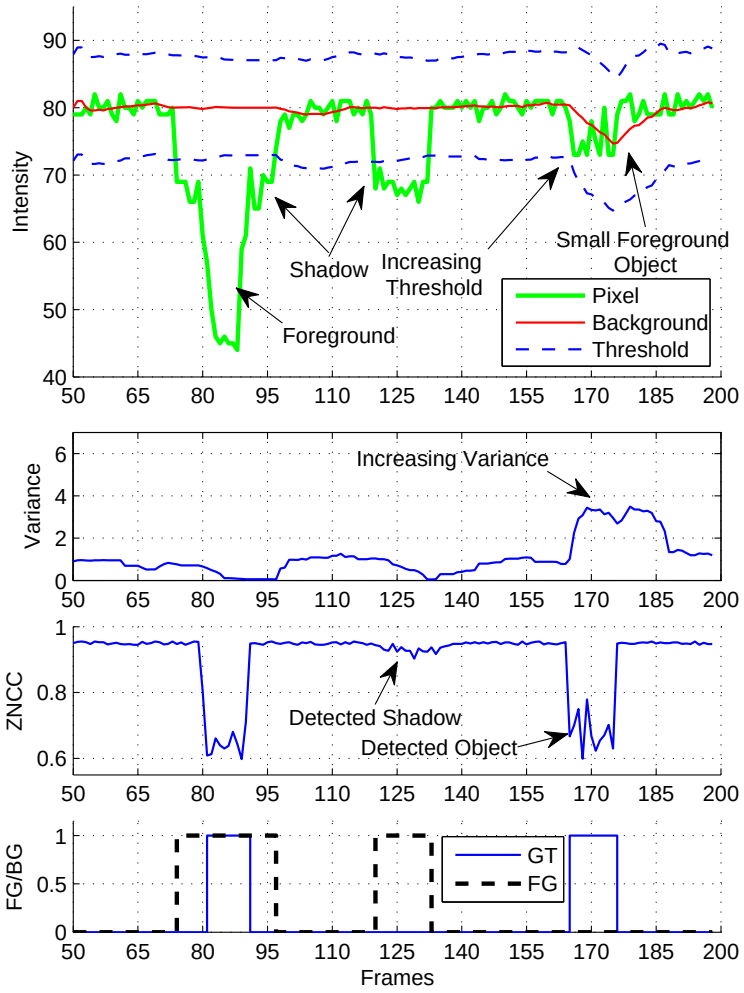
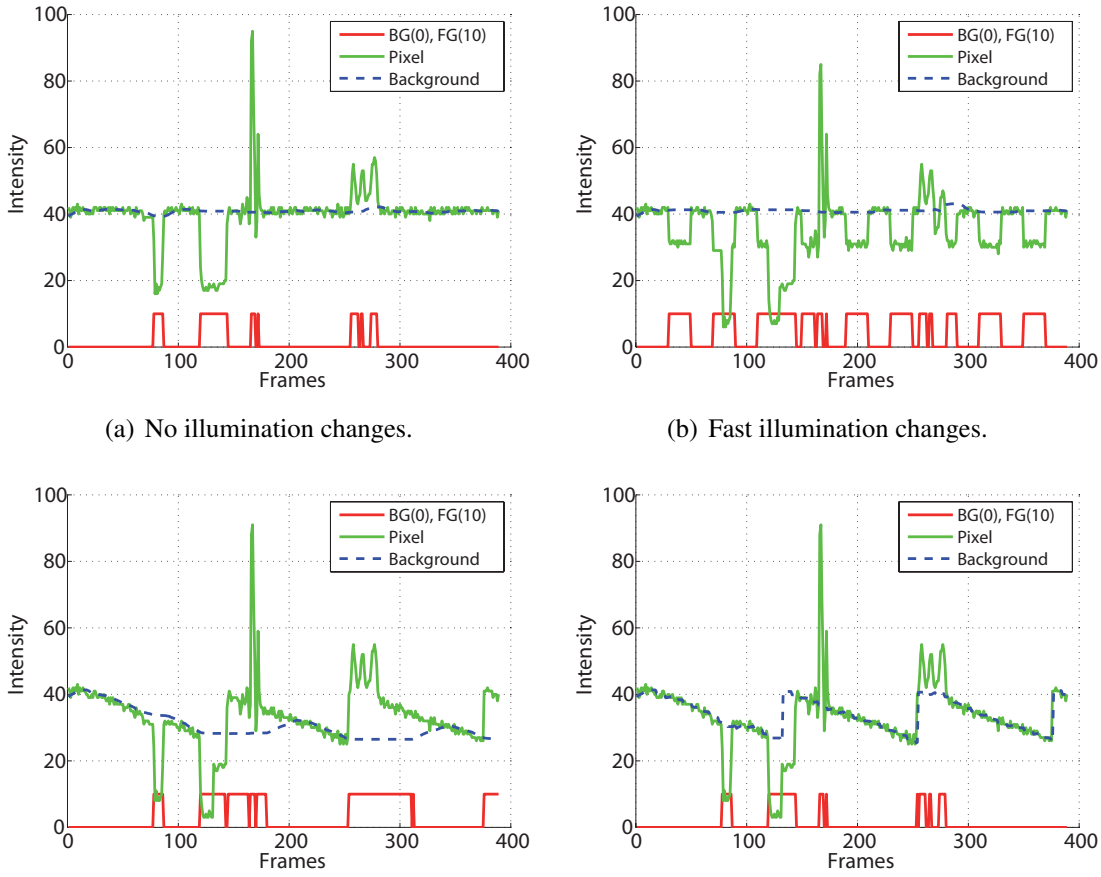


Figure 3.21: Image pixel over time with large and small foreground objects and strong shadow. The threshold containing information of the variance of background pixels over time is used to classify foreground or background pixels (Top). The characteristic of the variance for threshold adaptation proposed by Ridder *et al.* and the proposed shadow refinement technique (Middle). Foreground classification and shadow detection ($FG/BG = 1/0$) using the approach of Ridder *et al.* (dashed line) and the proposed approach (solid line, Bottom).

ground objects (see Figure 3.21, ZNCC) and is suitable for eliminating strong shadow borders that may be detected as foreground.

Illumination changes and background adaptation

Further experiments are conducted to analyze the capability of the proposed algorithm to compensate and to adapt illumination changes into the background model. Figure 3.22 illustrates experiments with various types of illumination change such as sudden or slow illumination



(a) No illumination changes.

(b) Fast illumination changes.

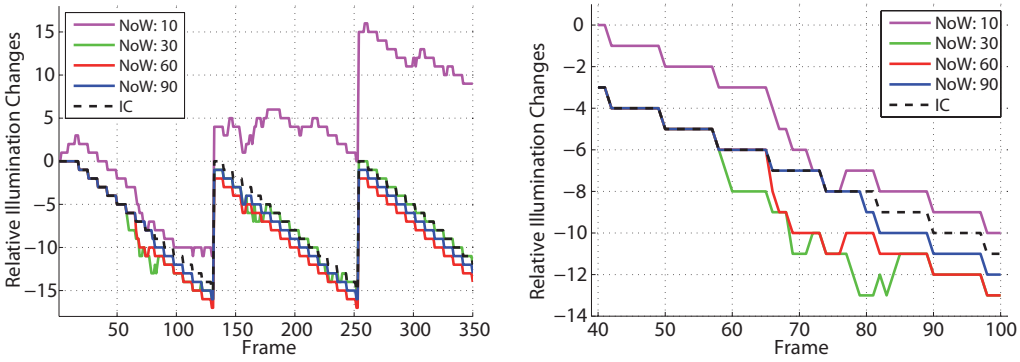
(c) Slow illumination changes.

(d) Compensation of illumination changes.

Figure 3.22: One pixel and the detected foreground over time (reference) (a). Misclassified foreground pixels caused by fast illuminations changes (b). Misclassified foreground pixels caused by slow illuminations changes (c). Adaptation of fast and slow illumination changes (d).

changes. Figure 3.22(a) presents a reference characteristic of the background for one pixel. Figure 3.22(b) and Figure 3.22(c) illustrate various characteristics of the background values for one pixel position which are disturbed by sudden and slow illumination changes. Experiments show that the background model considers slow illumination changes, even when the background was covered for a short time and when illumination changes were not too large (see Figure 3.22(c), frames $0 - 100$). Sudden illumination changes, which are larger than th_{bg}, cause wrong foreground information (see frames (280 - 310) and (380 - 400)). Finally, Figure (3.22(d)) demonstrates that illumination changes can successfully be compensated if they are considered by the background model (see Eq. 3.18).

Experiments were also conducted to find the optimal number of search windows (NoW) to detect global illumination changes. The number of search windows must be chosen so that the influence of illumination changes caused by foreground objects is minimized (see Eq. 3.16). A test profile of illumination changes (IC) was generated and tracked using a different number of search windows. For that purpose, a test sequence of panoramic images under different



(a) Illumination changes

(b) Frames 40 - 100

Figure 3.23: The more search windows (NoW) can be used for detecting illumination changes(IC), the better is the detection result. Good results give a NoW of 60 - 90.

illumination conditions following the profile of the illumination changes was generated and the illumination changes were detected using a different number of search windows. Experiments showed that at least 60 search windows were necessary to track the light profile sufficiently. The main problem of less than 60 search windows is that foreground objects lead to lighting changes which influence the detection of illumination changes.

This influence is almost suppressed using approximately 90 NoW. In Figure 3.23, the results of tracking the test profile using a different number of search windows were presented. This thesis also derived from the experiments that one search window should not be smaller than (15×15) pixels because of an increasing influence of image noise for smaller window sizes.

Validation of foreground pixels

Not all detected foreground pixels need to be valid (true positives = t.p.), i.e., there might also be false positives. For example, shadow pixels are often misclassified as valid foreground (false positives = f.p.). On the other hand, pixels having small differences to background can falsely be classified as background pixels (f.n., false negatives). Fig. 3.24 illustrates an example of a typical road scenario containing both true and false positives as well as false negatives. The algorithm is evaluated in terms of false negative, true positive and false positive detection rates under various conditions like diffused light, direct sunlight and indoor conditions and compared to results obtained with perfect detection.

These results are shown in Table 3.2, where the percentages were computed based on ≈200 test images. Shadow pixels in images with sunlit scenarios can easily be misclassified as valid foreground pixels. In general, small objects are also extracted in sunlit scenarios even when they are far away from the car: but only 76% of their pixels are classified as valid foreground. Clearly, such regions may consist of only 20 pixels – and approximately five pixels of such

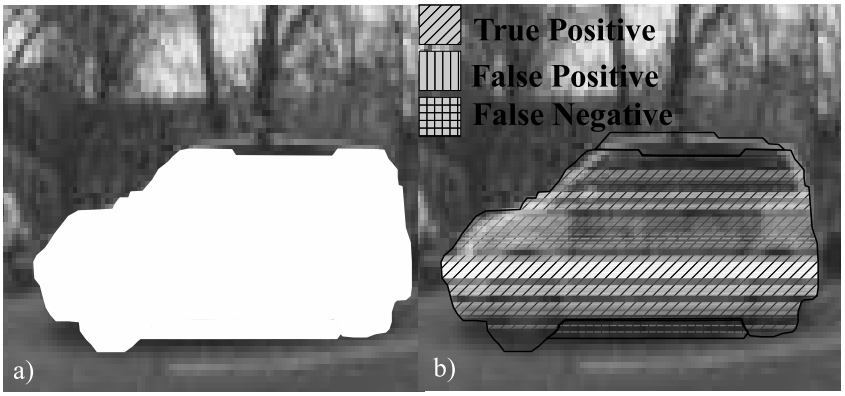


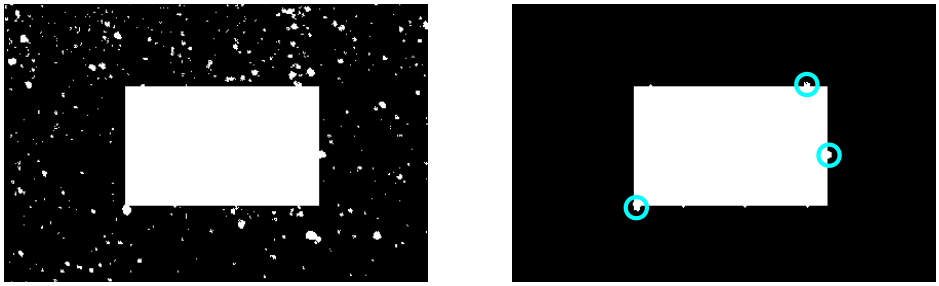
Figure 3.24: Detected pixels of an exemplary foreground object (a). Some pixels are not classified as background or shadow pixels and thus highlighted as true positives (b). There are also valid foreground pixels that are not highlighted as foreground (false negatives).

Scenario	Obj. Size	t.p.	f.n.	f.p.
Diffuse light	small	85%	15%	2%
	large	95%	5%	3%
Sunlight	small	76%	24%	7%
	large	93%	7%	10%
Indoor cond.	small	90%	10%	1%
	large	97%	3%	4%

Table 3.2: Overview of the obtained validation results: Percentages of true positive, false negative and false positive pixels in foreground regions.

objects not being classified as valid foreground result in a false negative rate of 25%. The false negative rates drastically decrease when objects approach to the car. Highly false negative rates may lead to incomplete foreground regions and hence to a bad body height estimation for objects far away from the car, but body height estimation becomes more precise for objects approaching the car. Good detection rates were achieved for large foreground objects in all tested scenarios. Clearly, having a large fraction of misclassified pixels results in an object not being detected. Furthermore, shadow pixels in images with sunlit scenarios can easily be misclassified as valid foreground pixels.

Figure 3.25 illustrates the detection of a simulated object surrounded by snowflakes in a snowy scenario. The snowflakes are detected as small, rapidly moving objects and can be removed using median filtering [97]. Figure 3.25(b) illustrates the detection result of an object in a snowy scenario when snowflakes were removed by median-filtering. Large snowflakes close to the camera overlapping the boundary of objects cannot be removed using median-filtering and may lead to wrong body height estimation results. However, tracking the object over long image sequences or using cameras with high frame rates overcomes this limitation. Since snowflakes move very fast, the number of images containing inaccuracies in the object's boundaries caused by snowflakes or heavy rain in a huge data set is small.



(a) (b)

Figure 3.25: Simulated foreground object and extracted snowflakes (a). Removed snowflakes using median-filtering and remaining disturbances at the boundary of a detected object (b).

time	Averaging	Similarity	Decomposition	Total Time
t_0	$\approx$ 2.7 ms	-	-	2.7 ms
t_1	$\approx$ 2.7 ms	$\approx$ 0.2ms	-	2.9 ms
t_2	$\approx$ 2.7 ms	$\approx$ 0.4ms	-	3.1 ms
t_3	$\approx$ 2.7 ms	$\approx$ 0.6ms	-	3.3 ms
t_{39}	$\approx$ 2.7 ms	$\approx$ 8.4 ms	-	11.1 ms
t_{40}	$\approx$ 2.7 ms	$\approx$ 9.0 ms	-	11.7 ms
t_{41}	-	-	$\approx$ 412 ms	412 ms

Table 3.3: This table illustrates the computation time for the C-implemented background initialization on a 2.54 GHz AMD Phenom 9650 Quad-Core CPU.

Execution time and parallelization

A complex indoor environment with three walking people, shadow effects and some illumination changes (switching light on/off) was chosen to measure the mean execution time of the proposed foreground detection algorithm. About 400 test images of this data set were used for execution time analysis and the mean execution time as well as the standard deviation (Std. Dev.) were computed. The proposed algorithm was realized in a C-based implementation. Table 3.3 gives an overview of the execution times for the proposed background initialization algorithm (see Section 3.3.1) using $N = 40$ input frames with size 480×204 and a block size of 9×9 pixels. The advantage of similarity computation based on averaging is the reuse of previously computed data. Table 3.3 illustrates a constant execution time for block averaging for every new input frame, and an increasing computation time for similarity computation. This computation time increases due to an increasing number of similarity computations for each incoming new frame: For instance, the similarity of one block at a pixel position in frame 2 has to be computed with the corresponding block in frame 1. For a third frame, the similarity of one block at a pixel position in frame 3 has to be computed both with the corresponding block in frame 2 and frame 1, and so on. While the similarity of two blocks can be computed between two incoming frames from a camera having a frame-rate of 30 frames per second, matrix decomposition takes $412ms$ for execution.

	Mean Time (1 Core)	Std. Dev.	Image Size	2 Core	4 Core
Rectification	$\approx$ 4.5 ms	0.7 ms	640 $\times$ 480	$\approx$ 2.3 ms	$\approx$ 1.7 ms
Background	$\approx$ 30.1 ms	1.2 ms	480 $\times$ 204	$\approx$ 15.1 ms	$\approx$ 7.5 ms
Shadow Dect	$\approx$ 24.1 ms	2.3 ms	”	$\approx$ 12.2 ms	$\approx$ 6.3 ms
Ill. Comp	$\approx$ 10.4 ms	1.6 ms	”	$\approx$ 5.6 ms	$\approx$ 2.8 ms
Interpolation	0.0 ms	0.0 ms	”	$\approx$ 2.0 ms	$\approx$ 2.2 ms
Total Time	69.1 ms			37.2 ms	20.5 ms

Table 3.4: This table illustrates the computation time for the C-implemented, non parallelized foreground detection algorithm (left) and for the C-implemented, parallelized foreground detection (right) on a 2.54 GHz AMD Phenom 9650 Quad-Core CPU.

The foreground detection algorithm has been parallelized using multi-threading on a quad-core CPU (see Section 3.3.6). Therefore, the image was subdivided into n sub-images, and each sub-image was processed by a concurrent thread. The result of all threads is then merged using a small thread called interpolation. Table 3.4, on the left side demonstrates the execution times for rectification using bilinear interpolation, background modeling, shadow detection and illumination changes using a single core of a 2.54 GHz AMD Phenom 9650 quad-core CPU. Table 3.4, on the right side gives an overview of the measured times using two and four cores. Table 3.4 also illustrates increasing time for merging and interpolation, whereas the total computation time decreases with an increasing number of threads. Both the algorithms for background initialization and foreground detection were implemented in C.

Finally, the efficiency of the proposed foreground extraction method has been compared to other well-known algorithms presented in Section 3.2 using various test scenarios. Figure 3.26 illustrates one of the evaluated scenarios containing up to 500 test frames. Difficulties of this scenario are a less textured environment and both weak and strong shadows induced by different sources of light. While the approach in [79] modeled the background well, shadow detection fails in some cases. Similarly, while the shadow detector in [82] performed well, the background model of [82] has some limitations. One limitation is that once the background is learned the background model is not updated. This results in many noisy foreground pixels caused by illumination changes etc. The combination of both algorithms and the modification of the shadow detector as well as the light compensation led to a powerful background estimator which resulted in better foreground detection on gray-scaled images, when compared with state of the art techniques (see Figure 3.26).

3.5.2 Camera tilt, driver tilt and ground distance estimation

After driver extraction and torso determination, estimation the camera tilt, the driver tilt and the ground distance L based on head and foot points is the first step to determine the body heights of approaching drivers. In this section, the results of camera and driver tilt as well as ground distance estimation are presented. In Section 3.4.5, a highly robust algorithm has been presented that estimates the camera pose by solving a multi-objective goal attainment problem using se-

Original image sequence.

Foreground detection using Mixture of Gaussian and low thresholding results in misclassification (shadows, noise).

Less noise in background modeling [79], but detected strong shadows pixels.

Good shadow detection is achieve in [82] but still image noise.

Our approach with proposed shadow and foreground detection. Noise and shadows are better suppressed.

Figure 3.26: Evaluation of the proposed foreground detector with different background models and shadow detection algorithms.

quential quadratic programming. Due to very long execution times, a fast algorithm has been introduced in Section 3.4.6 to determine the camera poses by solving the standard minimization problem $\underset{\alpha,\beta,\gamma,\delta}{min}\sum_{i=1}^{n}\mathbf{f}^2(\alpha,\beta,\gamma,\delta)\ \forall i \in \mathbb{N}$. The following experiments described in Section 3.5.2 and Section 3.5.3 consider both algorithms. The algorithms are not explicitly named if there are no significant differences in the results for both algorithms. In Section 3.4.5, a multiobjective goal attainment minimization algorithm is presented to estimate the camera and driver tilt by minimizing a model-based function f_{hm} using samples of head and foot points obtained from approaching drivers (see Eq. 3.58). Since the input data sets depend on the camera and driver tilt, the minimization function becomes zero if the estimated camera tilt and driver tilt

best match with the real camera and driver tilt. The goal attainment minimization converges in highly precise camera and driver tilts but the limitation of this algorithm is the prerequisite for initial points $\mathbf{x}_0$ close to the global minimum. Therefore, an algorithm is proposed to iteratively determine potential initial points (see Figure 3.13). This algorithm consists of two iteration stages. One iteration stage that is called *Inner Loop* or *border modification* and a second stage that is called *Main Iteration* (see Figure 3.13). The border modification stage restricts and refines the maximum search area to select optimal starting points. With these starting points, the main iteration stage minimizes function f_{hm} to find an appropriate estimation for the camera and driver tilt. However, if the global minimum has not been determined due to poor minimization results using the extracted starting points, then the main iteration stage increases the number of potential initial points and repeats the border modification procedure until best starting points have been obtained.

Table 3.5 illustrates the relative number of main iteration stages (*Main Iteration*) that are required for estimating camera and driver tilt using the goal attainment minimization algorithm. For the remainder of this thesis, camera tilt and driver tilt estimation is summarized as *pose* estimation. Seventy percent of all poses can be estimated using one main iteration stage and 4.8% of all poses require four main iteration stages. Additionally, Table 3.6 and Figure 3.27 give an overview of the mean number of border modification stages *Inner Loop* within each of the four main iteration stages. Experiments illustrated that normally 0, 1 or 2 border modification stages are required for estimating a suitable minimum for camera pose determination.

In further experiments, the number of potential starting points was determined that are recommended to optimally select initial points. Table 3.7 illustrates the optimal number of starting points for each main iteration stage. 256 starting points are used for the first and 10000 starting points are used for the fourth main iteration stage. Therefore, an increasing number of starting points is spread over the whole search area for each main iteration stage to find appropriate initial points close to a global minimum. In this manner, appropriate initial points may be found in further iteration stages if initial points obtained from previous stages led to bad minimization results.

Experiments are also conducted to illustrate the characteristics of the number of starting points and the occurrence of border modifications within one or several main iteration stages (see Figure 3.28). The upper part of Figure 3.28 shows the characteristics of the number of starting points (solid line) over the iteration steps that are required by the algorithm to select appropriate initial points close to the local minimum. The algorithm selects strong initial points by iteratively strengthening test conditions and by deleting points that do not pass the test. In this manner, an appropriate number of initial points can be found within one or several main iteration stages. The lower part of each figure in Figure 3.28 illustrates the occurrence of border modifications (red, solid bars) within each main iteration stage. Additionally, these figures also indicate the beginning of new main iteration stages (blue, dashed bars) within the minimization processes requiring one, two, three and four main iteration stages to estimate the camera poses by means of the initial points.

As mentioned above, the initial point determination is based on the minimization function f_{hm}. In a further experiment, both the minimization error and the final error for camera tilt and

Main Iteration	1	2	3	4
Percentage	70.0 %	18.2 %	7.0 %	4.8 %

Table 3.5: Relative number of *Main Iteration* stages required for camera/driver tilt estimation (computed across 2000 poses).

Border Mod.	0	1	2	3	4	5	6	7	8	9	10	11
1 Main It.	58.9	29.1	9.3	2.1	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2 Main It.	33.6	24.9	20.8	7.9	6.3	3.0	1.4	0.8	0.3	0.8	0.0	0.3
3 Main It.	21.6	18.0	23.0	12.9	6.5	7.2	4.3	3.6	0.0	1.4	1.4	0.0
4 Main It.	13.7	18.9	14.7	16.8	11.6	8.4	7.4	2.1	2.1	1.1	1.1	2.1

Table 3.6: Relative number [%] of border modifications (*Inner Loop*) that are required to estimate camera and driver tilt using 2000 test poses. For some pose estimations, two or more *Main Iteration* stages are required.

Main Iteration	1	2	3	4
Num. of Starting Points	256	12962	4096	10000

Table 3.7: Number of starting points used for initial point determination for each main iteration stage.

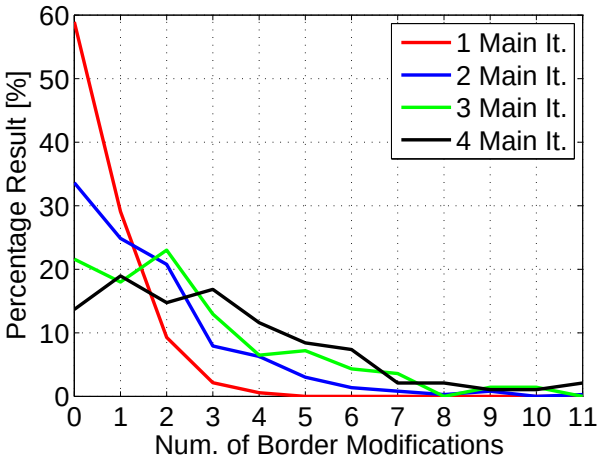
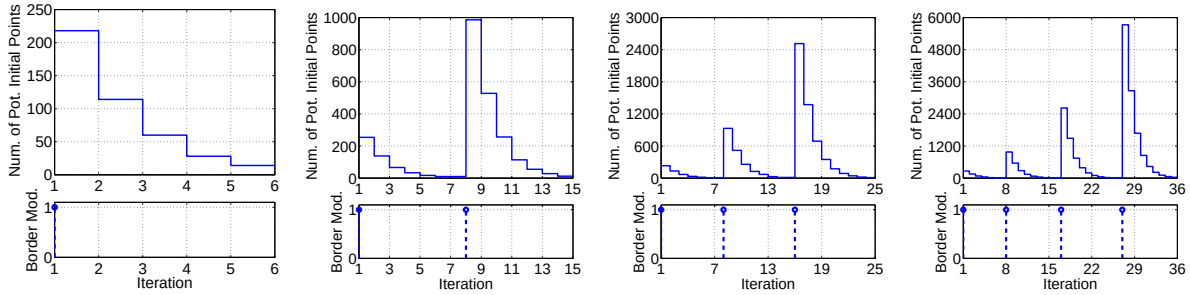
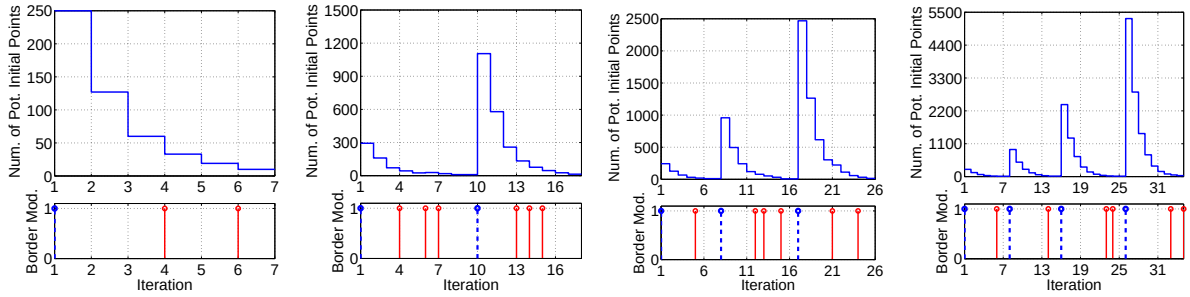


Figure 3.27: Mean number of border modifications that are required to estimate camera and driver tilt (computed across 2000 poses)

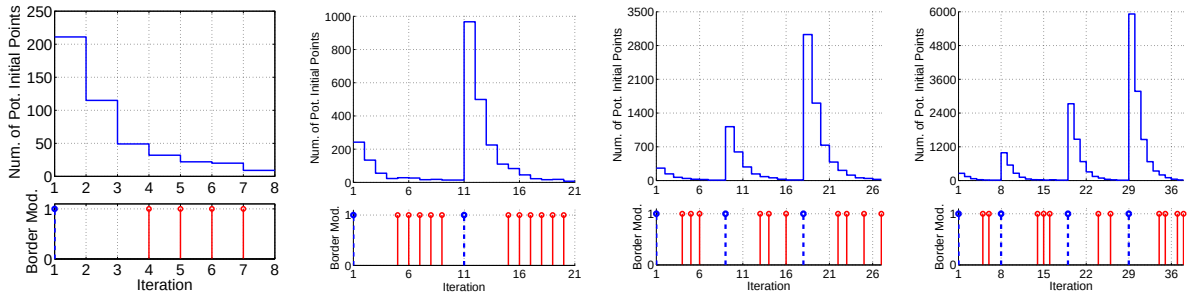
driver tilt estimation are determined. Both the goal attainment algorithm and the standard minimization algorithm display an error indicating the quality of the minimization results. This error indicates how good the design goals have been achieved during minimization – very small values close to zero indicate good achievements of the design goals and vice versa. Figure 3.29 illustrates the (mean) pre-estimation error and the final minimization error. The algorithm selects appropriate initial points, which serve as an input to the final minimization, by computing pre-estimation errors for the remaining starting points (solid line, upper figure) after each it-



Examples of 1, 2, 3 and 4 main iteration stages without border modification stages.



Examples of main iteration stages with 2 and 6 border modification stages.



Examples of main iteration stages with 4, 10 and 11 border modification stages.

Figure 3.28: This figure illustrates the characteristics of the number of starting points (solid lines, upper part of the figures) and the occurrence of border modifications (solid bars, lower part of the figures) within one or several main iteration stages (dashed bars, lower part of the figures) over the number of iterations that are required to select appropriate initial points close to the local minimum of f_{hm} and to estimate camera and driver tilt.

eration step and by computing the mean error of all pre-estimation errors (solid line, lower figure). New starting points are then selected based on the mean pre-estimation error for the next iteration stages. The solid bars in the lower part of Figure 3.29 illustrate the final error of the minimization using the selected initial points and goal attainment minimization. It can be seen that the mean error of the pre-selected starting points can be lower than the final error for camera and driver tilt estimation using goal attainment minimization, especially for the first or second main iteration stage. This means that initial points were found that match well with local minimas of function f_{hm}. But these points are not suitable as initial points for estimating camera and driver tilt since the goal attainment minimization would converge to a local min-

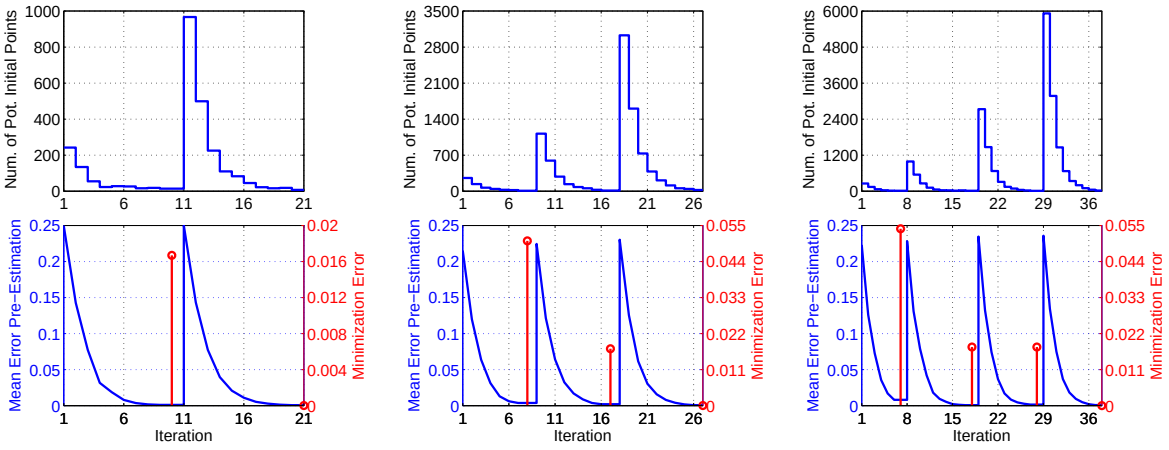


Figure 3.29: Mean pre-estimation error of the determined initial points over the iterations of the algorithm that are required to select optimal starting points within two, three and four main iteration stages (solid lines). Solid bars indicate the minimization error after each main iteration.

It.	Error	α_R	β_R	γ_R	δ_R	α_E	β_E	γ_E	δ_E
1	$1.3e^{(-5)}$	-0.191	-0.332	-0.161	0.334	-0.091	-0.332	0.199	0.398
2	$7.4e^{(-6)}$	-0.208	0.078	-0.016	0.319	-0.399	0.267	0.267	0.267
4	$2.1e^{(-6)}$	-0.271	-0.306	-0.081	0.265	0.399	0.200	0.402	0.401

Table 3.8: Wrongly determined poses having a minimization error less than $10e^{-4}$.

imum. Additional main iteration stages with an increased number of starting points help to better estimate the global minimum of f_{hm} and, hence, lead to small minimization errors (e.g. less than $10e^{-4}$).

In Section 3.4.5, a threshold has been introduced to guarantee a convergence to the global minimum of the minimization function f_{hm} using both the goal attainment minimization and the standard minimization algorithm (see Figure 3.13). Experiments demonstrated that a minimization error smaller than a fixed threshold $th < 10e^{-4}$ yields good detection results for both ideal and noisy input data. A larger threshold may lead to higher detection rates, but the detection rate of local minima drastically increases using the presented minimization algorithm. A smaller threshold yields very precise pose estimation results, but the detection rate to obtain the camera pose for noisy input data drastically decreases. In this case, it is no longer possible to estimate poses from noisy input data.

However, camera and driver tilt configurations exist that result in low minimization errors, but the real and the estimated tilt values strongly differ from each other. Table 3.8 illustrates examples of such pose configurations: The real tilt values are presented by α_R, β_R, γ_R and δ_R and the estimated tilt values are represented α_E, β_E, γ_E and δ_E.

Moreover, the quality of the computed ground distance L and the predicted body height highly relates to the quality of the extracted camera and driver tilt. Table 3.9 illustrates the influence

It.	Error	L'	L_R	h_R	L_E	h_E
1	$1.3e^{(-5)}$	1.12m	1.53 m	1.80 m	1.34 m	1.71 m
2	$7.4e^{(-6)}$	1.12m	1.48 m	1.80 m	1.86 m	4.61 m
4	$2.1e^{(-6)}$	1.12m	1.69 m	1.80 m	1.47 m	2.93 m

Table 3.9: Influence of wrongly determined poses on the estimated ground distance L_E and the determined body height h_E.

Goal Att.	Standard
98.55%	93.75%

Table 3.10: Convergence rates for both the goal attainment and standard minimization algorithm.

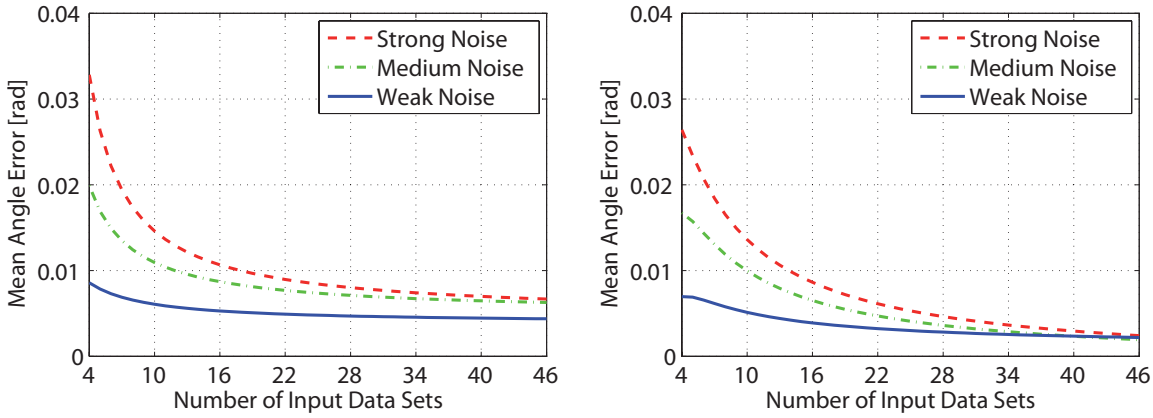
of wrongly determined camera and driver tilt on the ground distance L and the body height h. Variables L_R and h_R represent the real ground distance and the real body height, whereas L_E and h_E represent the estimated ground distance and the determined body height based on the camera tilt and driver tilt presented in Table 3.8. Finally, Table 3.10 illustrates the detection rates for both goal attainment minimization and standard minimization. The detection rate is a value indicating the relative number of convergences for a certain number of minimizations. It can be seen that the goal attainment minimization algorithm leads to better detection results compared to the standard minimization algorithm. This might be explained, as goal attainment minimization explicitly considers noise in the input data that can be modeled with the design goals.

Influence of noise on pose estimation

To analyze the robustness of the proposed camera tilt and driver tilt estimation algorithms, experiments with input data sets containing weak, medium and strong noise have been conducted. Figure 3.30 illustrates the interpolated detection error for camera and driver tilt estimation over the number input data sets. Figure 3.30(a) illustrates the estimation error for the proposed fast camera and driver tilt estimation algorithm, whereas Figure 3.30(b) illustrates the estimation error obtained for the goal attainment minimization algorithm. In Figure 3.30, the extraction error for camera and driver tilt seems to be very small (< 0.035 rad) for both implementations, but this small error may result in a large height estimation error. This is described in the next Section 3.5.3.

3.5.3 Body height estimation

In this section, the results of the realized body height estimation algorithm are presented and discussed in terms of accuracy and execution time. Previous sections describe two algorithms to estimate the camera pose based on a goal attainment and on a standard minimization problem. The first part of this section presents the height error for both an unknown offset Δ within the ground distance L and the height error made by a wrongly chosen height model for an



(a) Standard minimization

(b) Multi objective goal attainment

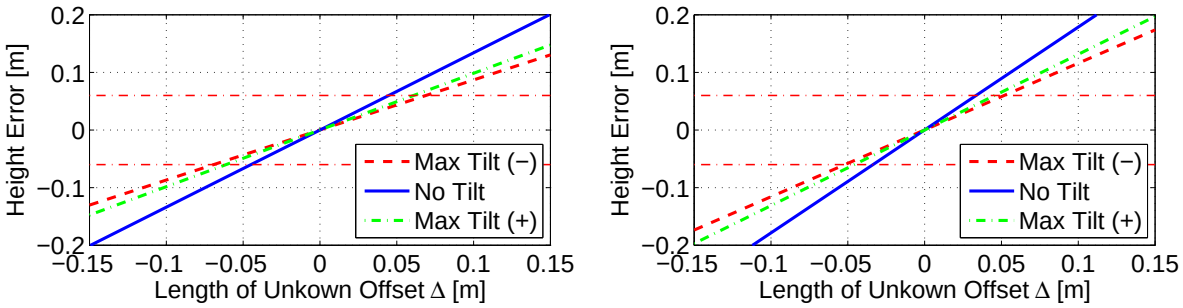
Figure 3.30: Error in camera and driver tilt estimation using input data with weak, medium and strong noise.

underlying parking scenario. The second part of this section describes the influence of noise in the data sets to height estimation and compares both methods in terms of accuracy, the number of input data sets that are required to obtain good minimization results and the execution time. This section ends with a timing analysis for the absolute height estimation process beginning with background initialization and ending with absolute body height computation.

Height error due to an offset Δ in L

The intersection point of the *car ground plane* and the *ground plane* (see Section 3.4.1) is not located in one of the car wheels if a vehicle is parked on the road and if there is a curbstone next to the car on which drivers approach. Such a parking situation leads to an unknown offset Δ within the ground distance L. Consequently, a new ground distance $L_{new} = L + \Delta$ must be determined considering the offset Δ. However, this offset Δ and L_{new} cannot be estimated using image correspondences of one camera only. To study the error of height estimation caused by this offset Δ, experiments were conducted with subjects that have different body heights and the height error is computed.

Figure 3.31 illustrates the estimation error influenced by the unknown offset $\Delta \neq 0$ for both short persons (1,5m, see Figure 3.31(a)) and tall persons ($2.0m$, see Figure 3.31(b)). The dashed lines illustrate the height error for a maximum positive camera tilt $\alpha = 0.4$, $\beta = 0.4$, and for the maximum negative camera tilt $\alpha = -0.4$, $\beta = -0.4$ over the offsetΔ. The solid line indicates the error for a camera system whose orientation is coincident with the world coordinate system. In general, the ground distance L depends only on the calibrated distance L' and on the camera tilt α, β (see Section). Therefore, the driver tilt does not influence the height error. It can be seen that the height error increases for decreasing camera tilts and vice versa. This is caused by the term $cos(\alpha) \cdot cos(\beta) \cdot L$ in Eq. 3.55: The trigonometrical term $cos(\alpha) \cdot cos(\beta)$ can be assumed to be a scale factor and scales the ground distance L depending on the camera tilt.



(a) $h = 1.5m$ (b) $h = 2.0m$

Figure 3.31: Influence of the unknown offset to height estimation for a short person (a) and a tall person (b). *Max Tilt (-)* represents the error assuming the maximum negative camera tilt -0.4 and *Max Tilt (+)* represents the error assuming the maximum positive camera tilt 0.4.

Increasing the camera tilt leads to a decreasing length L of the ground distance vector and also to a decreased influence of the offset Δ within L. Figure 3.31 also illustrates the influence of the physical body heights of approaching drivers influences on the height error. The unknown offset Δ leads to larger height errors for tall persons than for short persons.

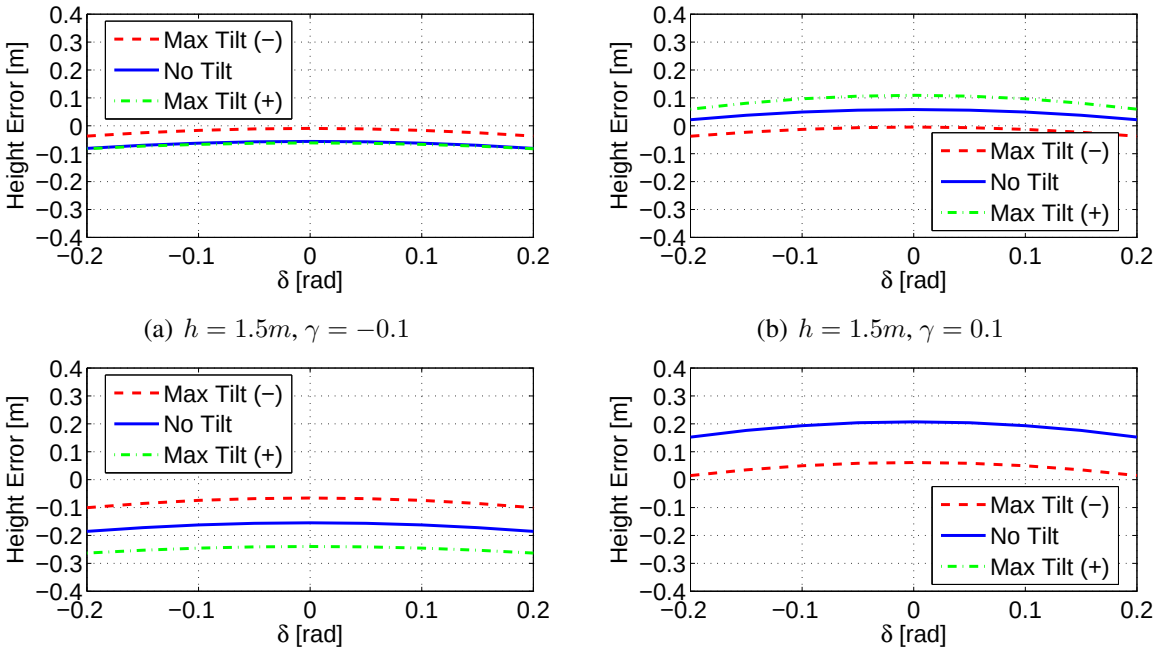
In ergonomics, seats can be pre-adjusted up to a height error of $h_\epsilon = \pm 6cm$. Therefore, an unknown offset $|\Delta| < 6cm$ for short persons and an unknown offset $|\Delta| < 5cm$ for tall persons can be tolerated for body height estimation.

Height errors due to wrongly chosen height models

In this section, the characteristics of height errors are presented for actual parking scenarios where wrong height and parking models have been chosen. In a first experiment, the height error is determined using the parking model of the *Curbstone Parking Scenario* (see Section 3.4.2) for an actual *Inclined Parking Scenario* (see Section 3.4.3).

Height error caused for inclined parking scenarios

Figure 3.32 illustrates the characteristics of height errors that occur in an real *inclined parking scenario* if a wrong parking model – the *curbstone parking scenario* – has been chosen for height computation. The height error obtained from this model depends on the driver tilt γ and has been computed for both short persons ($h = 1.5$, see Figure 3.32(a), 3.32(c)) and tall persons ($h = 2.0$, see Figure 3.32(b), 3.32(d)). In this experiment, driver tilt γ is assumed to be constant ($\gamma = -0.1$ and $\gamma = 0.1$) for all tested scenarios. Additionally, the experiments consider a maximum negative tilt of the camera (Max Tilt (-)) $\alpha, \beta = -0.4$, a maximum positive tilt of the camera (Max Tilt (+)) $\alpha, \beta = 0.4$ and the scenario where the orientation of the camera coordinate system is coincident with the orientation of the world coordinate system (tilt $\alpha, \beta = 0.0$).



(a) $h = 1.5m, \gamma = -0.1$

(b) $h = 1.5m, \gamma = 0.1$

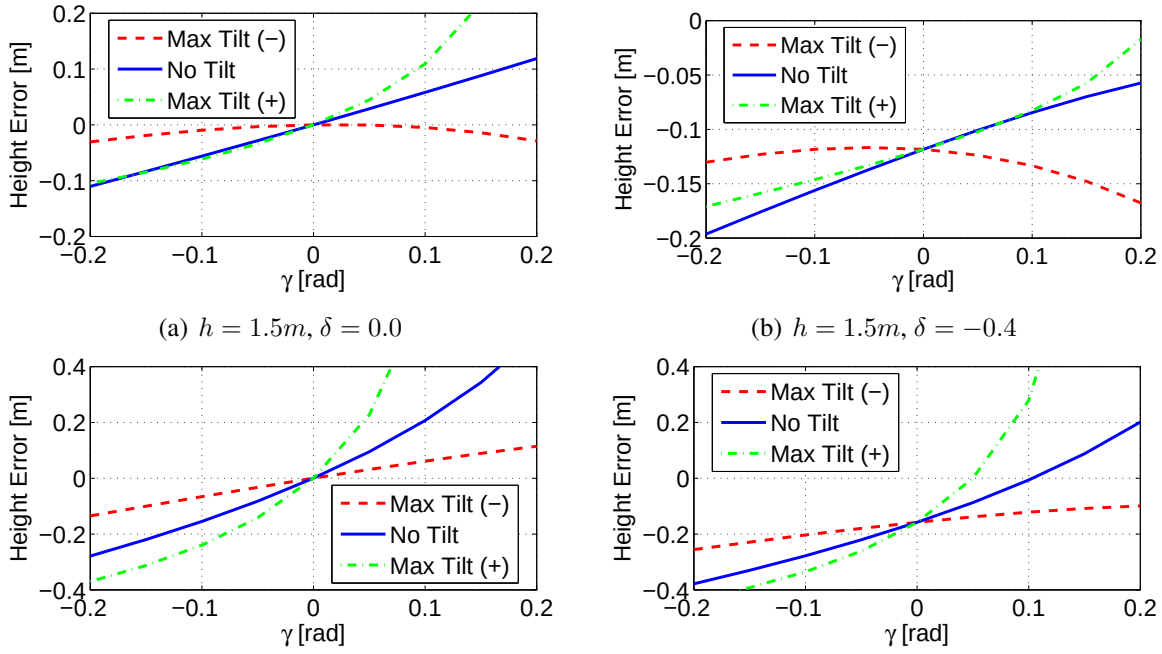
(c) $h = 2.0m, \gamma = -0.1$

(d) $h = 2.0m, \gamma = 0.1$

Figure 3.32: Height error incurred by a wrongly chosen height model for a short person ($1.5m$, (a),(b)) and for a tall person ($2.0m$, (c), (d)). Height estimation is based on the curbstone parking scenario for a real inclined parking scenario (fixed driver tilt γ, varying δ).

Following Eq. 3.40, δ relates to the height estimation within the term $cos(\delta)$. For this reason, resulting height errors are symmetric with respect to $\delta = 0.0$. The figures also illustrate that even low driver tilts δ and γ could result in a large height error. Moreover, the use of the height model of the curbstone parking scenario for a real inclined parking scenario results in a large offset. This leads to a height error even if there is no driver tilt ($\delta = 0.0$). The height model for the curbstone parking scenario assumes a changing ground distance L that depends on the camera tilts α and β (see Eq. 3.27), whereas the inclined parking scenario does not consider camera tilt. This leads to a large height error even for scenarios without driver tilt. The height error obtained for the inclined parking scenario also depends on the height of drivers, i.e. short persons cause a smaller height error than tall persons.

Similarly to Figure 3.32, Figure 3.33 illustrates the characteristics of height errors over driver tilt γ for two fixed driver tilts $\delta = 0.0$ and $\delta = -0.4$. As mentioned above, δ affects height estimation within the term $cos(\delta)$ (see Eq. 3.40) so that computed height errors are independent of positive and negative values for δ. Figure 3.33 also illustrates the characteristics of height errors over the driver tilt γ. These errors are computed for both a short person ($h = 1.5$, see Figure 3.33(a), 3.33(c)) and a tall person ($h = 2.0$, see Figure 3.33(b), 3.33(d)) for two fixed driver tilts $\delta = 0.0$ and $\delta = -0.4$. Contrary to Figure 3.32, the offset in the characteristic of height error is zero for a driver tilt $\delta = 0.0$. This offset depends on body height and increases for increasing driver tilts δ.



(a) $h = 1.5m, \delta = 0.0$

(b) $h = 1.5m, \delta = -0.4$

(c) $h = 2.0m, \delta = 0.0$

(d) $h = 2.0m, \delta = -0.4$

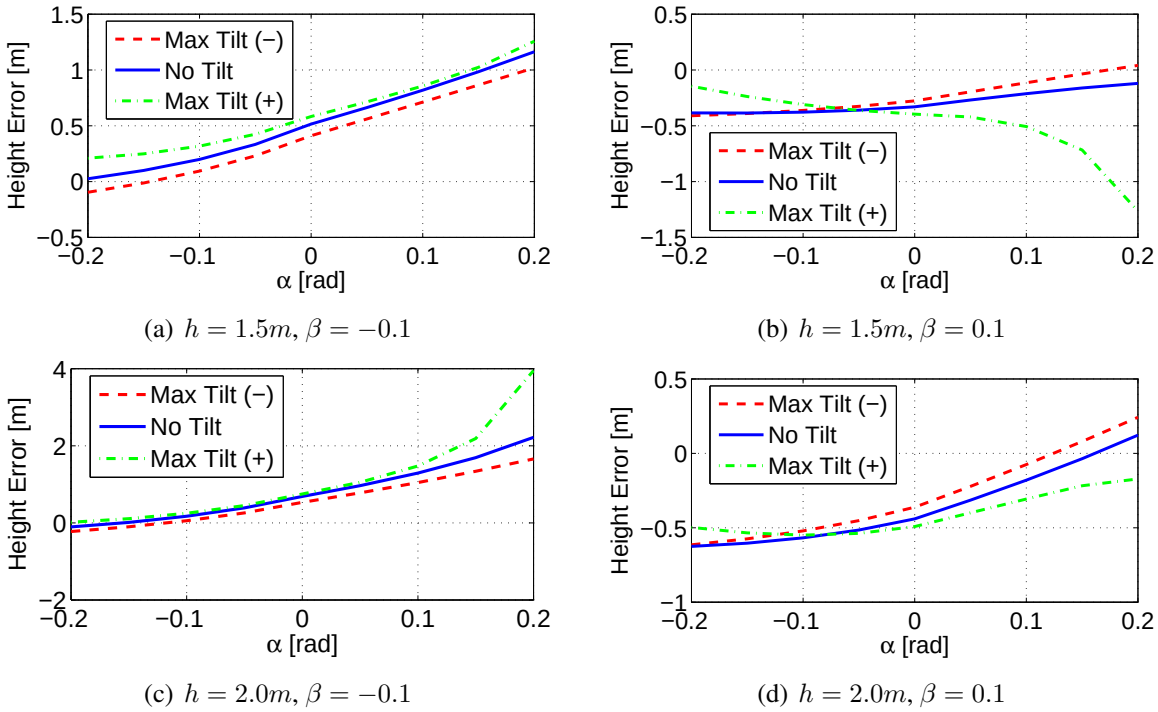
Figure 3.33: Height error incurred by a wrongly chosen height model for a short person ($1.5m$, (a),(b)) and for a tall person ($2.0m$, (c),(d)). Height estimation is based on the curbstone parking scenario for a real inclined parking scenario (fixed driver tilt δ and varying γ).

Height error caused for curbstone parking scenarios

In a second experiment, the height error is determined for an actual *curbstone parking scenario* if a wrong parking model – the *inclined parking scenario* – has been chosen for height computation. Figure 3.34 illustrates the characteristics of the height error that depends on the camera tilt α and that has been computed for both a short person ($h = 1.5$, see Figure 3.34(a), 3.34(c)) and a tall person ($h = 2.0$, see Figure 3.34(b), 3.34(d)). In this experiment, the camera tilt β is assumed to be constant for all tested scenarios ($\beta = -0.1$ and $\beta = 0.1$).

For the inclined parking scenario, the experiments consider test scenarios with a minimum driver tilt (Max Tilt (-)) $\gamma, \delta = -0.4$, a maximum positive driver tilt (Max Tilt (+)) $\gamma, \delta = 0.4$ and a test scenario without tilt between the world coordinate system and the driver ($\gamma, \delta = 0.0$). Figure 3.34 presents the characteristic of the height error over the camera tilt α, and Figure 3.35 the characteristic of the height error over the camera tilt β.

Similarly to the first experiment, there is a large offset in the estimation error for values $\alpha = 0.0$ and $\beta = 0.0$ resulting from a wrongly determined ground distance L due to a wrongly chosen height model. The ground distance L in the height model of the curbstone parking scenario depends on the detected camera tilt. The height model of the inclined parking scenario considers only the driver tilt in input data sets meaning that the car ground plane is co-linear with the ground plane. For this parking scenario, the height model does not consider the camera tilt



(a) $h = 1.5m$, $\beta = -0.1$

(b) $h = 1.5m$, $\beta = 0.1$

(c) $h = 2.0m$, $\beta = -0.1$

(d) $h = 2.0m$, $\beta = 0.1$

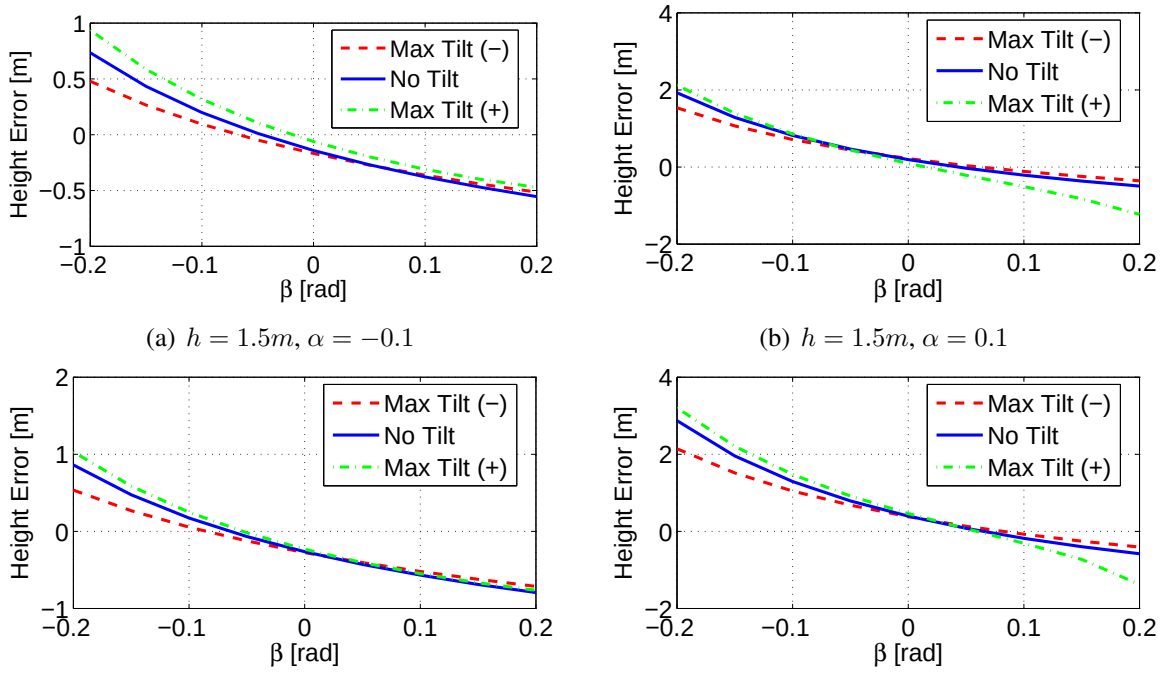
Figure 3.34: Height error caused by a wrongly chosen height model for a short person ($1.5m$, (a),(b)) and for a tall person ($2.0m$, (c),(d)). Height estimation is based on the inclined parking scenario for an actual curbstone parking scenario (fixed camera tilt β and varying camera tilt α).

and leads to inaccurate height results and hence to large errors in height estimation. Moreover, the body heights themselfs strongly influence the estimation and lead to differences in height estimation up to $4m$ even for small camera or driver tilts. In other words, a wrongly chosen height model for a real parking scenario leads to large errors in height estimation even for small camera or driver tilt.

Furthermore, real life parking scenarios usually consist of a combination of both *Curbstone Parking Scenario* and *Inclined Parking Scenario* so that a separation is not suitable for the most parking scenarios. However, if either the camera tilt or the driver tilt is very small ($\alpha, \beta < 0.05$, $\gamma, \delta < 0.05$), it is highly desirable to use one of the special height models. The advantage of using one of these parking models is the analytical determination of the driver height from the input data.

Influence of noise on body height estimation

When dealing with real life applications, input data are noisy and may include many outliers. As mentioned in Section 3.5.2, noisy input data cause an error in pose estimation and lead to errors in height estimation. To overcome this limitation, an algorithm is proposed in Section 3.4.8 to iteratively remove outliers in noisy input data and to refine the height estimation results obtained (see Figure 3.17). Outliers and noisy input data are removed during the height refinement process if their estimated height values strongly differ from an estimated mean value that has



(a) $h = 1.5m$, $\alpha = -0.1$

(b) $h = 1.5m$, $\alpha = 0.1$

(c) $h = 2.0m$, $\alpha = -0.1$

(d) $h = 2.0m$, $\alpha = 0.1$

Figure 3.35: Height error caused by a wrongly chosen height model for a short person ($1.5m$, (a),(b)) and for a tall person ($2.0m$, (c),(d)). Height estimation is based on the inclined parking scenario for a real curbstone parking scenario (fixed camera tilt α and varying camera tilt β).

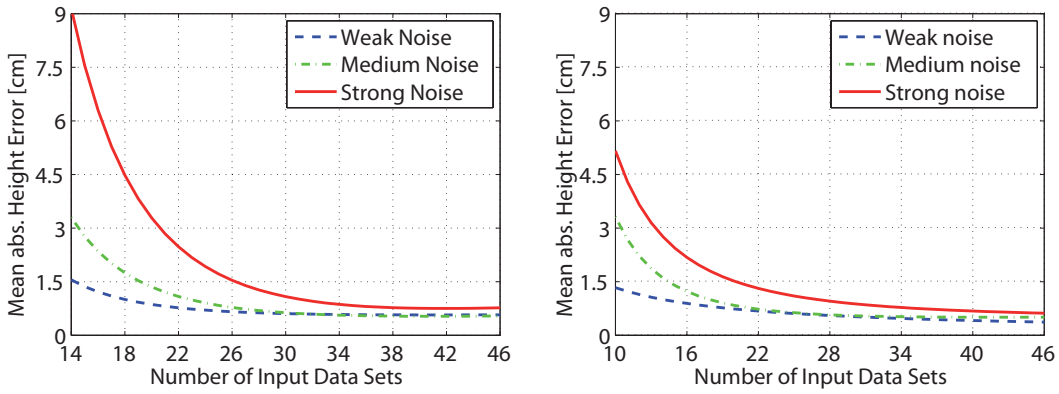
been computed across all input data. In other words, the input data sets are removed whose differences between the estimated heights and the mean height are larger than a fixed difference $d_{hi} = |h_i - h_{mean}| > \delta_{thres} \cdot h_{mean}$ – e.g. $\delta_{thres} = 20\%$.

Table 3.11 illustrates the percentage of correctly estimated body heights in a test set of input data. The percentage of correctly estimated driver body heights depends on the chosen threshold δ_{thres} and on the noise, and is determined using ≈ 500 different test configurations (foot and head points, camera and driver tilts) of approaching drivers. Each configuration consists of 42 input data sets (foot and head points obtained from the camera) and is overlaid with noise of different strength to analyze the robustness of the proposed height estimation algorithm. Weak noise in the input data led to low height errors ($\approx 4cm$), whereas medium and strong noise in the input data led to height errors of up to $12cm$. The body height for a specific test configuration was completely estimated if the camera and the driver tilt were determined and if the difference between the estimated body height and the reference body height was less than $3cm$.

Table 3.11 also demonstrates that low values for the threshold δ_{thres} lead to poor detection rates for input data overlaid with strong noise. The mean height h_{mean} that has been calculated from noisy input data at the first iteration stage may strongly differ from the real body height of an approaching driver. For too low values of δ_{thres}, only input data are considered for further iteration stages with small differences to the mean height h_{mean}. In this case, input data sets

δ_{thres}	weak	medium	strong
40 %	98.51 %	96.50 %	91.25 %
20 %	98.38 %	94.38 %	73.62 %
5 %	97.50 %	88.21 %	48.50 %

Table 3.11: Detection rate of correctly estimated body heights for different thresholds δ_{thres} and image noise.



(a) Standard minimization (b) Multi-objective goal attainment minimization

Figure 3.36: Height error over the number of input data sets for goal attainment minimization and standard minimization.

containing strong noise are assumed to be valid input data and lead to wrongly computed body heights. Additionally, threshold δ_{thres} has minor influence on the detection rates for input data sets containing weak noise. A chosen threshold $\delta_{thres} > 40\%$ leads to more robust detection results in particular for noisy input data, but the algorithm may theoretically require more iteration stages for body height estimation. However, it has been shown that only two iteration stages are required to estimate the body heights from noisy input data using the proposed height refinement process (see Figure 3.17) and a threshold $\delta_{thres} \leq 40\%$.

The quality of the extracted body height also depends on the number of input data sets used for body height estimation. Therefore, experiments have been conducted to study the influence of the number of input data sets on the height error. Figure 3.36 illustrates the height error over the number of input data sets for both the standard minimization algorithm (see Figure 3.36(a)) and for the multi-objective goal attainment algorithm (see Figure 3.36(b)) using input data sets overlaid with different noise levels (weak, medium, strong). It can be shown that both algorithms result in good height estimation for more than 38 input data sets even for very noisy input data. However, the multi-objective goal attainment problem described in Section 3.4.5 yields more precise results compared to the standard minimization algorithm and requires less input data sets for sufficient body height estimation. Moreover, the standard minimization algorithm leads to a lower detection rate for body height estimation compared to the multi-objective goal attainment algorithm (see Table 3.10)

Execution time

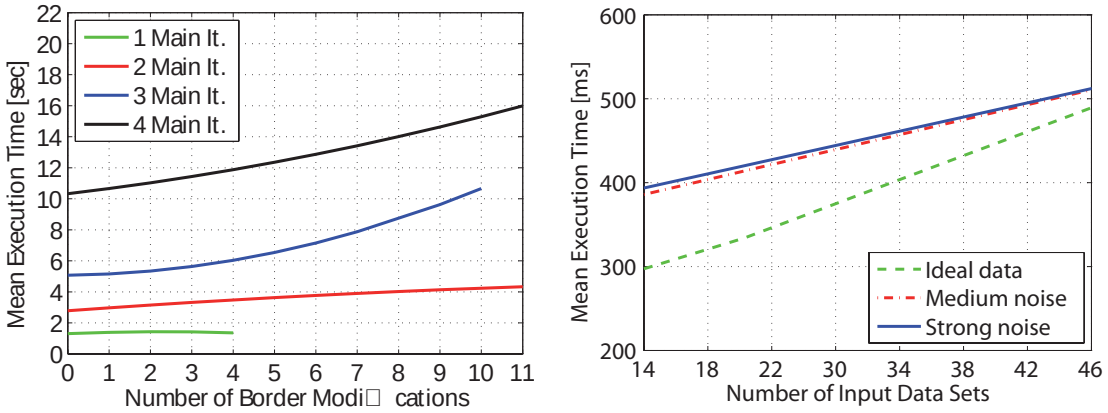
Figure 3.37 illustrates the computation time for the C-implemented height estimation algorithm using the goal attainment minimization algorithm and the standard minimization algorithm. Both algorithms are executed on an AMD Phenom 9650 2.54 GHz CPU. As mentioned in previous sections, the goal attainment algorithm yields precisely estimated body heights but requires a long time for execution.
Figure 3.37(a) presents the execution time for the goal attainment minimization algorithm using ideal input data sets and data sets overlaid medium and strong noise. Differences in execution times for the goal attainment algorithm using input data overlaid with medium and strong noise are less than $60ms$ so that corresponding curves overlap.

The execution times depend on the number of border modification stages for initial point determination and on the number of main iteration stages needed for pose estimation. However, experiments demonstrated only a slight influence of the number of input data sets on the execution time for the goal attainment algorithm. Figure 3.37(a) illustrates the execution time for height estimation over the number of border modifications for the goal attainment minimization using 42 input data sets and several main iteration stages (1-4).

By contrast, Figure 3.37(b) presents the execution time for height estimation based on the standard minimization algorithm over the number of input data sets. In that context, two iteration stages are required to refine the body heights. The execution time for the standard minimization algorithm is nearly identical for input data sets containing medium and strong noise. Contrary to height estimation that is based on the goal attainment algorithm, the execution time for height estimation using the standard minimization algorithm depends on the number of input data sets. Although the standard minimization algorithm yields less precise results than the goal attainment minimization algorithm, it can be used very well for estimating body heights of approaching drivers in automotive applications due to its fast execution times.

Table 3.12 gives an overview of the execution times for a complete height estimation process on a 2.54 GHz AMD Phenom 9650 quad-core CPU. To determine the execution times, a camera system has been chosen that has a frame-rate of 30 frames per second. The algorithm requires $2.41sec$ to compute an empty background from 60 initialization frames captured by the camera. Thereafter, the image sequence, on which background initialization is performed, contains information about the approaching drivers and are also feasible for driver extraction. Along with input data based on head and foot points extracted from 42 frames, height estimation is performed within $1.89sec$. For this reason, a total execution time of $4.3sec$ is required to estimate body heights of approaching drivers. The computation time is lower than the maximum available time of ($\approx 5sec$) for standard ingress scenarios.

Experiments are also conducted to analyze the accuracy of height estimation. Therefore, the body heights of previously measured drivers were estimated during their approaching. Fig. 3.38 illustrates a selection of previously measured drivers and their estimated heights in comparison to their real heights. It can be seen that body heights can be estimated with an accuracy of up to $3cm$ using two refinement stages for both the goal attainment and for the standard minimization algorithm. Within the domain of ergonomics, an accuracy of body height estimation of up to



(a) Multi-objective goal attainment minimization

(b) Standard minimization

Figure 3.37: Execution time required for height estimation based on the goal attainment algorithm over the number of border modifications (42 input data sets) (a). Execution time required for body height estimation using the standard minimization algorithm over the number of input data sets (b).

Task	Execution Time
BG. Initialization (60 FR.)	$\approx 2.00sec$
BG. Computation	$\approx 0.41sec$
Head/Foot Point Extraction (42 FR.)	$\approx 1.39sec$
Height estimation (fast)	$\approx 0.50sec$
Total Time	$4.30sec$

Table 3.12: Total execution time for height estimation.

$7cm$ is sufficient for individual seat pre-adjustments. However, high heeled shoes or hairstyle significantly influence body height measurements. Unfortunately, this cannot be compensated for in this thesis since only the highest and the lowest points of the walking drivers were extracted. A potential solution to overcome this limitation is to estimate properties like leg length, body size and head size, but these are potential optimizations for future work.

Finally, Figure 3.39 demonstrates the implementation and the integration of the body height estimation algorithm in a car prototype. The driver is recognized and his/her height is estimated, whereas other humans in the surroundings of the car are ignored. Three height classes are defined in order to individually pre-adjust the driver seat according to the measured body heights of approaching drivers. These classes are class 1 for short, class 2 for normally sized and class 3 for tall people. Figure 3.39 also illustrates examples of differently sized people and their categorization in one of these height classes. However, the detection range of body heights is limited to sizes between $1.5m$ for short persons and $2.0m$ for tall persons. It can also be noted that one input data set would theoretically be sufficient to estimate the body height of an approaching driver, but using at least 32 input data sets was found to be sufficient to overcome the effects of (strong) noise.

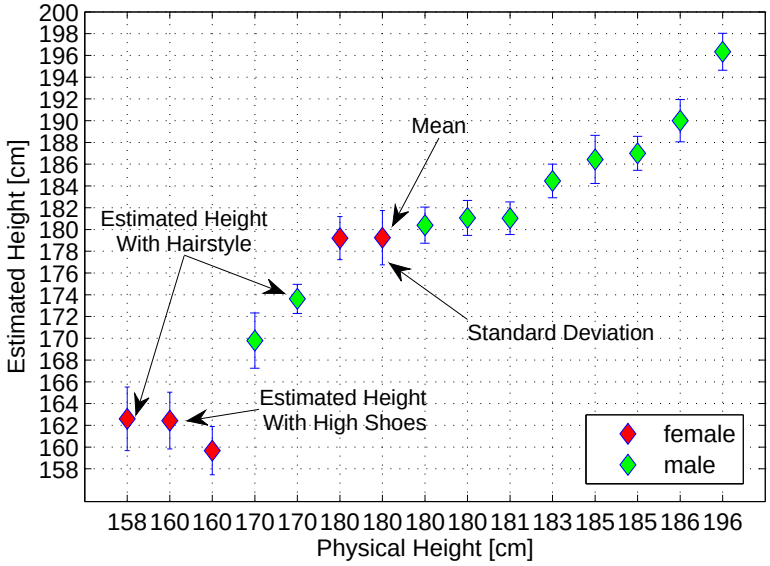


Figure 3.38: Estimated body heights and standard deviations of previously measured drivers over their real body heights. Body heights can be estimated with an accuracy of up to $2cm$.

3.6 Conclusion

This chapter proposes a novel method to estimate the absolute body height of approaching drivers in order to automatically adjust the seat position according to driver height. Driver height estimation is based on two processing stages. Driver extraction and torso determination in panoramic images captured by the camera system, tilt estimation due to potentially inclined parking cars and height estimation based on input data sets of head and foot points of approaching drivers. This chapter proposes new methods to robustly extract approaching drivers in low resolution panoramic images and to obtain absolute body height information using a single omnidirectional camera only.

A Kalman-based background model is presented to separate foreground objects from background and that can detect approaching drivers in low resolution panoramic images. A background model has been chosen instead of alternative detection methods such as optical flow since it is able to extract complete regions of approaching drivers even in low-contrast panoramic images. In particular, drivers that are far away from the car occupy fewer pixels on each video frame and are, thus, not easy to differentiate from the background. These regions, however, must be precisely determined as they serve as an input to the height estimation algorithm. In Section 3.3, a new extension to the Kalman-based background estimator is proposed to increase its robustness against shadows and illumination changes. These methods are especially targeted to eliminate shadows in gray-scaled images and to precisely separate the foreground regions with approaching drivers. From the extracted regions, foot and head points are obtained on which height estimation is based.

The key feature for enabling absolute body height estimation using a single omnidirectional camera is an estimated position and orientation (pose) of the camera relative to the ground.

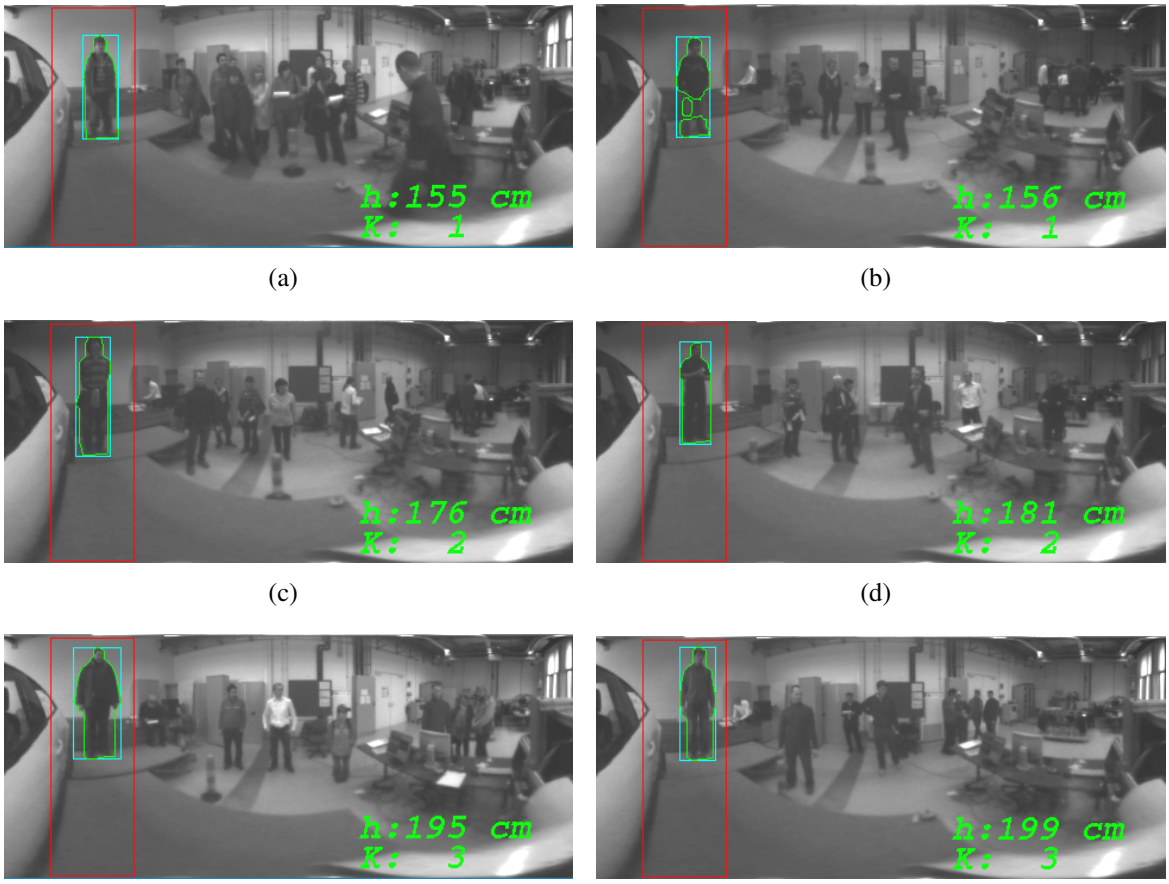


Figure 3.39: Body heights of short and tall persons that have been captured by the camera integrated in the car prototype.

However, this pose varies for each parking scenario and must, hence, be determined from image data only. Pose estimation is based on sets of extracted foot and head points obtained from the drivers as these approach. Therefore, a model-based function is proposed that can estimate the relation between the camera and the ground and that explicitly considers camera tilt caused by inclined parked cars. The camera pose is determined by minimizing this function that has a global minimum if the estimated camera pose best matches with the real camera pose (see Section 3.4). The introduction of the camera pose overcomes the scale factor problem in state-of-the-art approaches and enables height estimation with a single omnidirectional camera only. Based on sets of foot and head points and on the determined camera pose, the absolute body heights of approaching drivers are computed and serve as input for individually adjusting the seat position for better ingress.

Experimental results demonstrate a complete extraction of drivers in panoramic images even if they are far away from the car. The proposed shadow detection and illumination compensation algorithm has proved to be a powerful extension to background estimation and foreground detection in order to increase the robustness of background estimation under various illumination

conditions. Experiments also show a precise camera pose estimation based on sets of foot and head points, and an accuracy in height estimation of up to $3cm$.

The method proposed enables absolute height estimation for a wide range of parking scenarios without any knowledge of car users or geometrical information of the surroundings. This is new compared to state of the art car information systems used in the automotive domain that store height data of previously measured drivers in personal keys. Hence, this method is also feasible for rental cars and to avoid accidents caused by mistakenly chosen keys.

In this thesis, the proposed background estimator along with shadow detection is used to separate approaching drivers from background. The background estimator might also be used in an advanced driver assistance system to extract approaching cars or cyclists next to the door in order to avoid collision while opening the car door [5]. The background estimator can run in parallel and, hence, could be ported to small form factor embedded systems that blend well with a standard automotive electronic setup.

4 3D-Ambience monitoring

4.1 Introduction

Nowadays, driver assistance systems are increasingly gaining importance in high-end cars. Examples of these include Lane Departure Warning System (LDW), Adaptive Cruise Control (ACC), Forward Collision Warning (FCW) and Blind-spot detection (BSD) [23]. While there are many safety-oriented driver-assistance systems that function when the car is moving, a number of collisions with static obstacles happen while the car is stationary and one of its doors is being opened. A standard practice is to check the area close to the car before opening the door. But it is still fairly common to hit obstacles when opening the car door. An ambiance monitoring system, however, along with a car door controlling system, which is able to stop or to lock car door operations, would decrease the number of potential collisions with static objects while opening the door.

In this chapter, an image-based monitoring system is presented that detects static obstacles close to the door by generating 3D-information about the surroundings of the car. A motion-stereo-based algorithm is proposed to obtain 3D-ambiance information using a single omnidirectional camera. The ambiance information serves as an input to the collision avoidance sub-system of the car door controlling system in order to warn passengers against obstacles next to the door. This information is also feasible for controlling, stopping and locking actuated car doors in order to avoid potential collisions with static obstacles. Fig. 4.1 illustrates a high-level overview of the car door system – the smart car door system – including the image-based monitoring and the car door controlling system. In [1] and [22], a generic control system for intelligent, actuated car doors with arbitrary degrees of freedom has been presented. That paper focuses on the mechanical design and on the control of the actuated door.

The focus of this chapter is on the camera sub-system along with the generation of 3D-ambiance information. This information serves as an input to the collision avoidance planner in order to estimate the risk of collisions when opening the car door and, if necessary, to stop or lock door operations. Due to the large field of view of the omnidirectional camera, the vision system is integrated with the side-view mirror of the car and can to completely monitor the surroundings close to the door. The camera sensor, which is used for the omnidirectional camera, can be used for vision systems in the automotive domain, but it only provides images with VGA-resolution. This leads to low resolution panoramic images with an image size of only 720×204 pixels for a horizontal aperture angle of 360° and a vertical aperture angle of 101°. Panoramic images with low resolutions lead to several challenges for image processing algorithms, in particular for stereo algorithms that obtain 3D-ambiance information. Stereo algorithms use image features,

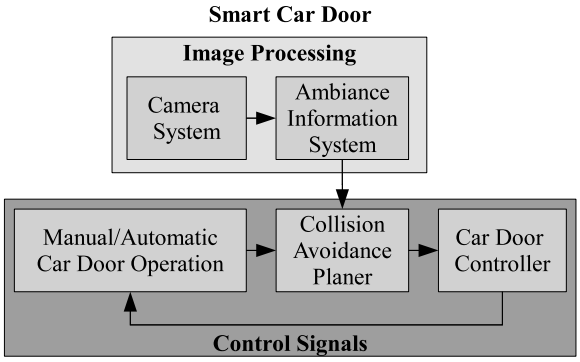


Figure 4.1: The *Smart Car Door System*. Information obtained from the ambiance information system is used to control, stop and lock car door operations in order to avoid collisions with static obstacles next to the door.

e.g. textures, to establish image correspondences for the generation of distance information. Objects such as walls, flower boxes or parked cars are common for typical parking scenarios but they only provide less or no textures. Moreover, still available textures of such objects can no longer be detected in low resolution panoramic images.

Hence, stereo algorithms must be able to obtain 3D-ambiance information from low textured and low resolution panoramic images. Besides other stereo algorithms based on dynamic programming or graph cuts, the semi-global matching algorithm proposed by Hirschmüller [106] seems to be feasible for producing disparity maps even from low textured and low resolution panoramic images. Moreover, Steingrube *et al.* [107] presented a performance evaluation of stereo algorithms for automotive applications and compared three real-time capable stereo algorithms in terms of accuracy and robustness. The semi-global matching algorithm yielded best performance of the tested algorithms and is, hence, preferred to other algorithms to produce disparity maps from panoramic images in this thesis. From the disparity maps obtained, the information system produces 3D-ambiance information of the surroundings close to the car door via triangulation. This information is represented by bounding boxes that model the locations and surfaces of potential obstacles and serves as input to the car door controlling system.

The rest of this chapter is organized as follows: Section 4.2 provides a summary of the state-of-the-art in stereo vision with omnidirectional cameras addressing applications and algorithms for robotics and for the automotive domain. Section 4.3 briefly describes the fundamentals of stereo vision and presents the principles – such as the epipolar geometry – of omnidirectional camera-based stereo vision systems. Section 4.3.2 and Section 4.3.3 describe a camera pose estimation and a feature point-based refinement stage to improve position sensor-based camera pose estimation. A rectification process is introduced in Section 4.3.4 to transform panoramic images into rectified images in order to perform a correspondence search along 1 dimension. Section 4.4 presents the computation of disparity maps based on the semi-global matching algorithm, and Section 4.5 the generation of 3D-ambiance information via triangulation. In Section 4.6, the calibration and quantization error when dealing with omnidirectional cameras are introduced and the results are presented and discussed in Section 4.7. Finally, this chapter ends with a conclusion in Section 4.8.

4.2 Related work and contributions

Applications with omnidirectional cameras are now active research topics in robotics. These cameras provide a very large field of view compared to perspective cameras and are, hence, useful for many applications such as navigation, path-planning, obstacle detection and ego-motion estimation [108]. For example, mobile robots might use omnidirectional cameras to generate maps of unknown environments in order to accomplish autonomous tasks. In the 90s, researchers realized that omnidirectional cameras are particularly suitable for improving ego-motion estimation [109]. It has been shown that ego-motion estimation algorithms cannot distinguish small translations parallel to an image plane from small rotations in images captured b perspective cameras. Omnidirectional cameras overcome this limitation by obtaining image correspondences from everywhere so that ego-motion estimation can be performed independently of the direction of motion. Later, ego-motion estimation using omnidirectional cameras was refined by Gluckman *et al.* [110] and others so that structure-from-motion for panoramic cameras became an active area of research.

4.2.1 Ego-motion and structure-from-motion from omnidirectional cameras

Svoboda *et al.* [109] were the first who presented the fundamental theory of epipolar geometry between a pair of central catadioptric cameras. They decomposed the mathematical model of a central panoramic camera into two central projections to derive the epipolar geometry. Kang [111] and Chang *et al.* [112] introduced discrete structure-from-motion estimators and applied to omnidirectional cameras the known structure-from-motion methods developed for perspective cameras. To achieve this, they used calibrated omnidirectional cameras and the consistency of pairwise point features across image sequences. Geyer *et al.* [113] extended and solved the structure-from-motion problem for uncalibrated omnidirectional cameras assuming unknown intrinsic camera parameters. To enable the estimation of the fundamental matrix, they introduced a new representation for point correspondences and lines in panoramic images. Simultaneously, Fitzgibbon [114] considered the radial distortion parameters of omnidirectional cameras and introduced a self-calibration method to estimate the distortion by means of several image views using a non-linear, quadratic eigenvalue solver.

At the same time, Bunschoten *et al.* [115, 116] proposed a method to estimate and to refine the 2D translation and rotation between two subsequent poses of a mobile robot from panoramic images allowing visual odometry. They determined the rotation and the length of the translation vector by projecting panoramic images on a plane parallel to the ground. Furthermore, they extended their work with an efficient multi-baseline stereo algorithm to suppress noise and to detect outliers in 3D-data [117]. The proposed algorithm was completed later with a recovering algorithm for 3D-scene structures by means of cylindric panoramic images [118]. Later, Micusik *et al.* [119] extended Fitzgibbon's approach to obtain the camera model of an omnidirectional camera from epipolar geometry by solving a polynomial eigenvalue problem. They continued their line of research in [120] and provided a solution to the uncalibrated

structure-from-motion-problem. However, their algorithm requires a known image center both for fish-eye lenses and for omnidirectional cameras. Scaramuzza *et al.* [26, 30] extended the algorithm proposed by Micusik so that no prior knowledge in terms of known image center etc. is required to calibrate omnidirectional cameras.

A lot of work has been done to improve existing structure-from-motion algorithms, to perform auto-calibration and to reconstruct a 3D-scene from point correspondences in panoramic images using only video input data [121, 122, 123]. Some approaches use additional odometry data, inertial sensors [124] or a laser range meter for applications with perspective cameras [125] in order to increase the robustness of scene reconstruction. Makadia *et al.* [122] mentioned that general structure-from-motion methods should be usable for any type of camera motion. Therefore, they proposed constrained ego-motions based on a generalized Hough transformation for spherical panoramic cameras. This transformation processes all potential combinations of corresponding pairwise features in consecutive images instead of selecting the best matches. Makadia *et al.* [122] also show that constrained camera motions greatly simplify vision tasks such as ego-motion estimation, mobile robot localization or structure-from-motion.

Very recent research addressing structure-from-motion has been presented by Scaramuzza and Pollefeys in [126]. That work proposes a solution to the the unknown scale factor problem for applications with a single (omnidirectional) camera that is attached to a vehicle. Therefore, they developed a method to automatically compute the absolute scale factor by using an offset of the camera to the vehicle's center of motion and by introducing non-holonomic constraints for wheeled vehicles. The absolute scale factor can then be determined accurately when the vehicle turns. Last, Kawanishi *et al.* [108] proposed a method to model and to reconstruct a 3D-environment based on feature point correspondences. In their approach, feature point correspondences are captured by a single omnidirectional camera placed on top of a moving robot.

4.2.2 Stereo vision in the automotive domain

Stereo algorithms, structure-from-motion and omnidirectional cameras left the field of robotics and the scientific labs to be used for automotive applications. Stereo algorithms in the domain of automotive system engineering are used to detect obstacles [127, 128] and pedestrians [129]. They are also essential for intersection assistance systems [130], for path planning tasks for autonomous driving in off-road environments [131] and for driver assistance systems [132]. A system that uses stereo algorithms to detect obstacles in a pair of images is presented by Bertozzi *et al.* [127]. Their algorithm extracts obstacles in real-time by using a parallel computing architecture and is also highly robust against shadows, illumination changes and different road textures. Krotosky *et al.* [129] use color, infrared, and multi-modal stereo approaches to detect pedestrian in front of a moving vehicle.

In the automotive domain, most of the obstacle and pedestrian detection algorithms are based on stereo vision and must run in real-time. For this reason, Franke *et al.* [133] present a precise correlation-based stereo vision approach for real-time interpretation of traffic scenes. That algorithm has been extended in [134] to quickly detect obstacles and pedestrians and to

interpret traffic situations for an urban traffic assistance system. Besides real-time aspects, many applications require accurate dense disparity maps to enable accurate scene estimation in front of the car [135]. For this reason, Steingrube *et al.* [107] presented a performance evaluation of three real-time capable dense stereo algorithms for automotive applications. The semi-global matching algorithm [136] yielded the best performance in terms of accuracy and robustness. Additionally, more performance evaluations of stereo algorithms feasible for the automotive domain were done by Klette *et al.* in [137].

4.2.3 Panoramic stereo vision in the automotive domain

Many applications in the automotive domain have been presented that use pairs of stereo images captured by perspective cameras. Gehrig *et al.* [138] mentioned that 3D-perception of the vehicle's surroundings is crucial for automotive applications. Therefore, large field of view cameras such as fish-eye and omnidirectional cameras are highly desirable for many driver assistance and safety systems. They proposed the use of fish-eye cameras for driver assistance systems and developed a low power hardware architecture that performs semi-global matching on an embedded system [139].

Todays trucks are equipped with omnidirectional cameras. The closest surrounding area of a truck cannot be seen directly by a truck driver. For this reason, omnidirectional cameras are used as a maneuvering aid [140] and provide a bird-eye view of the surroundings next to the vehicle (blind-spot detection, [141]). Furthermore, images captured by omnidirectional cameras can also serve as an input for parking assistance systems. These systems compute potential motion paths for parking and, hence, increase the safety when driving large vehicles [24]. For an automobile, Ghandi *et al.* presented several studies and experiments addressing the "... awareness of what surrounds a vehicle that directly affects the safe driving and maneuvering of an automobile ... "Ghandi *et al.* [142, 143]. They introduced the concept of *dynamic panoramic surround maps* (DPS) obtained from images of omnidirectional cameras integrated on each side of the vehicle to monitor the close surroundings when driving. Stereo algorithms, motion analysis based on optical flow and ego-motion estimation algorithms are then used to interpret the surroundings and to build DPS maps. Similarly, Tardif *et al.* [123] presented an algorithm for visual simultaneous localization and mapping (SLAM) in urban environments with an omnidirectional camera integrated within a car. They estimated the motion trajectory of the camera without any motion model by using the epipolar constraint for omnidirectional cameras and a 3D-map instead.

However, little work has been done on stereo vision with omnidirectional cameras in the domain of automotive engineering for smart car doors. Omnidirectional cameras are commonly used to monitor the ambiance in front of or next to the car while driving [143] or as a parking assistance aid [25]. However, no application was presented that uses omnidirectional cameras to monitor the ambiance next to the car in order to avoid collisions with obstacles when opening a car door. In this section, a new application is proposed which uses motion stereo with a single omnidirectional camera in order to detect static obstacles next to the car door. Figure 4.2(a) illustrates the omnidirectional camera integrated within each side-view mirror of the car. In this

(a) (b)

Figure 4.2: Omnidirectional camera integrated with the side-view mirror (a) and the mechanical device to provide camera motion [12](b).

application, a mechanical device [12] is attached to the side-view mirror to vertically position the camera and, hence, to obtain 3D-information from a stereoscopic device based on motion stereo (see Figure 4.2(b)). For following applications, the camera movement could be provided by a fold-in and a fold-out movement of the side-view mirror to dispense with the prototypical mechanical device. Disparity maps could then be obtained from panoramic images captured at several camera positions. From the disparity maps, 3D-ambiance information is produced using triangulation and serves as an input to the control unit of the car door to avoid collisions with potential obstacles by stopping or locking door operations.

For the car door controlling system, the sizes and locations of potential obstacles are important, but not in so much details as required for image-based scene representations [125] for telepresence applications [144]. For this reason, 3D-ambiance information of the surroundings next to the car is provided by bounding boxes (point clouds) representing the shapes, sizes and locations of potential obstacles next to the door. To obtain ambiance information about this application, the following assumptions and constraints are important:

- All obstacles in an arbitrary parking scenario are assumed to be static to enable the generation of 3D-data based on motion stereo.
- All obstacles should not be closer than $50cm$ to the car to provide sufficient space for door openings.
- Obstacles within the required space for door openings are of interest and must all be detected. Therefore, 3D-ambiance information represented by bounding boxes must only be generated for obstacles within this area.
- The area of interest is located close to the car door and its size is limited. Therefore, a small baseline length of $\approx 3.5cm$ could be sufficient to obtain 3D-ambiance information from pairs of panoramic images.

Figure 4.3 illustrates the block diagram for obtaining 3D-ambiance information from the surroundings next to the car. The mechanical device vertically positions the omnidirectional cam-

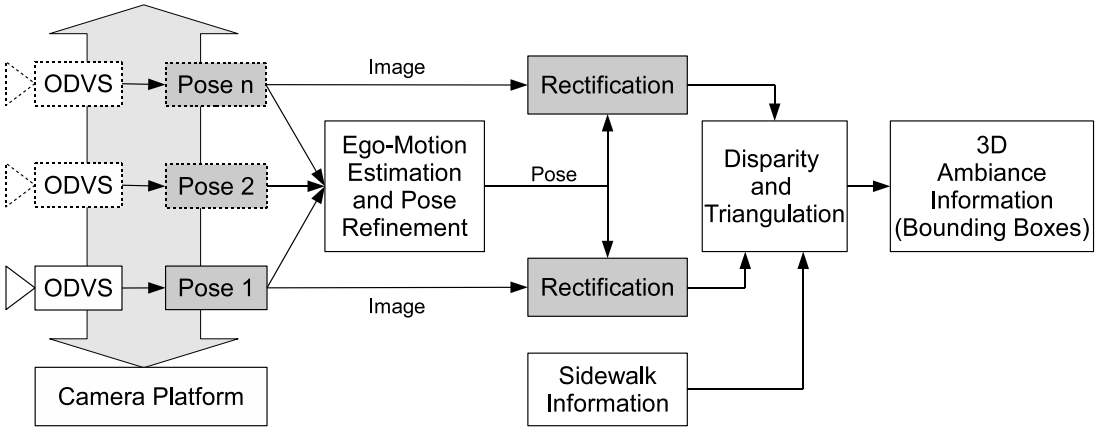


Figure 4.3: Block diagram of the proposed algorithm to obtain 3D ambiance information based on motion stereo.

era in at least two poses to capture images from these positions hereby providing a stereo setup. The coordinate system of the camera located at the lowest position provided by the mechanical device can easily be calibrated to the car coordinate system: Therefore, this position is denoted as the reference position on which pose estimation and refinement is based. The mechanical device is equipped with a position sensor to determine the (pure) translation of the camera. However, the coordinates of the reference camera system and the coordinates of the camera system at an arbitrary pose N are related to each other by a translation and a rotation due to inaccuracies in the mechanical device. The rotation, however, cannot be measured by the translation sensor and must be recovered from image data only. The resulting relation between the coordinate systems at different poses can then be used to refine the camera poses and to rectify panoramic images to obtain a *parallel panoramic configuration*. This configuration allows for a correspondence search along one dimension and to facilitate disparity map computations for pairs of panoramic images.

Triangulation is used to provide 3D-ambiance information about the objects located in the required space for door opening. This information is represented by bounding boxes and is used by the control unit to compute opening paths for collision free door operations. Due to the poor resolution of panoramic images used in this application and due to less textured regions, however, the bounding boxes may be very noisy and could contain many outliers. Additional information within panoramic images can be obtained from objects that are in contact with the floor. Clearly, the edges between such objects and the floor are suitable for refining the locations of bounding boxes by incorporating them into the 3D-data. In this manner, noise and outliers can be suppressed sufficiently.

4.3 Stereo with omnidirectional cameras

A traditional approach to obtaining 3D-information of a real scene is stereo vision. For this purpose, stereo algorithms are developed to recover depth information from at least two or more images captured by a n-camera stereo configuration or by a single camera that has been

moved to different positions. The problem of positioning a single camera to different places and to recover 3D-information is known as the *motion stereo problem*. In general, the underlying process of nearly all stereo algorithms is the establishment of point correspondences in pairs of panoramic images on which computation of depth information is based. These correspondences are mostly projections of the same physical scene points in one or more stereo images. With the data obtained, triangulation is often used to reconstruct a 3D-scene from the point correspondences and to determine the 3D-coordinates of these points relative to one reference camera coordinate system. In this chapter, the fundamentals of stereo vision – the epipolar geometry – for perspective and omnidirectional cameras are presented and explained in Section 4.3.1. Section 4.3.2 describes the estimation and the relation of the camera poses to each other. Pose estimation is based on image correspondences (feature points, see Section 4.3.3) and is required to overcome inaccuracies in sensorially estimated camera positions and to rectify panoramic images (see Section 4.3.4) thus enabling correspondence search along one dimension.

4.3.1 Epipolar geometry

In stereo vision, epipolar geometry is essential for determining depth information. Epipolar geometry is a mathematical model that describes a geometrical relation between (image) correspondences in pairs of images captured by a stereo device consisting of perspective or omnidirectional cameras. Such a stereo device must view a 3D-scene from at least two distinct positions, so that each image contains the same scene information from a different view point. In each stereo image, 3D-points of the scene are projected onto each image plane as 2D-points. The epipolar geometry describes then a geometrical relation between corresponding 2D-points as seen by the cameras in a pair of images.

The principles of epipolar geometry were first been studied by Hauck in 1883 and by von Sanden in 1908 [145]. However, epipolar geometry became more popular when computer-aided analyzing of digital images was feasible. Epipolar geometry enables correspondence search along one dimension and is primarily used to obtain 3D information from 2D images. It strongly depends on the relative camera poses and on the internal camera parameters for both perspective and omnidirectional cameras. Figure 4.4(a) illustrates the epipolar geometry for a pair of perspective cameras and Figure 4.4(b) shows the epipolar geometry for a pair of omnidirectional cameras using spherical projection.

Epipolar geometry can be derived as follows [146]: Let I_0 and I_1 be images of a 3D-scene point $\mathbf{M}$ that has been captured by a camera at two different poses C_0 and C_1. The coordinates $\mathbf{M}_i = [X, Y, Z]_i^T$ are the world coordinates of scene point $\mathbf{M}$ and $\mathbf{m}_i = [x, y, z]_i^T$ the coordinates of the projection on the image plane at the i-th pose. The coordinate system $\mathbf{C}_0$ is denoted as the reference coordinate system and, hence, I_0 as the corresponding reference image (see Figure 4.4). Furthermore, the two camera coordinate systems at two different poses C_0 and C_1 can be transformed into each other by a translation $\mathbf{t}$ and a rotation $\mathbf{R}$. In this manner, each world point $\mathbf{M}$ is represented in both camera systems following Eq. 4.1.

$$\mathbf{M}_1 = \mathbf{R}\mathbf{M}_0 + \mathbf{t} \tag{4.1}$$

In fact, only the projections $\mathbf{m}_i$ of a point $\mathbf{M}$ up to an unknown scale factor s_i are available. For this reason, the representation of a world point $\mathbf{M}$ presented in Eq. 4.1 must be rewritten as follows:

$$s_1\mathbf{m}_1 = s_0\mathbf{R}\mathbf{m}_0 + \mathbf{t} \tag{4.2}$$

Moreover, the straight lines spanned by $\mathbf{m}_1$ and $\mathbf{m}_0$ meet in the 3D-scene point $\mathbf{M}$ whereas vectors $\mathbf{t}$, $\mathbf{m}_0$ and $\mathbf{m}_1$ are assumed to be co-planar. The plane spanned by these vectors is called the *epipolar plane* and can be defined by its normal $\mathbf{n}_0 = \mathbf{t} \times \mathbf{m}_0$. The corresponding normal $\mathbf{n}_1$ represented by the camera system $\mathbf{C}_1$ can be derived based on Eq. 4.3:

$$n_1 = \mathbf{R}(\mathbf{t} \times \mathbf{m_0}) \tag{4.3}$$

Following the co-planarity condition, the epipolar constraint can now be established and is expressed in Eq. 4.4:

$$\mathbf{n}_1^T \cdot \mathbf{m}_1 = 0 \tag{4.4}$$

Figure 4.4(a) illustrates the epipolar geometry for conventional perspective cameras. An *epipolar plane* is spanned by the three vectors $\mathbf{t}$, $\mathbf{m}_0$ and $\mathbf{m}_1$, and the intersections of each epipolar plane with the sensor plane forms the *epipolar lines* for any 3D-scene point. The epipolar lines and epipolar planes can be obtained for all 3D-scene points that are visible in all stereo images. For a stereo setup consisting of perspective cameras, all epipolar lines meet in a single image point that is called the *epipole*. Depending on the chosen stereo configuration, the epipole may be located within or beyond the sensor plane (see Fig. 4.4(a)) or may be located in infinity for *parallel camera configurations*. The epipole is also the point on the sensor plane on which a line – called the *baseline* – joining the effective pinholes of the cameras or of one camera placed at different poses intersects the image plane. The parallel camera configuration describes a special stereo setup designed to drastically facilitate matching of point correspondences in stereo images. In this configuration, all epipolar lines are parallel and relate to the image rows. Therefore, matching can be performed along single rows and depth information can be directly computed from the disparities obtained [147].

Figure 4.4(b) displays the epipolar geometry for spherical panoramic images. Bunschoten [148] mentioned in that panoramic images – that are transformed from original images captured by an omnidirectional camera – can be treated as directly captured by a virtual panoramic (omnidirectional) camera. He uses cylindrical panoramic images for n-occular stereo vision tasks. However, the use of spherical panoramic images obtained by omnidirectional cameras has some advantages over cylindrical panoramic images and is preferred in this thesis. The virtual panoramic camera is specified by constructing a target projection area whose center is placed in the projection center of the mirror (see Section 2.4). Rectified panoramic images taken by omnidirectional cameras can then be processed as if truly captured by a virtual panoramic camera and the epipolar constraint for omnidirectional cameras can be derived based on the epipolar constraint of perspective cameras. A very good description about epipolar geometry and epipolar curves for cylindric panoramic images has been presented by Bunschoten [148] and is summarized below.

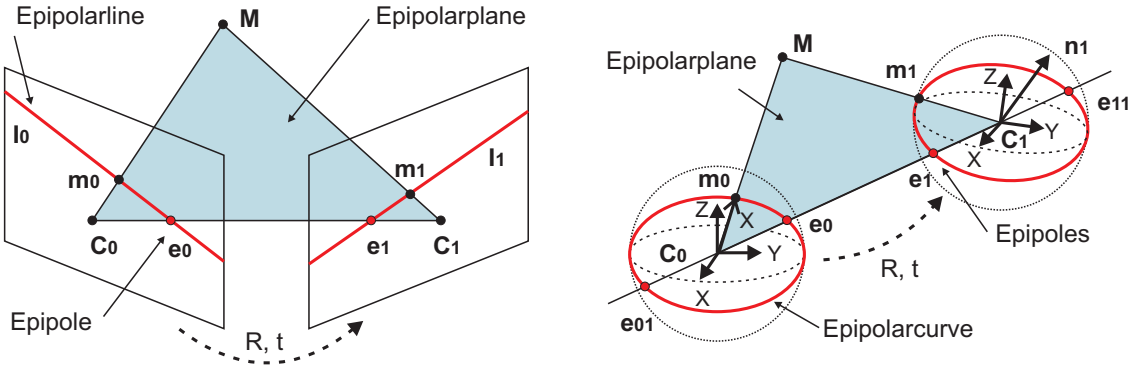


Figure 4.4: Stereo setup that consists of perspective cameras (a) and stereo setup with omnidirectional cameras whose image planes are represented by spheres (b).

Let $\mathbf{M}$ be the coordinates of a 3D-scene point and let $\mathbf{m}_0$ and $\mathbf{m}_1$ be the corresponding image coordinates in the spherical camera coordinate systems $\mathbf{C}_0$ and $\mathbf{C}_1$. Similar to stereo configurations consisting of perspective cameras, the coordinate system $\mathbf{C}_0$ is denoted as the reference coordinate system and I_0 the corresponding panoramic reference image. Furthermore, the two camera coordinate systems at two different poses C_0 and C_1 can be transformed into each other by a translation $\mathbf{t}$ and a rotation $\mathbf{R}$ (see Figure 4.4(b)).

In contrast to stereo setups that consist of perspective cameras, the baseline joining the effective viewpoints of the omnidirectional cameras intersects each image plane two times (see Figure 4.4(b)). Therefore, two epipoles exist in panoramic images captured by a panoramic stereo setup. For this configuration, the epipolar plane is spanned by the vectors $\mathbf{t}$, $\mathbf{m}_0$ and $\mathbf{m}_1$. The intersection line of the panoramic projection area with the epipolar plane forms the *epipolar curve* for each 3D-scene point. For stereo setups with horizontally arranged panoramic cameras, panoramic images do not preserve straight epipolar lines and epipolar curves are obtained instead. The resulting epipolar curves start in one epipole and end in the other one. Figure 4.5(a) illustrates epipolar curves in an original image captured by an omnidirectional camera and in a panoramic image for a horizontal setup.

Bunschoten uses the epipolar constraint (see Eq. 4.4) to derive a parameterization for the epipolar curves for cylindric panoramic images [148]. Therefore, he expressed $\mathbf{m}_1$ as $\mathbf{m}_1 = [cos(\phi_1), sin(\phi_1), z_1]^T$ using cylindric coordinates and expanded Eq. 4.4 as follows:

$$0 = \mathbf{n}_1^T \cdot \mathbf{m}_1 = [n_x, n_y, n_z][cos(\phi_1), sin(\phi_1), z_1]^T \tag{4.5}$$

The elevation z_1 of the epipolar curve can be expressed as a function of ϕ_1 after reorganizing Eq. 4.5 [148]

$$z_1(\phi_1) = -\frac{n_x cos(\phi_1) + n_y sin(\phi_1)}{n_z} \tag{4.6}$$

Eq. 4.4 is also suitable for deriving a parametrization of the epipolar curves for spherical panoramic images using $\mathbf{m}_1 = [sin(\theta_1)cos(\phi_1), sin(\theta_1)sin(\phi_1), cos(\theta_1)]^T$ in spherical coor-

dinate systems.

$$0 = \mathbf{n}_1^T \cdot \mathbf{m}_1 = [n_x, n_y, n_z][sin(\theta_1)cos(\phi_1), sin(\theta_1)sin(\phi_1), cos(\theta_1)]^T \tag{4.7}$$

The elevation angle θ_1 is expressed as a function that depends on the rotation angle ϕ_1 (see Eq. 4.8)

$$\theta_1(\phi_1) = \arctan\left(-\frac{n_z}{n_x cos(\phi_1) + n_y sin(\phi_1)}\right) \tag{4.8}$$

For deriving epipolar curves, unit radii are assumed for the parametrization of the epipolar curves for both cylindrical and spherical panoramic images. However, a special case exits for stereo setups with omnidirectional cameras. This configuration is similar to the parallel stereo camera configuration for perspective cameras where all epipolar lines are in parallel to each other. In this case, the omnidirectional cameras are perfectly aligned vertically so that the two coordinate systems C_0 and C_1 can be transformed into each other by a translation only. This configuration is obtained by a baseline direction being equal to the direction of the z-axes for the coordinate systems of the virtual panoramic cameras.

With this configuration, panoramic images can be generated whose orientations of the coordinate systems are coincident for each pose $\mathbf{C}_i$ and in which all epipolar lines are in parallel. The epipolar lines relate to the image columns and, hence, matching of point correspondences in pairs of panoramic images can be drastically simplified. For this reason, depth information can be directly determined from disparities obtained in panoramic images. Figure 4.5(b) illustrates epipolar lines in an original image and in a panoramic image for a perfectly aligned vertical stereo setup. In this thesis, the stereoscopic configuration is also called *parallel panoramic configuration*.

In this application, motion stereo is used to obtain 3D ambiance information next to the car door. A mechanical device is attached to the side-view mirror and provides a vertical movement of the omnidirectional camera along the z-axis of its coordinate system. The projection center of the camera is referred to as the origin of the coordinate systems $\mathbf{C}_i$ for spherical panoramic images at the poses i. The coordinate system of the lower pose $\mathbf{C}_0$ is referred to as world coordinate system $\mathbf{W}$ (the reference coordinate system) whereas the coordinate system of the upper position is referred to as $\mathbf{C}_1$. Panoramic images that have been captured from different vertical positions correspond to two images that have been captured from a parallel panoramic configuration in order to perform correspondence search along single columns. However, an ideally vertical camera motion cannot be realized due to clearances in the mechanical device. This leads the translation in the direction of the baseline to be superimposed by additional translation components along other directions. Moreover, the translation $\mathbf{t}$ is also superimposed by an additional rotation $\mathbf{R}$ required to align the two coordinate systems $\mathbf{C}_0$ and $\mathbf{C}_1$.

Figure 4.6(a) illustrates a non-parallel panoramic configuration for a pair of spherical panoramic images positioned at different poses i. In this case, the coordinate systems of the panoramic images are coincident with the coordinate systems of the poses $\mathbf{C}_i$. The baseline joining the coordinate systems is also called the *orientation axis for rectification* and is a prerequisite for rectifying panoramic images in order to obtain a parallel panoramic configuration. In other words, pairs of panoramic images can be generated in order to achieve a parallel panoramic

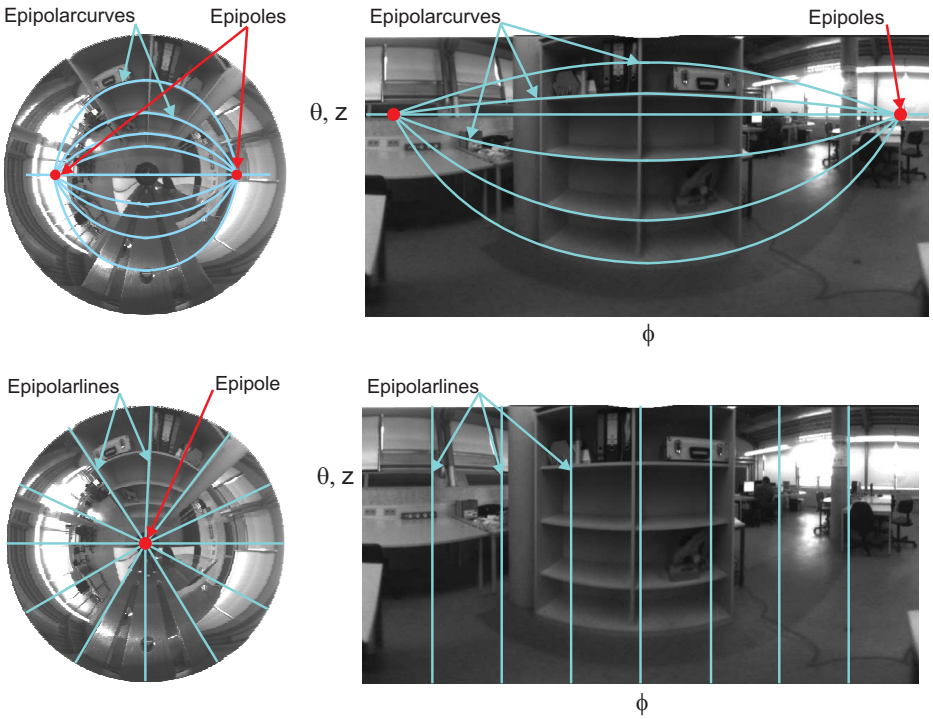


Figure 4.5: Epipolar curves and epipolar lines for a horizontal stereo setup (top) and for a vertical stereo setup (parallel panoramic configuration) (bottom).

configuration if the rotation $\mathbf{R}$ and the translation $\mathbf{t}$ between the coordinate systems $\mathbf{C}_i$ are known. Original images are projected into panoramic images that are based on rectified coordinate systems $\mathbf{C}_{ri}$ meaning that their orientations are coincident to each other and the relation between the coordinate systems $\mathbf{C}_{r0}$ and $\mathbf{C}_{r1}$ is represented by a pure translation. This way, the epipolar lines are in parallel in each panoramic image and matching of point correspondences can be drastically facilitated. Figure 4.6(b) illustrates the parallel panoramic configuration with coincident coordinate systems $\mathbf{C}_{ri}$ for a pair of spherical panoramic images captured from two poses i. In this setup, the direction of the baseline is equal to the direction of the z-axis for both camera coordinate systems.

4.3.2 Determining and refining camera poses

Known camera poses of a stereo vision setup are a prerequisite for obtaining information of a 3D-scene in a pair of 2D-images via triangulation. Although the mechanical device provides camera motions relating to a pure translation along in direction, clearances in the mechanical device lead to small additional translations and additional rotations.

The camera platform is equipped with a position sensor to precisely determine the translation in the direction of the z-axis, but additional translations and rotations cannot be measured. This leads to inaccuracies in camera pose estimation and, hence, to camera poses that do not fulfill the epipolar constraint for parallel panoramic configurations. Consequently, the actual camera poses must be estimated to perform image rectifications of panoramic images in order to obtain a parallel panoramic configuration. The relation between the camera poses – clearly the

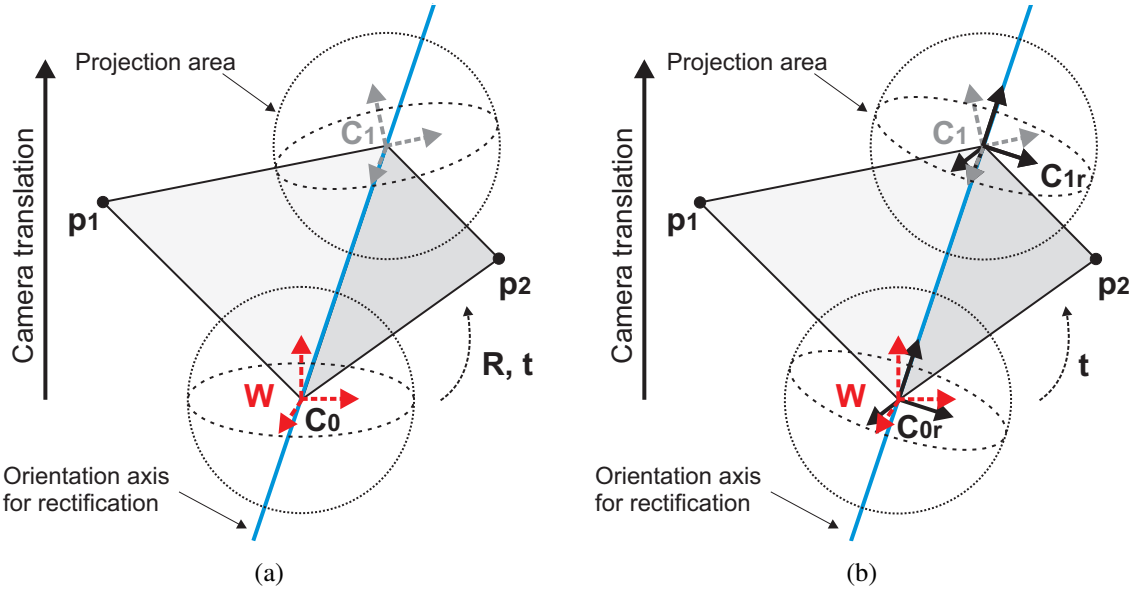


Figure 4.6: Non-parallel panoramic configuration (a) and parallel panoramic configuration (b) with coincident coordinate systems $\mathbf{C}_{ri}$.

rotation $\mathbf{R}$ and translation $\mathbf{t}$ which relate an image I_n to a reference image I_0 – can be estimated via the epipolar geometry and a sufficient number of correspondences. Therefore, Faugeras [147], Hartley [149] and others proposed an algorithm to estimate the epipolar geometry and the essential matrix with N numbers of correspondences tracked in a set of (panoramic) images $I_0..I_n$. This algoritm is briefly presented below. Any true point correspondence must fulfill the epipolar constraint following Eq. 4.9.

$$\mathbf{m}_n = \mathbf{R}\mathbf{m}_0 + \mathbf{t} \tag{4.9}$$

Eq. 4.9 can be rewritten by taking the vector product with t followed by a scalar product with $\mathbf{m}_n^T$. Eq. 4.10 illustrates the resulting equation

$$\mathbf{m}_n^T \mathbf{t} \times \mathbf{R}\mathbf{m}_0 = 0 \tag{4.10}$$

where $\mathbf{R}$ denotes the rotation matrix and $\mathbf{t}$ the translation vector. Eq. 4.10 can then be rewritten in matrix form as follows:

$$\mathbf{m}_n^T \mathbf{S}\mathbf{R}\mathbf{m}_0 = \mathbf{m}_n^T \mathbf{E}\mathbf{m}_0 = 0 \tag{4.11}$$

The matrix $\mathbf{E} = \mathbf{SR}$ obtained is called the *essential matrix* [147] where $\mathbf{S}$ is the (3×3) skew symmetric matrix.

$$\mathbf{S} = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ t_y & t_x & 0 \end{pmatrix} \tag{4.12}$$

However, image correspondences $\mathbf{m}_{0,i} \leftrightarrow \mathbf{m}_{n,i}$ obtained from real images are noisy and the epipolar constraint is only approximately satisfied. For this reason, Eq. 4.11 must be rewritten

as follows for any point correspondence i in a pair of panoramic images [149].

$$\mathbf{m}_{n,i}^{T}\mathbf{E}\mathbf{m}_{0,i} = [x_n\ y_n\ z_n]_i \begin{pmatrix} e_0 & e_1 & e_2 \\ e_3 & e_4 & e_5 \\ e_6 & e_7 & e_8 \end{pmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}_i \approx 0 \tag{4.13}$$

Next, Eq. 4.14 can be reorganized into the following matrix vector form by stacking the entries of the essential matrix into a vector $\mathbf{e} = [e_0\ ..e_8]^T$.

$$[x_k x_0\ x_k y_0\ x_k y_0\ y_k x_0\ y_k y_0\ y_k y_0\ z_k x_0\ z_k y_0\ z_k y_0]_i \cdot \mathbf{e} = \mathbf{d}_i \cdot \mathbf{e} \approx 0 \tag{4.14}$$

The advantage of Eq. 4.14 is in providing a constraint of the essential matrix $\mathbf{E}$ for each correspondence i and in setting up a system of linear equations (see Eq. 4.15) that can be solved for all correspondences simultaneously.

$$\begin{bmatrix} \mathbf{d}_0 \\ \vdots \\ \mathbf{d}_N \end{bmatrix} \cdot \mathbf{e} = \mathbf{D}\mathbf{e} \approx 0 \tag{4.15}$$

Variable $\mathbf{D}$ is a $(N \times 9)$ design matrix containing N vectors $\mathbf{d}_i$ of point correspondences. A solution for the essential matrix $\mathbf{E}$ can be found by linearly solving Eq. 4.15

$$min \parallel \mathbf{D}\mathbf{e} \parallel^{2} \text{ with } \parallel e \parallel = 1. \tag{4.16}$$

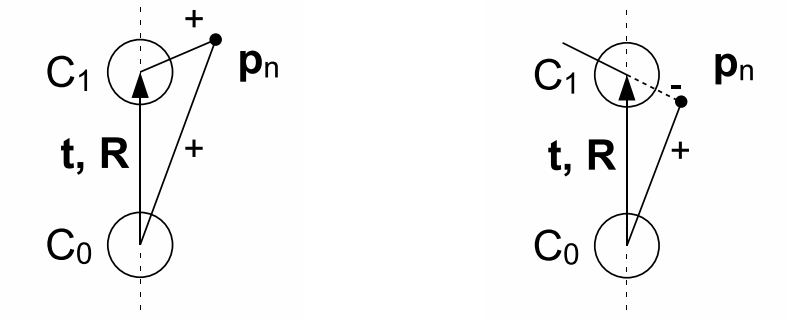
where the constraint $\|e\| = 1$ is introduced to fix the scale of the essential matrix $\mathbf{E}$ and to remove one degree of freedom. In this manner, a solution for e can be theoretically obtained for eight correspondences by solving the moment matrix $\mathbf{M} = \mathbf{D}^T\mathbf{D}$ using singular value decomposition. The eigenvector of the moment matrix $\mathbf{M}$ associated with the smallest eigenvalue relates to the minimum solution that fulfills Eq. 4.16. This algorithm described above has been proposed by Hartley and is known as the *8-point algorithm* [149].

However, the essential matrix E has two equal eigenvalues and rank two [147]. Bunschoten [148] mentioned that the rank condition is not considered in the 8-point algorithm. Due to noisy correspondences, the recovered essential matrix E has rank three. Instead, Bunschoten [148] introduced a *nearest true essential matrix* $\mathbf{E}$ derived from an estimated essential matrix $\tilde{\mathbf{E}}$ using singular value decomposition, where $\tilde{\mathbf{E}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ and $\mathbf{\Sigma} = diag(\sigma_1, \sigma_2, \sigma_3)$. He proposed a method to estimate the nearest true essential matrix by means of $\mathbf{E} = \mathbf{U}\mathbf{\Sigma}'\mathbf{V}^T$, where $\mathbf{\Sigma}' = diag((\sigma_1 + \sigma_2)/2, (\sigma_1 + \sigma_2)/2, 0)$.

Once the essential matrix has been determined, the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ are computed. To achieve this, Hartley proposed a method to determine the translation matrix $\mathbf{S}$ and the rotation matrix $\mathbf{R}$ from the singular value decomposition $\mathbf{E} = \mathbf{U}\mathbf{\Sigma}'\mathbf{V}^T$ following Eq. 4.17 and Eq. 4.18 [150].

$$\mathbf{R} = \mathbf{U}\mathbf{Y}\mathbf{V}^T \quad \text{or} \quad \mathbf{R} = \mathbf{U}\mathbf{Y}^T\mathbf{V}^T \tag{4.17}$$

$$\mathbf{S} = \mathbf{U}\mathbf{Z}\mathbf{U}^T \quad \text{or} \quad \mathbf{S} = \mathbf{U}\mathbf{Z}^T\mathbf{U}^T \tag{4.18}$$



(a) Correct Rotation $\mathbf{R}$ (b) Wrong Rotation $\mathbf{R}$

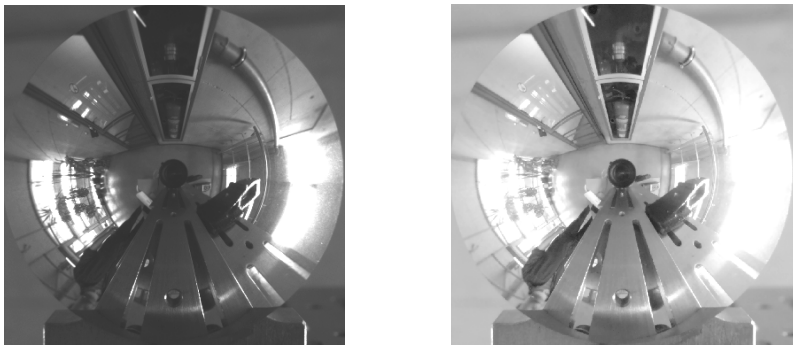
Figure 4.7: Solutions for the two rotation matrices: Correct rotation $\mathbf{R}$ with the two positive depths (a) and the wrong rotation $\mathbf{R}$ with one negative depth to the 3D-point (b).

where

$$\mathbf{Y} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{Z} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.19}$$

The essential matrix $\mathbf{E}$ is a projective value and is, hence, defined up to an unknown scale factor. In other words, the rotation matrix can be determined, but the length of the translation vector $\mathbf{t}$ is only known up to an unknown scaling $s > 0$. Additionally, it can be seen that four possible pairings of $\mathbf{S}$ and $\mathbf{R}$ are compatible with the essential matrix following Eq. 4.17 and Eq. 4.18 whereas only one solution is valid from the practical point of view. When using perspective cameras, three solutions obtained from the extracted correspondences produce 3D-points that lie behind one of the two cameras and only one that produces 3D-points lying in front of both cameras. In this manner, the correct solution for the rotation and translation matrices can be easily found for perspective cameras. However, this theorem is not suitable for omnidirectional cameras due to the circumferential view of the camera allowing for observations of points in front and behind the camera. Due to the mechanical device, however, the direction of the translation vector $\mathbf{t}$ is well-known. This reduces the number of potential solutions since only the correct rotation matrix $\mathbf{R}$ needs to be determined. Figure 4.7 illustrates two potential locations of a 3D-point $\mathbf{p}_n$ depending on the solutions for the rotation matrix $\mathbf{R}$.

To select the correct rotation matrix $\mathbf{R}$, the distance d_1 from the lower camera position C_0 and the distance d_2 from the upper camera position C_1 to each 3D-point $\mathbf{p}_n$ are computed considering both solutions of the rotation matrix. The correct rotation $\mathbf{R}$ along with the known translation will produces positive depths both for d_1 and for d_2 for any selected point (see Figure 4.7(a)). Wrong entries in the rotation matrix produce negative depths for d_1 or d_2 (see Figure 4.7(b)). However, tracked correspondences are noisy so that even the correct rotation $\mathbf{R}$ might cause negative depths. For this reason, Bunschoten [148] proposed selecting the solution for the rotation $\mathbf{R}$ that yields the most positive depths from both camera viewpoints.



(a) Intensity Image (b) Logarithmic Image

Figure 4.8: Intensity image (a) and logarithmic image (b) used for the selection of strong corners.

4.3.3 Feature point extraction

To determine and to refine the estimated camera pose, corresponding feature points of a 3D-scene are extracted in a pair of stereo images. Feature points are directly tracked in original images by the LKT feature tracker proposed by Kanade and Lucas [151] and by using the C-implementation for the LKT implemented by Shi and Tomasi [152]. In many approaches, feature tracking is performed in panoramic images [118]. Image transformation for applications where only low resolution images are available leads to a loss of information in particular in lower parts of panoramic images. Moreover, noise from the camera is amplified especially in lower regions of panoramic images so that wrong features could be determined in panoramic images. But the epipolar constraint derived for the parallel panoramic configuration is also valid for point correspondences obtained in original images. For this reason, 2D-image coordinates of detected feature points $\mathbf{m}_{n,i} \leftrightarrow \mathbf{m}_{0,i}$ in original images can be transformed into 3D-world coordinates using the camera model introduced in Section 2.2.

The LKT-feature tracker detects salient points by determining eigenvalues of local gradient matrices and requires strong corners to obtain good tracking results. However, corners of objects are rarely projected as strong corners into original images obtained by omnidirectional cameras. Therefore, the sensitivity of the LKT feature tracker must be decreased to obtain good tracking results. Consequently, disturbances are also detected and classified as valid feature points and must, hence, be removed to precisely estimate the essential matrix. To achieve this, an algorithm is proposed to select strong corners as feature point correspondences in pairs of original images for camera pose refinement. Due to disturbances such as noise and a decreased sensitivity of the LKT-tracker, neighboring pixels of a white wall may have different intensities and may hence be detected as valid feature points. In order to avoid feature points caused by disturbances, the algorithm transforms intensity images into the logarithmic domain in a first step.

Figure 4.8 illustrates an intensity image (see Figure 4.8(a)) and the corresponding logarithmic image (see Figure 4.8(b)) used to select strong feature points. The intensity and the logarithmic image captured from one position is called an image set. Feature points are extracted both in intensity and in logarithmic images using the LKT-feature tracker. Strong features are corners

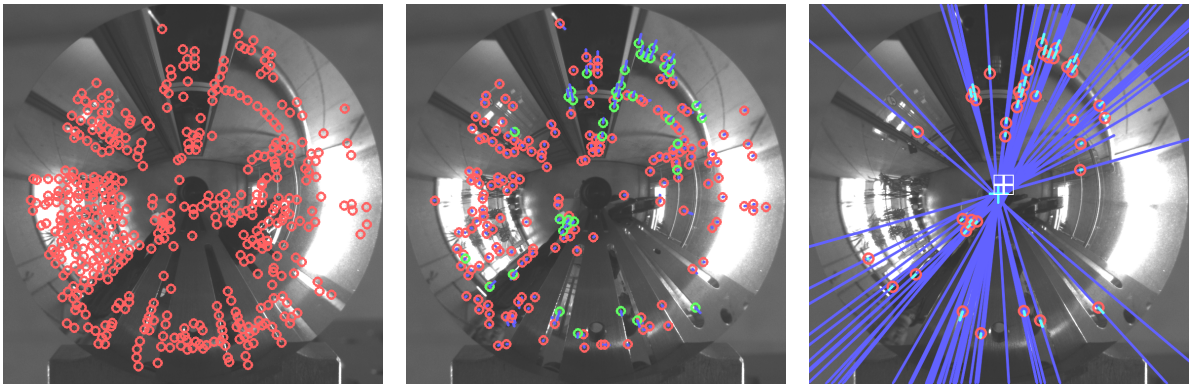
that occur both in the logarithmic and in the intensity image at nearly the same image position. The algorithm selects strong features and automatically removes weak features. Additionally, the algorithm recovers previously selected features in the next image set that has been captured at a new camera position. The 3D-world coordinates of strong feature points are computed and are stored in a matrix if the feature points have been successfully extracted in all image sets captured at different camera positions. The use of multiple image sets (> 2) improves feature point extraction, since only strong features can be tracked over the entire image sequence. The detected correspondences serve as an input for the 8-point algorithm.

The 8-point algorithm presented in the last section is very sensitive to noisy image correspondences. Hartley *et al.* [149] proposed a normalization method for homogeneous image coordinates to decrease the sensitivity of the 8-point algorithm to noisy image correspondences. However, homogeneous image coordinates are not available, so that the method presented by Hartley *et al.* is not suitable for this application. The 3D-vectors to the corresponding feature points $\mathbf{m}_n$ obtained at different camera positions are used instead (see above). Pajdla *et al.* [153] proposed normalizing these vectors by dividing them by their length to better condition the moment matrix $\mathbf{M}$. In this manner, they demonstrated a decreasing sensitivity of the 8-point algorithm against outliers and noise. Additionally, robust estimation techniques like Least of Median Square, RANSAC [154] and M-estimator [155] allow to identify and to eliminate outliers and, hence, to improve the pose estimation result.

Figure 4.9 illustrates strong feature point correspondences in a test scenario and the resulting epipolar lines after outlier removal. Figure 4.9(a) illustrates strong features that have been detected in both the logarithmic and the intensity image. The tracking result is presented in Figure 4.9(b), whereas true correspondences are highlighted in green and outliers in red. Figure 4.9(c) displays the resulting epipolar lines spanned by the (image) coordinates of corresponding feature points tracked in a pair of original images. For the parallel panoramic configuration, these epipolar lines must meet in the projection center (green cross). However, the two coordinate systems at pose $\mathbf{C}_0$ and $\mathbf{C}_1$ can be transformed into each other by a translation $\mathbf{t}$ and a rotation $\mathbf{R}$. In this manner, the epipolar lines do not meet in the projection center (see white cross) and a virtual rectification must be performed to obtain a parallel panoramic configuration instead.

4.3.4 Image rectification

Figure 4.6(b) illustrates a virtual parallel panoramic configuration generated for a non-parallel stereo setup (see Figure 4.6(a)). For real configurations, the poses $\mathbf{C}_0$ and $\mathbf{C}_1$ are related by means of a translation $\mathbf{t}$ and a rotation $\mathbf{R}$. The real poses can be transformed into virtual poses $\mathbf{C}_{r0}$ and $\mathbf{C}_{r1}$ to obtain a parallel panoramic configuration. In this configuration, the direction of the z-axes of the virtual coordinate systems must be co-linear with the direction of the baseline. A transformation function has to be found to align the real poses such a way that the z-axes of the virtual poses are co-linear with respect to the baseline. The transformation can be computed if the translation $\mathbf{t}$ and the rotation $\mathbf{R}$ between the real coordinate systems $\mathbf{C}_0$ and $\mathbf{C}_1$ are known. In a first step, the rotation $\mathbf{R}_{rectC0}$ is computed that aligns the reference coordinate system $\mathbf{C}_0$ to the direction of the baseline with respect to the z-axis of $\mathbf{C}_0$. With the rotation $\mathbf{R}_{rectC0}$, the virtual coordinate system $\mathbf{C}_{r0}$ is obtained from $\mathbf{C}_0$. Similarly, the rotation $\mathbf{R}_{rectC1}$



(a) Extracted Correspondences (b) Valid Correspondences (green) (c) Epipolar Lines

Figure 4.9: Strong feature point correspondences (a) and the resulting epipolar lines after outlier removal (b,c).

can be determined from the transformation $\mathbf{R}_{rectC0}$ and the rotation matrix $\mathbf{R}$ to transform the coordinate system $\mathbf{C}_1$ into the virtual coordinate system $\mathbf{C}_{r1}$. The rotation matrix $\mathbf{R}$ aligns the two camera coordinate systems $\mathbf{C}_0$ and $\mathbf{C}_1$ and has been determined during the camera pose estimation step.

Figure 4.10 illustrates a potential procedure to obtain the rotation matrix $\mathbf{R}_{rectC0}$ from the baseline (translation vector $\mathbf{t}$). The transformation of a coordinate system $\mathbf{C}_x$ into a rectified coordinate system $\mathbf{C}_{rx}$ can be performed using two rotation steps. After performing the rotation steps, the direction of the z-axis of the new coordinate system is co-linear with the direction of the baseline, viz. the direction of the translation vector $\mathbf{t}$. To achieve this, two rotations – one with respect to the z-axis and one with respect to the y-axis – are performed among others.

Let $\mathbf{t}_{ox} = [1\ 0\ 0]^T$ be the reference axis for the computation of the first rotation matrix γ with respect to the z-axis (see Figure 4.10(a)). The rotation γ and the corresponding rotation matrix $\mathbf{R}_z(\gamma)$ can then be computed as follows:

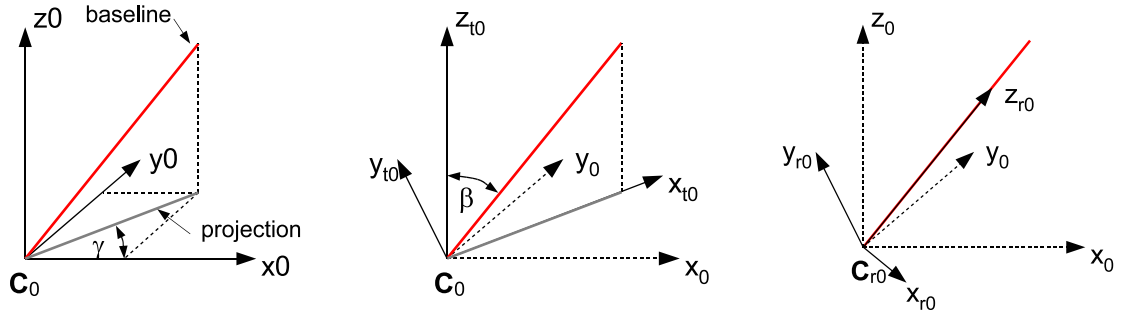
$$\gamma = \arccos\left(\frac{\mathbf{t}_{ox}^T \cdot \mathbf{t}_{Pxy}}{|\mathbf{t}_{ox}| \cdot |\mathbf{t}_{Pxy}|}\right) \tag{4.20}$$

The variable $\mathbf{t}_{Pxy} = [t_x\ t_y\ 0]^T$ is referred to as the projection of the baseline $\mathbf{t} = [t_x\ t_y\ t_z]^T$ onto the xy-plane (see Figure 4.10(a)). To compute the second rotation, the translation $\mathbf{t}$ and the vector of baseline must be transformed into a temporary coordinate system $\mathbf{R}_t$:

$$\mathbf{t}_t = \mathbf{R}_z(\gamma) \cdot \mathbf{t} \tag{4.21}$$

The y-axis of the temporary coordinate system is referred as the reference axis in order to obtain the second rotation β (see Figure 4.10(b)). The angle β spanned by the temporary translation $\mathbf{t}_t$ and the z-axis and, hence, the rotation matrix $\mathbf{R}_y(\beta)$ are computed as follows:

$$\beta = \arccos\left(\frac{\mathbf{t}_{oz}^T \cdot \mathbf{t}_{Pxz}}{|\mathbf{t}_{oz}| \cdot |\mathbf{t}_{Pxz}|}\right) \tag{4.22}$$



(a) (b) (c)

Figure 4.10: Determination of the rotation matrix $\mathbf{R}_{rectC0}$ for camera pose refinement.

Thereby, $\mathbf{t}_{oz} = [0\ 0\ 1]^T$ and $\mathbf{t}_{Pxz} = [t_{tx}\ 0\ t_{tz}]^T$ relate to the projection of the baseline $\mathbf{t}_t = [t_{tx}\ t_{ty}\ t_{tz}]^T$ onto the zx-plane (see Figure 4.10(b)) of the temporary coordinate system. Then, the rotation matrices $\mathbf{R}_{rectC0}$ and $\mathbf{R}_{rectC1}$ for image rectification to obtain a parallel panoramic configuration can be computed following Eq. 4.23 and Eq. 4.24:

$$\mathbf{R}_{rectC0} = \mathbf{R}_y(\beta) \cdot \mathbf{R}_z(\gamma) \tag{4.23}$$

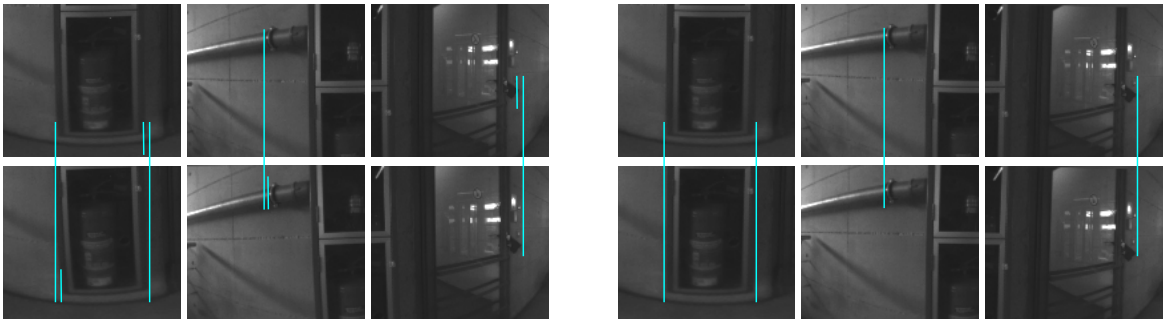
$$\mathbf{R}_{rectC1} = \mathbf{R}_{rectC0} \cdot \mathbf{R}^{\mathbf{T}} \tag{4.24}$$

Figure 4.11 illustrates a pair of panoramic images for non-parallel panoramic configuration (see Figure 4.11(a)) and the corresponding rectified pair of panoramic images using the virtually parallel panoramic configuration (see Figure 4.11(b)). Disparities cannot be obtained for non parallel panoramic configurations using 1D-vertical correspondence search along the image columns. The (virtually) parallel panoramic configuration, however, enables the performance of correspondence search in one dimension.

The length of the translation vector $\mathbf{t}$, viz. the baseline length, which is required to determine the 3D-world coordinates of any 3D-scene point by triangulation, can be precisely estimated by the translation sensor integrated within the camera platform. The coordinates of any 3D-scene point obtained from the disparity maps, however, are computed for the virtual parallel panoramic configurations and must, hence, be re-transformed into the original coordinate systems using the transposed rotation matrix $\mathbf{R}_{back} = \mathbf{R}_{rectC0}^T$.

4.4 Disparity map generation

Disparity map generation is the next step for obtaining 3D-ambiance information from a pair of stereo images. A 1D-correspondence search can be performed along epipolar lines in panoramic images obtained by cameras in a parallel panoramic configuration. The disparity δ is defined as the pixel-wise difference in the image location of the same correspondence (3D-scene point) seen in a pair of (panoramic) images. Therefore, stereo algorithms are designed to determine disparity maps from pairs of images on which triangulation and 3D-scene reconstruction are based. The extraction of correspondences and the generation of disparity maps are not possible



(a) Non parallel panoramic configuration.

(b) Parallel panoramic configuration.

Figure 4.11: Panoramic images from non-parallel panoramic configuration (a) and from a parallel panoramic configuration (b). 1D-correspondence search along image columns is performed for images satisfying the parallel panoramic configuration.

if one correspondence, which is viewable in one of the stereo images, is covered in the other one. This is in particular the case for stereo setups that have a large baseline. In this application, however, only a short baseline of $3.5cm$ between the two camera positions is required. The short baseline allows to determining valid depth information for objects close to the camera system only, but it overcomes the problem of covered correspondences in one of the stereo images.

Two methods exist to produce disparity maps from pairs of (rectified panoramic) images. One that is based on corresponding features and one that is based on an area-based block matching correspondence search [146]. The feature-based correspondence search is suitable for stereo images containing many edges, corners and features based on square angles. The advantage of feature-based algorithms is the use of a reduced amount of information within the pairs of stereo images. Additional knowledge of the scene context is required in order to match such features and to obtain geometric 3D-information of a scene. However, feature-based methods do not provide dense disparity maps required for some applications.

In contrast to feature-based matching methods, area-based block-matching algorithms provide dense disparity maps and can overcome the problem of providing additional knowledge of scene context. Due to the huge number of data, however, block-matching algorithms require more time for executing the generation of dense disparity maps.
In typical parking scenarios, objects such as parked cars or white walls typically do not have strong features such as corners or square angles that can be used for matching. Due to missing features, block-matching algorithms have been preferred over feature-based matching algorithms for this application. Moreover, the resolution of the camera system and thus of the panoramic images is too low to extract sufficient numbers of strong features. Block matching overcomes this limitation, and global optimization algorithms such as *dynamic programming* [156, 146] or *semi-global-matching* [136] allow for a refinement and an estimation of disparity information in less-textured and low-resolution panoramic images.

4.4.1 Local correspondence search

The simplest way to establish correspondences in a pair of images is by performing a maximum search or minimum search along 1D-epipolar lines using cost functions and block-matching algorithms. Banks *et al.* provide a good overview and an evaluation of the commonly used block-matching algorithms [157] that is briefly presented below. Figure 4.12 illustrates the general procedure for extracting correspondences with a local search algorithm [14]. A search window (template) with fixed block sizes M_B and N_B is generated for a certain image position (m_K, n_K) and contains the intensities $i_R(m, n)$ of each pixel within that template. This template is used to extract the corresponding template in the other image. Therefore, a target block with the same dimensions as the template is shifted over the other image starting from the position (m_K, n_K) to a position with the maximum distance δ_{max}. At the same time, a correlation value (matching cost) indicating the similarity between the template and the target block is computed for each disparity δ using correlation-based functions. The correlation value reaches its maximum/ minimum when the target block best matches the template block. The process described here is called *local correspondence search* using block-matching and is used to establish correspondences for each pixel in a pair of stereo images.

Eq. 4.25 represents a general function for computing the similarity s between a template and a target block. Index p represents the norm of the similarity function s and i_L, i_R are the intensities of pixels used for computation in the left and right images.

$$s_p(m,n) = \sum_{m_B=1}^{M_B} \sum_{n_B=1}^{N_B} \Big[i_R(m+m_B, n+n_B) - i_L(m+m_B, n+\delta(m,n)+n_B) \Big]^p \quad (4.25)$$

$$\text{with} \quad \delta(m,n) = 0 \ldots \delta_{max} \quad (4.26)$$

The similarity is computed for all disparities within a fixed disparity range $[0 \;..\; \delta_{max}]$ (see Eq. 4.26). The most probable disparity is the one where the (similarity) error between the template and search block is minimal (see Eq. 4.27).

$$\delta(m,n)_{opt} = \arg \min_{\delta(m,n)} [s(m,n)] \quad (4.27)$$

SAD, SSD and NCC

Many cost functions exist for computing the similarity between a reference and a target block. One of these functions is the SAD - cost function. Block-matching algorithms based on SAD (sum of absolute differences) use the absolute differences of intensities within a fixed block. SAD sums up intensities values for each pixel within one block. Two blocks are similar if the differences between the target and the reference block are minimal. Eq. 4.28 presents a mathematical expression for SAD:

$$s_{SAD}(m,n) = \sum_{m_B=1}^{M_B} \sum_{n_B=1}^{N_B} \Big[i_R(m+m_B, n+n_B) - i_L(m+m_B, n+\delta(m,n)+n_B) \Big] \quad (4.28)$$

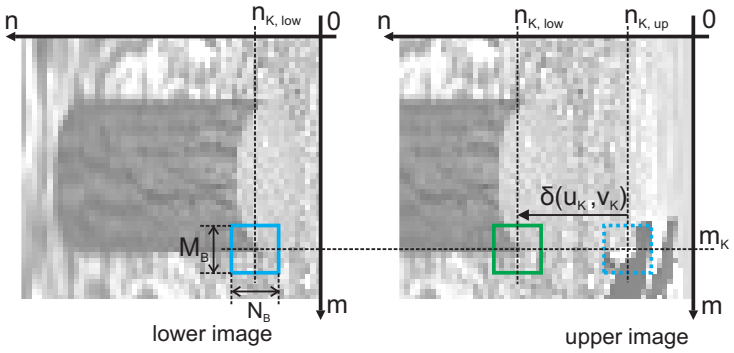


Figure 4.12: Block-matching with a block size of 9·9 pixels [14].

where parameter p is set to unity. In other words, the values of $s_{SAD}(m,n)$ become minimal when the target block best matches the template block at a certain disparity $\delta(m,n)$.

Similarly to block-matching algorithms based on SAD, other block-matching algorithms use the *sum of squared differences* (SSD) to penalize large differences between pixels and to better obtain a global minimum following Eq. 4.29:

$$s_{SSD}(m,n) = \sum_{m_B=1}^{M_B} \sum_{n_B=1}^{N_B} \Big[i_R(m+m_B, n+n_B) - i_L(m+m_B, n+\delta(m,n)+n_B) \Big]^2 \quad (4.29)$$

Eq. 4.29 can be reorganized to obtain terms that relate to the energy and similarity of the blocks.

$$s_{SSD}(m,n) = \sum_{m_B=1}^{M_B} \sum_{n_B=1}^{N_B} \Big[i_R(m+m_B, n+n_B) \Big]^2 +$$

$$\sum_{m_B=1}^{M_B} \sum_{n_B=1}^{N_B} \Big[i_L(m+m_B, n+\delta(m,n)+n_B) \Big]^2 - \quad (4.30)$$

$$2 \cdot \left(\sum_{m_B=1}^{M_B} \sum_{n_B=1}^{N_B} \Big[i_R(m+m_B, n+n_B) \cdot i_L(m+m_B, n+\delta(m,n)+n_B) \Big] \right)$$

The first two terms of Eq. 4.30 describe the energy and the third term the similarity between the target and the reference block [146]. However, the energy-related terms of the cost function strongly influence the results whereas the third term provides useful information for indicating the similarity between two blocks only. For this reason, the normalized cross correlation (NCC) has been introduced to make cost functions independent of the energy of the target and template blocks by normalizing the third term of Eq. 4.30 with the energy of the image blocks:

$$s_{SSD}(m,n) = \frac{\sum_{m_B=1}^{M_B} \sum_{n_B=1}^{N_B} \Big[i_R(m+m_B, n+n_B) \cdot i_L(m+m_B, n+\delta(m,n)+n_B) \Big]}{\sqrt{\sum_{m_B=1}^{M_B} \sum_{n_B=1}^{N_B} \Big[i_R(m+m_B, n+n_B) \Big]^2 + \sum_{m_B=1}^{M_B} \sum_{n_B=1}^{N_B} \Big[i_L(m+m_B, n+\delta(m,n)+n_B) \Big]^2}}$$

(4.31)

In contrast to SAD and SSD, cost functions based on NCC have a maximum at the disparity $\delta(m,n)$ when the target block best matches the template block.

Zero-mean correlation functions (ZSAD, ZSSD, ZNCC)

Different illumination conditions cause offsets in the intensity values of stereo images and may lead to poor minimization and maximization result for SAD, SSD and NCC-based cost functions. This influence can be reduced by subtracting the mean intensity of a target block from the intensities of each pixel within a block before computing the similarity. In Eq. 4.32, formalism is presented as an example for the SAD correlation function for subtracting the mean-intensity value of a block from the intensity values of the pixels within the block. The new function obtained is called the ZSAD-cost-function (zero-mean sum of absolute differences).

$$s_{SAD}(m,n) = \sum_{m_B=1}^{M_B} \sum_{n_B=1}^{N_B} \Big[\big(i_R(m+m_B,n+n_B) - \overline{i_R}(m,n)\big) - \big(i_L(m+m_B,n+\delta(m,n)+n_B) - \overline{i_L}(m,n)\big) \Big] \quad (4.32)$$

where

$$\overline{i_R}(m,n) = \frac{1}{M_B \cdot N_B} \cdot \sum_{m_B=1}^{M_B} \sum_{n_B=1}^{N_B} \Big[i_R(m+m_B, n+n_B) \Big] \quad (4.33)$$

and

$$\overline{i_L}(m,n) = \frac{1}{M_B \cdot N_B} \cdot \sum_{m_B=1}^{M_B} \sum_{n_B=1}^{N_B} \Big[i_L(m+m_B, n+\delta(m,n)+n_B) \Big] \quad (4.34)$$

Similarly to the ZSAD, subtracting the mean value is also performed for SSD-based and NCC-based correlation functions (ZSSD, ZNCC).

Rank transformation

Cost functions such as SAD, SSD or ZNCC can use intensity values to compute the similarity between target and reference blocks. However, a common method to increase the robustness of local correspondence search algorithms is transforming intensity-based target and reference blocks into lighting invariant blocks. For this purpose, the rank transformation (T_{Rank}) is suitable for transforming the intensity value of each pixel into intensity independent values. The transformation replaces the intensity of target pixels by a value that indicates the number of neighboring pixels within a block $M_{Rank} \cdot N_{Rank}$ having lower intensity values than the target pixel. The block size of the rank transformation can differ from the block size of target and template blocks for computing the similarity. A pixel can take the following values after rank transformation: $[0 \ldots (M_{Rank} \cdot N_{Rank} - 1)]$. The disparity of each pixel can then be established by determining the similarity between a target and a reference block using block matching algorithms based on cost functions such as SAD or NCC [158].

Census transformation

Similarly to the Rank-transformation, the Census-transformation (T_{Census}) also replaces intensity-based values of image pixels by intensity-independent values [158]. The new value computed for an image pixel relates to a bit vector whose entries (bit positions) describe the intensities of target pixels relative to a reference pixel. In other words, a bit in this vector is set to one if the intensity value of the corresponding pixel is lower than the intensity value of the reference pixel and zero otherwise. The length of the vector is $(M_{Census} \cdot N_{Census} - 1)$ and represents the number of pixels within a transformation block with size $M_{Census} \cdot N_{Census}$. Disparity maps can be computed from the census-transformed pixel values and by means of a block-matching algorithm whose cost function is based on the Hamming distance s_{Hamm} (see Eq. 4.35). The Hamming distance indicates the number of bits in a vector that differ from the bits in another vector with the same length.

$$s_{Hamm}(m,n) = \sum_{m_B=1}^{M_B} \sum_{n_B=1}^{N_B} \left[D_{Hamming}\left(T_{Census,R}(m+m_B, n+n_B) - T_{Census,L}(m+m_B, n+\delta(m,n)+n_B) \right) \right] \tag{4.35}$$

4.4.2 Global correspondence search

Local matching methods are based on correlation functions and can have very efficient implementations. However, these methods assume continuous disparities along an epipolar line. This assumption is not fulfilled at discontinuities in images such as object borders and leads to blurred object boundaries. This effect can be reduced by certain techniques [156] but cannot be eliminated [136]. Global matching algorithms can overcome this limitation by optimizing disparities obtained that were computed by local correspondence search algorithms and by incorporating constraints for unambiguous matching. Ambiguities are resolved by considering global information and constraints such as the uniqueness constraint. The uniqueness constraint describes the fact that only one valid correspondence can be found in a target image for a correspondence established in a reference image.

Global correspondence search methods incorporate previously obtained information and stereo constraints such as the uniqueness constraint or the continuity constraint in the optimization procedure and introduce energy functions for minimizing the total energy of matching costs in order to determine optimal disparity maps [156]. This is an advantage over local correspondence search algorithms and increases the robustness in estimating true disparities in pairs of stereo images. Blurred boundaries of objects can be avoided. However, global search methods are more complex than local correspondence search methods due to their optimization and refinement algorithms. These algorithms are able to generate robust disparity maps from cost values obtained by a local correspondence search algorithm.

Very popular global matching algorithms are based on dynamic programming. Dynamic programming was introduced by Richard Bellman [159] to solve complex problems in mathematics and in computer science. The idea of dynamic programming is to break complex problems

down into simpler steps. Although dynamic programming is performed along the epipolar lines in stereo vision, it is nevertheless referred to as a global matching method. Point candidates are extracted along epipolar lines in rectified images and are checked against each other to find true image correspondence. The cost values for each disparity δ of a correspondence are stored in a cost matrix when performing local correspondence search. The best path through the cost matrix can be found by the dynamic programming algorithm by means of minimizing the costs of the disparities along the epipolar lines [160, 144, 146, 161, 162]. A prerequisite for dynamic programming is the ordering constraint describing the fact that true correspondences must have the same order in each pair of stereo images [146]. Dynamic programming is a powerful global search method and is suitable for closing gaps in disparity maps caused by local correspondence search methods due to missing textures. The generation of streaking artifacts is the limitation of stereo algorithms based on dynamic programming. Streaking artifacts are caused by disparity optimizations along individual image rows and particularly appear in disparity maps obtained from low textured panoramic images. In this case, the optimization process does not consider disparities of neighboring image rows in 2D that may be useful for increasing the robustness in generating disparity maps.

In general, stereo algorithms based on dynamic programming optimize correspondence search along one particular search direction by assuming the continuity constraint. Other algorithms and methods extend this approach by also assuming valid continuity constraints for other directions. Zitnick *et al.* [163] proposed a cooperative correspondence search algorithm that uses the continuity constraint for both vertical and horizontal search directions. This algorithm performs optimization over the whole stereo image and especially considers the influence of valid pixel correspondences to neighboring pixel correspondences. Similar to dynamic programming, their algorithm iteratively optimizes 3D cost matrices $[u,\ v,\ d]$ assuming the uniqueness and the continuity constraint. The optimization process stops when a unique maximum or minimum has been established [144].

4.4.3 Semi-global matching

A very recent, robust global stereo algorithm technique has been proposed by Hirschmüller [136, 106]. The main purpose of this algorithm is to aggregate matching costs and to consider constraints from all search directions equally in order to minimize the energy of the matching costs over the whole image. The algorithm computes path costs $L(\mathbf{p}, d)$ for a certain number of paths $\mathbf{p}$ as a first step. The disparity d of a correspondence at an image position is determined by minimizing the accumulated path costs $S(\mathbf{p}, d)$ of all 1D search paths $\mathbf{p}$. The path costs are computed for each disparity d. In that context, each path starts at an outer image pixel and ends at the target image position. Hirschmüller mentioned that only the path costs are required for disparity computation and not the path itself. Due to a path search over the whole image, the algorithm is called *semi-global matching*. Figure 4.13 illustrates the aggregation of path costs for four and for eight search paths. However, semi-global matching does not consider the ordering constraint. The ordering constraint can only be fulfilled for paths that coincide with the epipolar lines. Hence, semi-global matching relates more to a scan-line optimization algorithm [136]. The computation of correlation costs using local matching methods can lead to wrongly

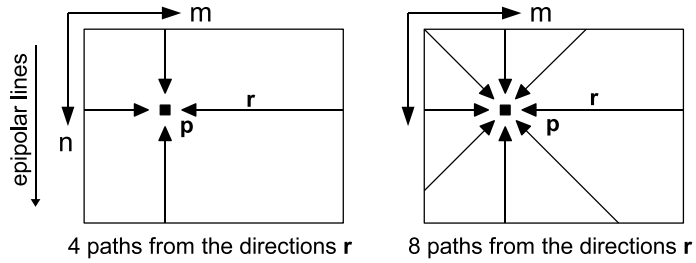


Figure 4.13: Aggregation of cost for four (left) and eight (right) search paths.

determined point correspondences in pairs of stereo images. Due to image noise or illumination changes within the search windows, wrong matches may have lower costs than correct matches and may, hence, be classified as true correspondences. Semi-global matching overcomes this problem by introducing a consistency constraint that is also used to resolve ambiguities. The constraint is formulated recursively by defining path costs $L_{\mathbf{r}}(\mathbf{r}, d)$ along fixed search paths $\mathbf{r}$ [164]. $L_{\mathbf{r}}$ is referred to as the path cost in a certain direction $\mathbf{r}$ and $L_{\mathbf{r}}(\mathbf{p}, d)$ is referred to as the cost value of a position $\mathbf{p}$ for a disparity d. The costs $L_{\mathbf{r}}(\mathbf{p}, d)$ are computed as follows.

$$L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + min \begin{bmatrix} L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d), \\ L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d + 1) + P_1, \\ L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d - 1) + P_1, \\ \min\limits_{l} L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, l) + P_2 \end{bmatrix} - \min\limits_{l} L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, l) \tag{4.36}$$

where $l \in [0, d_{max}]$. $C(\mathbf{p}, d)$ is referred as the pixel by pixel matching cost of a pixel $\mathbf{p}$ for a disparity d. The costs $C(\mathbf{p}, d)$ can be computed by using local matching methods based on SAD, SSD, NCC, RANK or mutual information. In this application, the matching costs are computed by using a SAD-based cost function and RANK-transformed stereo input images. The second term of Eq. 4.36 selects the minimum values of previously computed path costs at the pixel positions $\mathbf{p} - \mathbf{r}$ for several disparities $(d + 1)$ and $(d - 1)$ and aggregates the chosen value to the path costs.

Hirschmüller proposes the penalties P_1 and P_2 for reducing the influence of discontinuities caused by image noise and for stressing the consistency constraint. Different values of the penalties can be chosen to distinguish between small discontinuities caused by slanted or curved surfaces (P_1) and large discontinuities caused by outliers in the disparity map (P_2). The penalties stress the consistency constraint and can also be used to adjust the smoothness of disparity maps. Large values of the penalties lead to smooth disparity maps since noise in the disparity maps caused by disturbances is mostly suppressed. The disparity maps contain more details when choosing lower values for the penalties, but also contain more noise and outliers. In general, large intensity differences of pixels along a search path $\mathbf{r}$ come along with discontinuities in disparity maps. Therefore, an intensity gradient has been proposed to automatically adjust the penalty P_2 to potential discontinuities [136, 164].

$$P_2 = \frac{P_2'}{|I(\mathbf{p}) - I(\mathbf{p} - \mathbf{r})|} \tag{4.37}$$

where P_2' is a fixed constant with $P_2' \leq P_1$. In other words, the algorithm considers discontinuities potentially caused by object boundaries and decreases the penalty P_2 when detecting large intensity gradients. Additionally, edge images generated from the panoramic images can be used to predict potential discontinuities and to adapt the penalty P_2 to take objects' boundaries into account. The last term $\min_l L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, l)$ in Eq. 4.36 prevents a continuous increase of the path costs $L_{\mathbf{r}}(\mathbf{p}, d)$ along the search path $\mathbf{r}$. This term is a constant for all disparities at a pixel position $\mathbf{p}$ and does not change the location of a potential minimum/ maximum in the disparity space. After path cost computation, the sum $S_{\mathbf{r},d}$ over all paths $\mathbf{r}$ is computed for each disparity d following Eq.4.38.

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d) \tag{4.38}$$

The disparity at a pixel position is determined by selecting the disparity d that corresponds to the minimum or maximum costs. A quadric curve can be interpolated across the cost values of neighboring disparities to obtain sub-pixel accuracy (see Eq. 4.39, [136]).

$$d_{sub} = d + \frac{S(\mathbf{p}, d+1) - S(\mathbf{p}, d+1)}{2 \cdot S(\mathbf{p}, d-1) - 4 \cdot S(\mathbf{p}, d) - 2 \cdot S(\mathbf{p}, d+1)} \tag{4.39}$$

In stereo vision, each disparity image corresponds to a reference image. This dependency can be used to perform a simple consistency check for enforcing the uniqueness constraint. For this purpose, the disparity map D_{up} relating to the upper panoramic image and the disparity map D_{lo} relating to the lower panoramic image are computed. Each disparity $D_{up\mathbf{p}}$ at a pixel position $\mathbf{p}$ in the disparity map D_{up} is compared with the corresponding disparity $D_{lo\mathbf{p}}$ from the disparity map D_{lo}. The resulting disparity is set to zero when the disparities of both disparity maps differ.

4.5 Generation of 3D-ambiance information

4.5.1 Triangulation

The disparity maps obtained from a stereo setup can be used to compute 3D ambiance information by means of *triangulation*. Triangulation is the process of determining the location and the distance to a 3D point by measuring two angles θ and ρ to this point from at least two distinct positions A and B. The distance ΔZ between these two positions A and B is called the *baseline length* and must be known. The points A, B and P span a triangle with one known side length ΔZ and two known angles θ and ρ. For this reason, the length of vector $\mathbf{p}_u$ with $d_2 = |\mathbf{p}_u|$ from point A to P and the length of vector $\mathbf{p}_l$ with $d_1 = |\mathbf{p}_l|$ from point B to P can be computed using *triangulation* (see Figure 4.14).

In technical applications, optical 3D measuring systems use at least one camera as one of the two sensors for triangulation. The other device can be a light projector, an additional camera or the same camera used to measure the angles to a scene point from several positions. Triangulation is also suitable for generating 3D ambiance information with omnidirectional cameras if

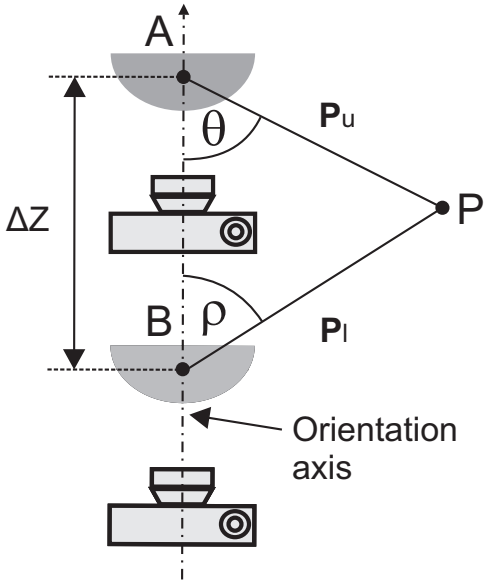


Figure 4.14: Stereo setup with omnidirectional cameras suitable for obtaining distance information by triangulation [14].

correspondences to 3D scene points can be obtained from images captured at different camera poses. In this application, the distance d_1 from the camera position B to a point P has to be determined. In this configuration, position B relates to the lower position of the camera platform. The advantage of determining the distance d_1 from the lower camera position to the point P is in the precise calibration of the camera coordinate system to the car coordinate system. Following the law of sine [165], Eq. 4.40 is valid for all plain triangles:

$$\frac{d1}{\sin(\theta)} = \frac{d2}{\sin(\rho)} = \frac{\Delta Z}{\sin(\varphi)} \tag{4.40}$$

Eq. 4.40 can be reorganized to compute both the distances d_1 and d_2 depending on the baseline length ΔZ and the two angles θ and ρ. Eq. 4.41 presents an expression for the distance d_1 from B to P and for the distance d_2 from A to P:

$$d1 = \sin(\theta) \cdot \frac{\Delta Z}{\sin(\varphi)} \quad \text{and} \quad d2 = \sin(\rho) \cdot \frac{\Delta Z}{\sin(\varphi)} \tag{4.41}$$

with $\varphi = 180 - \rho - \theta$. The vectors $\mathbf{p}_l$ and $\mathbf{p}_u$ in world coordinates can be estimated if the calibration data of the camera and the sensor coordinates of the point correspondences are available. With this information, angles θ and ρ are computed as follows:

$$\rho = 90^\circ - \arctan\left(\frac{z_{\vec{p_u}}}{\sqrt{x_{\vec{p_u}}^2 + y_{\vec{p_u}}^2}}\right) \quad \theta = 90^\circ + \arctan\left(\frac{z_{\vec{p_o}}}{\sqrt{x_{\vec{p_o}}^2 + y_{\vec{p_o}}^2}}\right) \tag{4.42}$$

Images correspondences obtained from rectified, spherical panoramic images are used in this application: Each pixel in a spherical image represents both a vertical and a horizontal solid

angle. Both are suitable for triangulation. For this reason, an expression can be derived to directly determine the length of the vectors d_1 and d_2 from the elevation angles θ and ρ and from the disparities δ_P obtained from a disparity map. For spherical panoramic images, the angles θ and ρ can be directly determined from the image rows. In other words, the rows in the rectified image captured at the upper platform position represent the solid angles θ whereas the rows in the image captured at the lower position represent the solid angles ρ. The tilting offset β_{offset} of the projection area and the number of vertical pixels N in a rectified image are obtained from the calibration data and from the rectification process. Eq. 4.43 defines then the vertical angle β_P to a scene point P as seen in the spherical projection area:

$$\beta_P = \left(\beta_{offset} + \frac{\beta}{2}\right) - \left(\frac{\beta}{N} \cdot n\right) \tag{4.43}$$

The angle β_P is also called the elevation angle whose direction is defined anti-clockwise beginning at the x-axis of the image coordinate system. θ and ρ can then be computed following Eq. 4.44 and Eq. 4.45:

$$\rho_P = 90° - \left(\beta_{offset} + \frac{\beta}{2}\right) + \left(\frac{\beta}{N} \cdot n_l\right) \tag{4.44}$$

$$\theta_P = 90° + \left(\beta_{offset} + \frac{\beta}{2}\right) - \left(\frac{\beta}{N} \cdot n_u\right) \tag{4.45}$$

with

$$\begin{aligned} \rho_P &= 90° - \beta_{P\_l} \\ \theta_P &= 90° + \beta_{P\_u} \end{aligned} \tag{4.46}$$

Value n_l represents the image rows of the lower camera system and n_u the image rows of the upper camera system. However, this assumption is only valid for identical projection parameters for both the upper and lower camera system. Using $\varphi = 180 - \rho - \theta$, Eq. 4.47 represents an expression to obtain angle φ

$$\varphi_P = \left(\beta_{offset} + \frac{\beta}{2}\right) - \left(\frac{\beta}{N} \cdot n_l\right) - \left(\beta_{offset} + \frac{\beta}{2}\right) + \left(\frac{\beta}{N} \cdot n_u\right) \tag{4.47}$$

which is reduced to Eq. 4.48:

$$\varphi_P = \left(\frac{\beta}{N}\right) \cdot (n_u - n_l) \quad \text{mit} \tag{4.48}$$

The disparity δ_P of correspondences in a pair of spherical panoramic images for a point P is defined as follows.

$$\delta_P = n_u - n_l \tag{4.49}$$

whereas $\Delta\beta = \frac{\beta}{N}$ is defined as the vertical solid angle. Starting with Eq. 4.41 and using Eq. 4.48, an expression can be derived to compute the length of vector $d_1 = |\mathbf{p}_l|$ as follows:

$$d1_P = \sin(\theta_P) \cdot \frac{\Delta Z}{\sin(\delta_P \cdot \Delta\beta)} \quad \text{with} \quad \theta_P = 90° + \left(\beta_{offset} + \frac{\beta}{2}\right) - (\Delta\beta \cdot n_o) \tag{4.50}$$

After reorganizing Eq. 4.50, Eq. 4.51 describes an expression to determine d_1.

$$d1_P = \cos\left(\left(\beta_{offset} + \frac{\beta}{2}\right) - (\Delta\beta \cdot n_o)\right) \cdot \frac{\Delta Z}{\sin(\delta_P \cdot \Delta\beta)} \tag{4.51}$$

The argument of the cosine function in Eq. 4.51 corresponds to the angle β_P to a scene point P as seen by the upper camera system (see Eq. 4.42). Finally, Eq. 4.52 gives an expression for estimating the distance d_1 from the projection center of the lower camera system to a scene point P using spherical panoramic images.

$$d1_P = \cos\left(\beta_{P\_u}\right) \cdot \frac{\Delta Z}{\sin(\delta_P \cdot \Delta\beta)} \tag{4.52}$$

4.5.2 Disparity map reduction

The proposed algorithm uses extracted disparity maps to obtain 3D information about objects in the surroundings of the car door. The 3D information serves as input to the car door control unit that enables or stops car door operations in order to avoid collisions with obstacles next to the door. Clearly, the control unit computes the risk of potential collisions with static objects and generates collision-free motion sequences to safely operate the door. For this purpose, the control unit requires 3D-ambiance information in form of spherical or rectangular bounding boxes. These bounding boxes represent the 3D-geometry of the ambiance close to the door and are used for generating opening paths and motion sequences to safely open the car door. Theoretically, each pixel in a disparity map could be used to compute the location of a 3D scene point and its distance to the car. But even small disparity maps with 480×204 pixels would generate 97920 bounding boxes that would overlap each other due to their fixed size. Moreover, the control unit must compute door opening paths in real-time. For this reason, only few bounding boxes can be used for representing obstacles close to the car. Consequently, a rough 3D model of the ambiance close to the door is sufficient for collision avoidance in this application. Besides, other applications such as augmented reality or telepresence [144] require detailed geometric 3D information and texture for scene reconstruction.

Even small disparity maps provide dense depth information that is not required when using bounding boxes for roughly modeling the ambiance close to the car. Thus, the resolution of the disparity maps can be reduced and hence the resulting number of bounding boxes. When reducing the disparity maps, it is important to consider objects that are located close to the car door. Correspondences of such objects have larger disparity values than correspondences of objects having larger distances to the car. Therefore, a maximum reduction scheme can be performed to preserve large disparity values of objects located close to the door. Figure 4.15(a) illustrates the maximum reduction scheme for dense disparity maps using reduction windows with sizes M_{Red} and N_{Red}. However, one limitation of the maximum-based reduction scheme is the direct mapping of outliers, which can have very large disparity values, into the reduced disparity map. To overcome this limitation, a median-based reduction scheme could be used to suppress outliers in disparity maps.

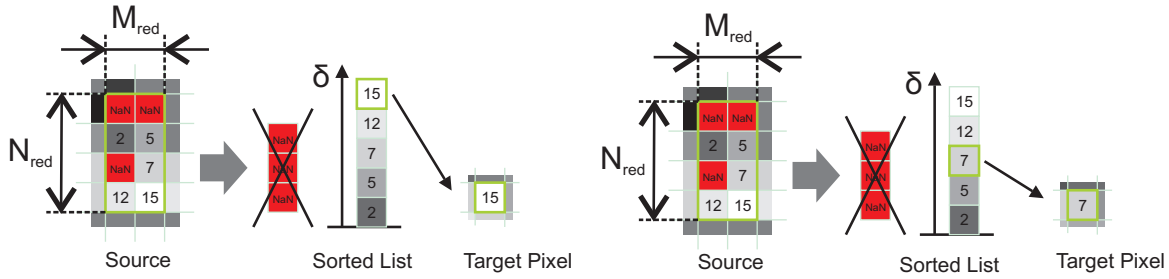
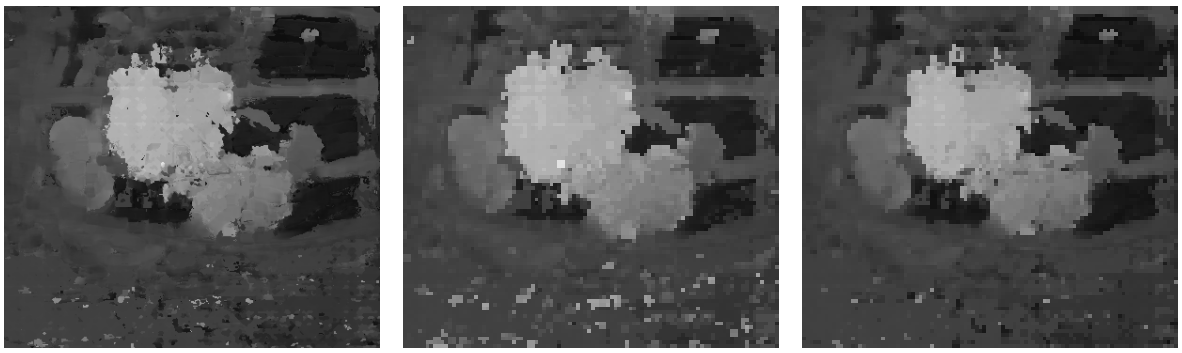


Figure 4.15: Disparity map reduction using a maximum-based reduction scheme (a) and a median-based reduction scheme (b) [14].



(a) Original Disparity Map (b) Maximum Reduction (c) Median Reduction

Figure 4.16: Reduction of disparity maps (a) using maximum reduction (b) and median reduction (c).

Figure 4.15(b) illustrates the median-based reduction scheme to better suppress outliers in disparity maps. But disparities of small objects close to the car door might also be suppressed by the median-based reduction scheme. Figure 4.16 illustrates the maximum and the median reduction for an exemplary disparity map (see Figure 4.16(a)) using a reduction window with a 4×4 pixel size. The maximum-based reduction scheme directly maps large disparity values to target disparity maps. Consequently, noisy disparities caused by disturbances are directly mapped into the final disparity map and object boundaries are increased (see Figure 4.16(b)). Median reduction can better preserve object boundaries and decreases the influence of outliers on the reduced disparity map (see Figure 4.16(c)).

However, disparity maps must not be reduced when information of the quantization error (see Section 4.6.1) is required for further processing stages. The reduction process would falsify the computation of the quantization error intervals caused by a change of the resolution in panoramic images and also by a change of the resolution of the solid angles. A distance information map must be computed from the disparity map instead and the resolution of the distance map can be reduced to decrease the total number of bounding boxes. In this case, a minimum-based reduction method must be applied for selecting the least distances in a set of different distance values. In stereo vision, large disparity values relate to short distances between a camera and the objects and vice versa.

4.5.3 Bounding box refinement

Disparity maps obtained from low resolution panoramic images may contain inhomogeneities and outliers. In particular, less textured regions and image noise lead to many wrongly determined disparities and hence to wrongly computed locations of bounding boxes for ambiance modeling. These disturbances must be removed in order to increase the robustness of the ambiance modeling algorithm. One way to overcome these disturbances is post-processing the disparity maps using median-filters or morphologic image operators [144]. Unfortunately, such post-processing methods are not suitable for disparity maps obtained from low resolution panoramic images. Disparities of small objects are wrongly misclassified as disturbances and are removed. Another difficulty is in determining valid ambiance information from less textured objects such as white walls in a parking garage. No useful disparities can be generated for such regions. Geometric information of the ambiance next to the car-door could be used instead for both generating distance information about less-textured regions and for bounding box refinement. For the most common parking situations, there are many objects such as walls or flower boxes that touch the ground and whose surfaces are assumed to be perpendicular to the floor. Additional 3D information can be obtained from edges appearing in images when an object is in contact with the ground. The 3D location of these edges can be precisely computed by means of triangulation and may be used for the bounding-box refinement stage.

For this purpose, the $x,\ y$ coordinates of edge information and the $x,\ y$ coordinates of objects in contact with the ground and whose surface orientations are perpendicular to the orientation of the ground are assumed to be similar. The refinement stage uses this assumption to remove outliers in the bounding boxes. Clearly, bounding boxes are classified as outliers and are removed if they are located beyond a specified distance $\pm\Delta_{out}$ from the coordinates $x,\ y$. But this condition is not suitable for objects with slanted surfaces or for objects that do not touch the floor – e.g. baskets attached to light posts. In order to avoid wrong refinement results, the refinement stage tries to detect bounding boxes that relate to slanted surfaces based on the disparity maps. Additionally, it preselects bounding boxes by restricting the search area $(x \pm \Delta_{in}, y \pm \Delta_{in}), \Delta_{in} > \Delta_{out}$ depending on the edge/floor coordinates $x,\ y$.

Figure 4.17 illustrates the bounding box refinement state for a parking scenario where a car is parked close to a wall. Bounding boxes and the edge between the floor and the wall are obtained from the disparity map and from a panoramic image. Outliers and disturbances are then detected and removed to better approximate the surface of objects by means of the bounding boxes.

4.6 Error estimation

This section analyzes the quantization and the calibration error intervals for stereo applications with omnidirectional cameras. Both errors are suitable for determining the accuracy of 3D data obtained from disparity maps (bounding boxes). The calibration error depends on the reprojection error caused by approximating the mirror function when calibrating omnidirectional cameras. The quantization error depends on the camera resolution and on the resolution of

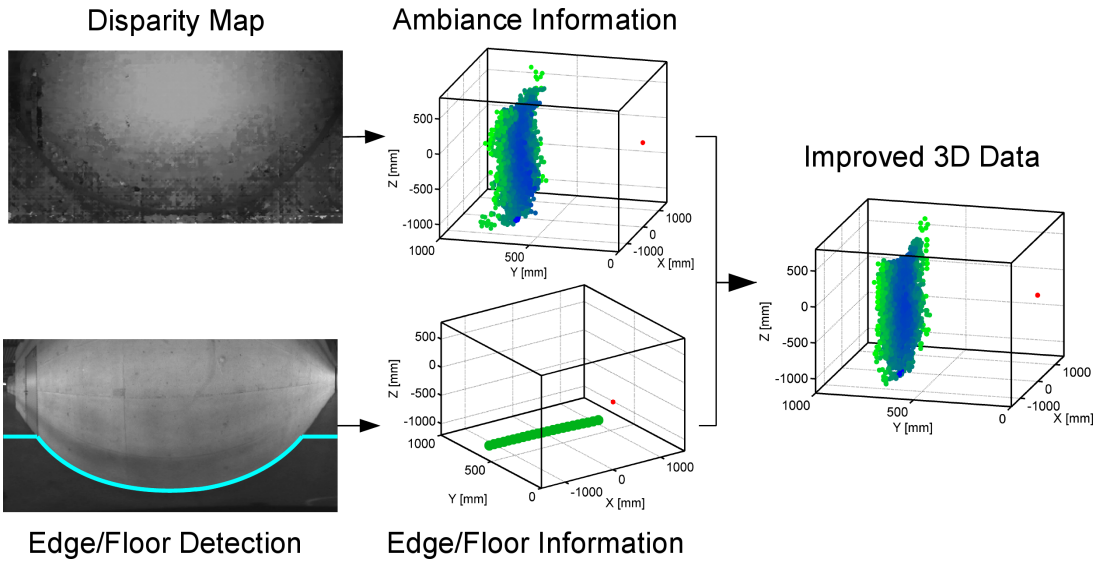


Figure 4.17: Bounding box refinement stage that is based on edge/floor information of objects located on the floor.

disparity maps. These errors have been described by Florian Böhm [14] in his masterthesis and are briefly presented below.

4.6.1 Quantization error

The quantization error is a measure for indicating differences between analoque values and the corresponding quantized digital values. This error is caused by rounding or truncation. There are two quantization errors in image processing – one that occurs when digitalizing the analoque intensity of an image pixel into a discrete value and one caused by the resolution of the camera. In this thesis, the position error of determined 3D data is called the *quantization error of the disparity*. The error is caused by a limited resolution of disparity maps and also by the resolution of the disparity values obtained from a stereo algorithm. The quantization error of the disparity d_{1Q} and of a point P at an image position $P(m,n)$ can be described in spherical coordinates (α_Q, β_Q) (see Eq. 4.53).

$$\begin{aligned} [\alpha_{Q,min} \dots \alpha_{Q,max}] &= \alpha_P \pm \frac{\Delta\alpha}{2} \\ [\beta_{Q,min} \dots \beta_{Q,max}] &= \beta_P \pm \frac{\Delta\beta}{2} \\ [d_{1,Q,min} \dots d_{1,Q,max}] &= \left[d1\left(\delta_P + \frac{\Delta\delta_Q}{2}\right) \dots d1\left(\delta_P - \frac{\Delta\delta_Q}{2}\right) \right] \end{aligned} \tag{4.53}$$

The solid angles $\Delta\alpha$ and $\Delta\beta$ in a panoramic image depend on the resolution of the images. Additionally, $\Delta\delta_Q$ describes the resolution of the disparity and depends on the accuracy of the chosen stereo algorithm. The accuracy of $\Delta\delta_Q$ is less than unity for stereo algorithms with sub-pixel accuracy and equal to or larger than unity for stereo algorithms with an accuracy of one

pixel. For spherical projections, the minimum and maximum distances $d_{1,Q,min}$ and $d_{1,Q,max}$ of the distance error interval $\Delta d1_Q$ can be computed following Eq. 4.54 and Eq. 4.55 and depend on the resolution of the disparities $\Delta\delta_Q$. Thereby, the quantization error of the elevation angle β_P can be neglected due to $\Delta d1_Q \gg \Delta\beta_P$ for common omnidirectional cameras.

$$d_{1,Q,min} = d_1\left(\delta_P + \frac{1}{2}\right) = \cos\left(\beta_P\right) \cdot \frac{\Delta Z}{\sin\left(\left(\delta_P + 0,5\Delta\delta_Q\right) \cdot \Delta\beta\right)} \tag{4.54}$$

$$d_{1,Q,max} = d_1\left(\delta_P - \frac{1}{2}\right) = \cos\left(\beta_P\right) \cdot \frac{\Delta Z}{\sin\left(\left(\delta_P - 0,5\Delta\delta_Q\right) \cdot \Delta\beta\right)} \tag{4.55}$$

Clearly, the distance error interval $\Delta d1_Q$ denotes an interval within which distance values are represented by the same disparity value. This interval is not symmetrical to the distance d_1 and its size depends on the upper and on the lower boundaries of the distance values $d_{1,Q,min}$ and $d_{1,Q,max}$ (see Eq. 4.56).

$$\Delta d1_Q = d_{1,Q,max} - d_{1,Q,min} \tag{4.56}$$

Eq. 4.57 represents the quantization error interval $\Delta d1_Q$ in spherical coordinates.

$$\Delta d1_Q = \cos\left(\beta_P\right) \cdot \frac{\Delta Z}{\sin\left(\left(\delta_P - 0,5\right) \cdot \Delta\beta\right)} - \cos\left(\beta_P\right) \cdot \frac{\Delta Z}{\sin\left(\left(\delta_P + 0,5\right) \cdot \Delta\beta\right)} \tag{4.57}$$

After reorganizing Eq. 4.57, Eq. 4.58 presents the quantization error interval $\Delta d1_Q$ that depends on the resolution of the disparity $\Delta\delta_Q$, on the elevation angle, on the disparity and on the baseline length ΔZ (see Eq. 4.58).

$$\Delta d1_Q = \cos\left(\beta_P\right) \cdot \Delta Z \cdot \left(\frac{1}{\sin\left(\left(\delta_P - 0,5\Delta\delta_Q\right) \cdot \Delta\beta\right)} - \frac{1}{\sin\left(\left(\delta_P + 0,5\Delta\delta_Q\right) \cdot \Delta\beta\right)}\right) \tag{4.58}$$

In this equation, the quantization error is represented in spherical coordinates. However, it is also available for other projections like cylindrical, conical or plane projections.

4.6.2 Calibration error

The quantization error is used to determining the accuracy of re-computed 3D data from scene points P. Besides, the quality of the camera calibration, in particular the approximation of the mirror function, influences the accuracy of computed distances between scene points P and the projection center of the camera. The calibration process proposed by Scaramuzza [27] allows for the determination of a mean camera calibration error E_{calib} for omnidirectional cameras. The accuracy of the camera model can only be estimated up to a minimum error due to misalignments and inaccuracies of the mirrors. The re-projection error is determined by computing the differences between the true positions of chessboard corners and the re-projected positions of chessboard corners in calibration images – the mean re-projection error obtained from all chessboard corners is defined as the calibration error for omnidirectional cameras [27]. The

calibration error indicates the quality of the camera calibration and is valid for all pixels in original images. The calibration error of an ideal camera system is zero.

Additionally, the pixel density σ can be used for determining the calibration error of pixels in rectified panoramic images. The pixel density is a value for indicating the number of sensor pixels that are projected onto a pixel position in panoramic images for certain projections. The pixel density depends on the projection type of panoramic images and on the pixel positions in rectified images. Hence, the pixel density also influences the accuracy of 3D information obtained from an image point $P(n, m)$ and must be considered. Eq. 4.59 describes the error intervals in spherical coordinates caused by the calibration error.

$$\begin{aligned} [\alpha_{C,min} \dots \alpha_{C,max}] &= \alpha_P \pm \frac{\Delta\alpha_C}{2} \\ [\beta_{C,min} \dots \beta_{C,max}] &= \beta_P \pm \frac{\Delta\beta_C}{2} \\ [d1_{C,min} \dots d1_{C,max}] &= \left[d1\left(\delta_P + \frac{\Delta\delta_C}{2}\right) \dots d1\left(\delta_P - \frac{\Delta\delta_C}{2}\right) \right] \end{aligned} \tag{4.59}$$

The error intervals of the horizontal and vertical angles $\Delta\alpha_C$ and $\Delta\beta_C$ can be directly computed by dividing the calibration error E_{calib} by the horizontal and the vertical pixel densities $\sigma_{P,h}$ and $\sigma_{P,v}$.

$$\begin{aligned} \Delta\alpha_C &= \frac{E_{calib}}{\sigma_{P,h}} \cdot \Delta\alpha \\ \Delta\beta_C &= \frac{E_{calib}}{\sigma_{P,v}} \cdot \Delta\beta \end{aligned} \tag{4.60}$$

Again, the calibration error influences the error intervals and depends on the chosen projection and on the position P of the point correspondences in panoramic images. For the spherical projection, Eq. 4.62 considers this by combining Eq. 4.59 and Eq. 4.60 as follows.

$$\begin{aligned} [\alpha_{C,min} \dots \alpha_{C,max}] &= \alpha_P \pm \left(\frac{E_{calib}}{\sigma_{P,h}} \cdot \frac{\Delta\alpha}{2} \right) \\ [\beta_{C,min} \dots \beta_{C,max}] &= \beta_P \pm \left(\frac{E_{calib}}{\sigma_{P,v}} \cdot \frac{\Delta\beta}{2} \right) \\ [d1_{C,min} \dots d1_{C,max}] &= \left[d1\left(\delta_P + \frac{\Delta\delta_C}{2}\right) \dots d1\left(\delta_P - \frac{\Delta\delta_C}{2}\right) \right] \end{aligned} \tag{4.61}$$

In contrast to the quantization error intervals, symmetrical distance error intervals $[\alpha_{C,min}, \alpha_{C,max}]$ and $[\beta_{C,min}, \beta_{C,max}]$ are assumed for error estimation in order to facilitate their derivation. The distance error interval $\Delta\delta_C$ of a disparity is estimated under the assumption of identical mirrors and hence of an identical calibration error for omnidirectional cameras used in a stereo-setup. This assumption can always be met in motion stereo applications due to the same camera moved to different poses. Otherwise, the twofold calibration error of the elevation angles must be considered when estimating the distance error of the disparity. However, the assumptions

presented above can only be fulfilled for small disparities due to the position-dependent values of the pixel density σ. In this application, this can be guaranteed by the small baseline length ($\approx 3.5cm$) and by the resulting disparity range of at most $30pixels$. If the stereo-camera system had a baseline length larger than $5cm$, then the pixel density for each camera system or at each pose would have to be considered for precisely estimating the influence of the calibration error to the disparity. Eq. 4.62 presents an expression that describes the influence of the calibration error E_{calib} on the disparity $\Delta\delta_C$.

$$\Delta\delta_C = 2 \cdot \frac{E_{calib}}{\sigma_{P,v}} \tag{4.62}$$

Similarly to Eq. 4.54 and Eq. 4.55, the minimum distance $d_{1,min,C}$ and the maximum distance $d_{1,max,C}$ can be computed following Eq. 4.63 and Eq. 4.64. The influence of the calibration error E_{calib} on the elevation angle β_P is negligible due to $\Delta\delta_C \gg \Delta\beta_P$ for common omnidirectional cameras.

$$d1_{min,C} = d1_P\left(\delta_P + \frac{\Delta\delta_C}{2}\right) = \cos\left(\beta_P\right) \cdot \frac{\Delta Z}{\sin\left((\delta_P + \Delta\delta_C) \cdot \Delta\beta\right)} \tag{4.63}$$

$$d1_{max,C} = d1_P\left(\delta_P - \frac{\Delta\delta_C}{2}\right) = \cos\left(\beta_P\right) \cdot \frac{\Delta Z}{\sin\left((\delta_P - \Delta\delta_C) \cdot \Delta\beta\right)} \tag{4.64}$$

The maximum disparity error $\Delta\delta_C$ is subtracted from δ_p within the sin arguments of the terms of the maximum distance $d_{1,max,C}$ (see Eq. 4.55) and of the maximum distance $d_{1,max,Q}$ (see Eq. 4.64). Disparity values of scene points with infinite distances are zero. This case must be considered by means of an automatic quantization and calibration error estimation algorithm, since subtraction would lead to negative disparities.

4.7 Results

In this section, the results of the proposed ambiance detection algorithm are presented and discussed. Section 4.7.2 analyzes and discusses the influence on the calibration error and the quantization error to distance estimation. The calibration and quantization error depend on the chosen projection for panoramic images, and the results obtained for spheric, conic and cylindric projection are studied and discussed. Section 4.7.3 illustrates the disparity maps generated from low-resolution panoramic images using the semi-global matching stereo algorithm [136]. Experiments were conducted for four test scenarios and their disparity maps were computed. The results obtained were compared with disparity maps generated by a stereo algorithm based on dynamic programming [144]. The resulting 3D information (bounding boxes) for the four test scenarios is presented in Section 4.7.4. Experiments with ground truth data were also conducted to determine the accuracy of the position of the bounding boxes. This sections ends with a presentation of the results obtained for the bounding box refinement stage and the execution times for generating ambiance information.

In this section, the influences of the calibration and the quantization error on distance estimation based on stereo vision is analyzed. The calibration error is independent of the chosen projection

whereas the quantization error depends on the chosen projection. Experiments are conducted to analyze the properties of the quantization error using cylindric, spheric and conic projections for panoramic images. The projections are also suitable for studying the properties and the accuracies of disparity maps obtained from a pair of panoramic images. For this purpose, a simulated parallel panoramic configuration with a baseline length of $\Delta Z = 5cm$ is used. Studies illustrated no significant differences in the results obtained from experiments using real- or simulated data: Therefore, data from a simulated camera system were preferred over data from a real camera system to better illustrate the results.

Figure 4.18(a) illustrates a simulated scenario used for analyzing and illustrating the calibration and quantization error. Figure 4.18(b) displays the locations of determined bounding boxes (yellow spheres) and the corresponding calibration error intervals (red cylinders). The calibration error is independent of the chosen projection and indicates the quality of camera calibration [26]. The calibration error obtained by means of camera calibration causes a distance interval in which 3D points can be located for a certain disparity. The calibration error of an ideal camera system is zero. Similarly, the quantization error arises due to the limited resolution of the camera sensor and, hence, of the limited resolution of solid angles as seen by the camera system. The quantization error interval represents the dimensions of a 3D area in which a reconstructed 3D scene point can be located for a certain disparity with regard to a solid angle. The quantization error depends on the resolution of the disparity map computed by a stereo algorithm and proportional to the solid angles for all types of projections.

Figure 4.18(c) illustrates the dimensions of quantization error intervals and Figure 4.18(d) the cumulated error combining the quantization and calibration error interval. Additionally, Figure 4.18 displays a correlation between the distances to 3D points (as seen by the camera) and the sizes of the calibration and quantization error intervals. Following Eq. 4.62 in Section 4.6.2, the calibration error intervals increase with decreasing disparities and vice versa. In particular, distance error intervals increase for points with large distances to the camera. Similarly to the calibration error intervals, the quantization error intervals increase for decreasing disparities following Eq. 4.59, Section 4.6.1.

4.7.1 Accuracy and resolution of the projections

The cylindric projection is commonly used in robotics to transform original images into panoramic images and to generate depth information from stereo setups with omnidirectional cameras. Therefore, the geometry of the imaging device is adjusted to the properties of the cylindric projection for obtaining optimal resolutions of panoramic images [166, 167]. For this reason, an accuracy analysis of 3D data obtained from stereo panoramic vision is done with regard to cylindric projection in literature only. In this thesis, however, the accuracy of 3D information is analyzed based on calibration and quantization error intervals for several pairs of panoramic stereo images. These panoramic images have been obtained from original images using conical and spherical projections for image transformation. The results are compared with cylindrical panoramic images.

In a first step, the accuracy of 3D data obtained from pairs of conical, spherical and cylindrical stereo panoramic images is analyzed. Therefore, an ideal stereo algorithm is presumed that

Cylindric Projection	Conic Projection	Spheric Projection
$d = 1m$	$d_{top} = 1m$	$\beta = 90°$
$Z_{top} = 0,5658m$	$Z_{top} = 0,5658m$	$\beta_{offset} = -15,5°$
$Z_{bottom} = -1,7675m$	$d_{bottom} = -0,3201m$	
	$Z_{bottom} = -0.5658m$	

Table 4.1: This table illustrates the parameters chosen for image rectification to analyze the accuracy of 3D information depending on the chosen projection.

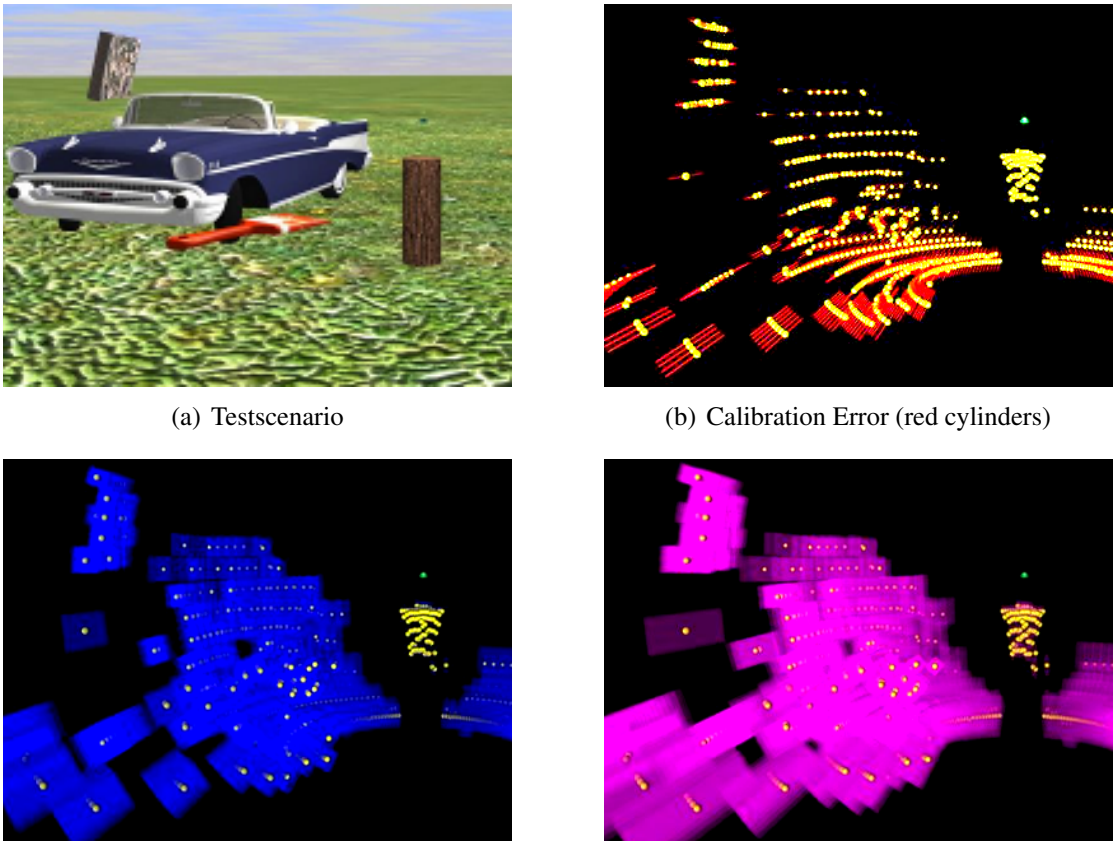
establishes correspondences with an accuracy of 1 pixel. Panoramic images that use the same field of view of the camera for several projections are generated to better compare the results. Table 4.1 gives an overview of the projection parameters used in this setup for conic, cylindric and spheric projections. The results presented below are computed for a parallel panoramic configuration with a baseline length of $\Delta Z = 5cm$. All computed distances d_2 for accuracy analyses are related to the upper camera system as follows:

$$d2 = \sin(\rho) \cdot \frac{\Delta Z}{\sin(\varphi)} \tag{4.65}$$

The results obtained for this configuration can be directly transferred to other omnidirectional camera-based stereo configurations with baseline lengths $\Delta Z \neq 5cm$. It is shown that the baseline length ΔZ is directly proportional to the distance d_2 (see Eq. 4.65). The elevation angles ρ and φ of scene points P_i as seen by the cameras can be directly obtained from the image coordinates in panoramic images if the omnidirectional camera is calibrated. Figure 4.19 illustrates the quantization and the calibration error intervals over the distances for cylindrical, conical and spherical panoramic images using the tested parallel panoramic configuration (see Table 4.1). For the rest of this chapter, the quantization and calibration error intervals refer to the distance error intervals for a certain disparity.

4.7.2 Quantization and calibration error

The calibration error is independent of the projection type and can, hence, be used as a reference for comparing the different quantization error intervals of several projections. The quantization error intervals should ideally be identical or less than the calibration error intervals for a certain projection. In other words, the calibration error intervals are suitable as a reference measure for dimensioning the quantization error intervals in terms of best resolution in pairs of panoramic stereo images. The resolution of the solid angles and, hence, of panoramic images should be chosen, so that the resulting dimensions of the quantization error intervals are identical with those of the calibration error intervals. Increasing the resolution of solid angles in panoramic images would lead to smaller quantization error intervals but at the same time to a loss of (texture) information in panoramic images. In a nutshell, the ratio between the calibration and the quantization error intervals indicates the usability of a chosen projection for a certain stereo application.



(a) Testscenario (b) Calibration Error (red cylinders)

(c) Quantization Error (d) Cumulated Error

Figure 4.18: Calibration and quantization error intervals for a simulated parallel panoramic configuration with baseline length $\Delta Z = 5cm$ (see [14]).

Figure 4.19, top, illustrates the quantization and calibration error intervals for cylindrical panoramic images at certain elevation angles $-40°$, $0°$ and $20°$. The difference between the calibration and quantization error intervals at the elevation angle of $-40°$ (see Figure 4.19(a)) is larger than the differences at the elevation angles $0°$ and $20°$ (see Figure 4.19(b) and Figure 4.19(c)). The reason for this is the bad conformity of the cylindric projection to the geometry of the camera mirror. Moreover, the course of the quantization error intervals strongly differs for different elevation angles. The conic projection area relates more to the geometry of the mirror and leads, hence, to a better conformity of the calibration and quantization error intervals for all elevation angles (see Figure 4.19(d), Figure 4.19(e) and Figure 4.19(f)). This results in smaller differences between the calibration and quantization error intervals, but the course of the quantization error interval also differs for certain elevation angles.

The best course of the quantization error intervals can be obtained for spherical panoramic images at all elevation angles (see Figure 4.19(g), Figure 4.19(h) and Figure 4.19(i)). The advantage of using spherical panoramic images is in the approximately constant courses of the quantization error intervals for all elevation angles. This leads to more homogeneous quantization error intervals over all disparities for the whole panoramic image and, hence, to an easier

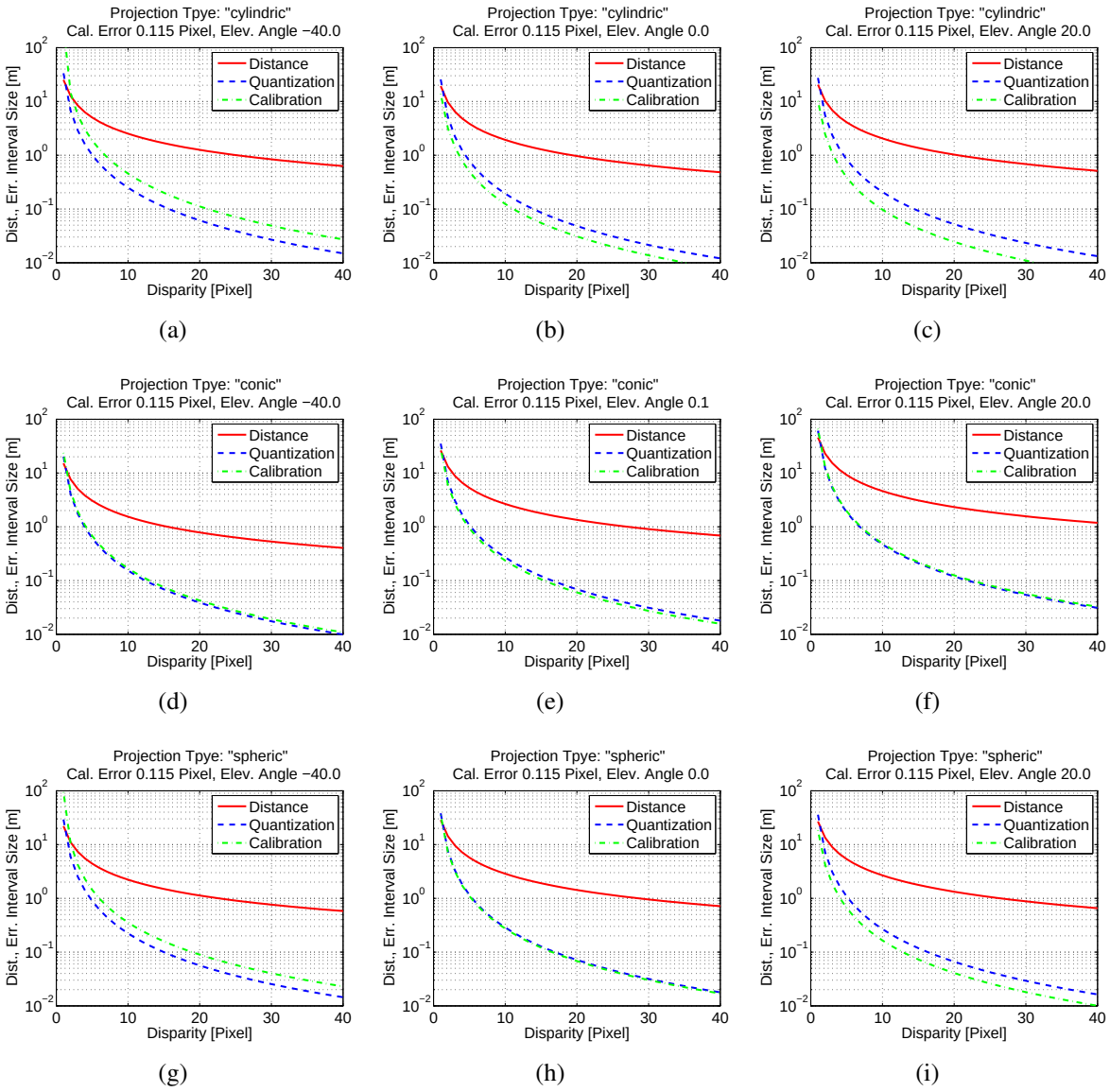


Figure 4.19: Quantization and calibration error intervals for cylindric (top), conic (middle) and spheric (bottom) projections for certain elevation angles $-40°$, $0°$ and $20°$ [14].

parametrization of panoramic images. Figure 4.20 illustrates a direct comparison of the quantization error intervals for several projections over the distance to the camera system for the test angles $-40°, 0°, 20°$. The quantization error intervals of conical panoramic images best match the calibration error intervals for all tested angles. For this reason, the conic projection seems to be the best projection in terms of the resolution of the solid angles in panoramic images and hence for obtaining 3D information.

Figure 4.21(a) illustrates the ratio of the quantization error intervals over the calibration error intervals for cylindric, conic and spheric projections for fixed disparity $\delta = 30\ pixels$. This ratio can be used for optimally adjusting the resolution of panoramic images – and, hence, the resolution of the solid angles – to the calibration error intervals. In other words, the ratio is

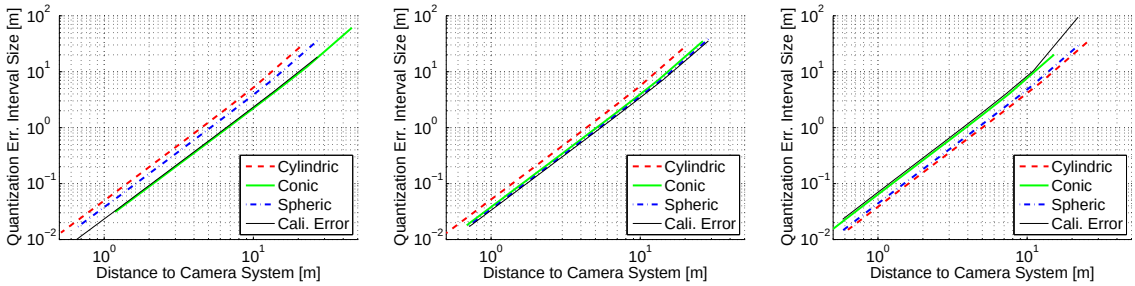


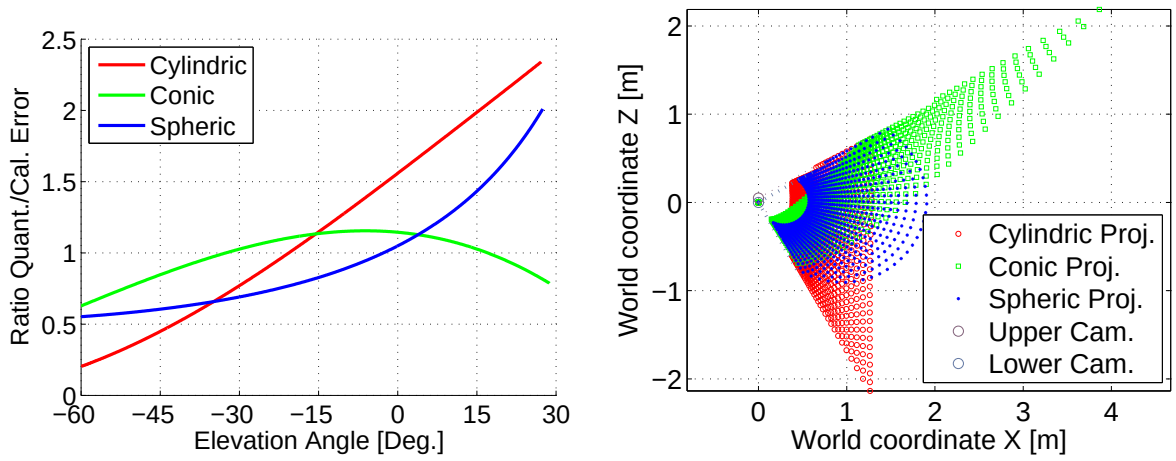
Figure 4.20: Comparison of the calibration and quantization error interval for spheric, conic and cylindric projections [14].

suitable for designing stereo setups with omnidirectional cameras optimized for obtaining 3D information with smallest possible distance errors having best resolution in panoramic images at the same time. For all test angles, the conic projection seems to be the best one for this stereo configuration due to little variance of the ratio over the elevation angles. Cylindric projection may also be a good alternative to the conic projection when the elevation angle interval $\beta \in [-60°..5°]$ is used only. For both the cylindric and spheric projection, the ratio increases for elevation angles larger than $10°$.

Figure 4.21(b) illustrates the location of 3D world points computed from disparity maps – for disparities within the interval $\delta = [8Pixel..40Pixel]$ – obtained from cylindric, conic and spheric panoramic images. This figure also illustrates the different measurement ranges and blind zones for the presented projections. A blind zone exists in front of the stereo camera system in which no distance information can be obtained. The dimensions of blind zones depend on the disparities and the chosen projections. Increasing the disparity range would reduce the blind zones but would also lead to higher computation times for disparity maps. In this setup, only obstacles with a distance of at least $50cm$ have to be detected so that the chosen disparity range is sufficient.

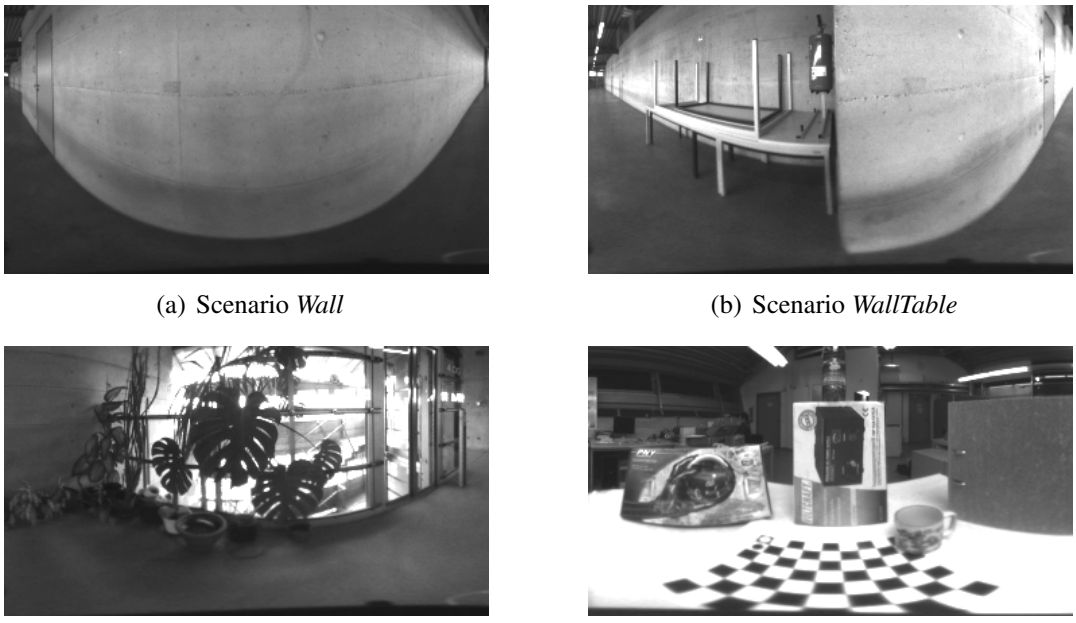
4.7.3 Disparity maps

In this section, the disparity maps, which are computed by the semi-global matching stereo algorithm, and the results of ambiance modeling are presented and discussed. Four test scenarios have been chosen for evaluation, three of them existing in real parking situations. The first scenario presents a parking scenario where a car is parked close to a wall. This scenario is called the *Wall* scenario and can occur when parking in a parking garage next to a wall. Figure 4.22(a) illustrates a panoramic image captured from this scenario. The second scenario is similar to the first parking scenario but contains more complex scene content with discontinuities and less textured objects such as fire extinguisher and tables. This scenario is called the *WallTable* scenario and is illustrated in Figure 4.22(b). The third scenario simulates a parking scenario close to plants (see Figure 4.22(c) and is called the *Plant* scenario. For all scenarios, the obstacles have a distance of at least $50cm$ to the car door and, hence, to the camera system. This is a prerequisite for successfully opening car doors to enable safe ingress/egress. A fourth



(a) (b)

Figure 4.21: Ratio of calibration and quantization error intervals for all cylindric, conic and spheric projection [14].



(a) Scenario *Wall* (b) Scenario *WallTable*

(c) Scenario *Plant* (d) Scenario *Table*

Figure 4.22: The four test scenarios *Wall* (a), *WallTable* (b), *Plant* (c) and *Table* (d) used for stereo evaluation. The scenarios *Wall*, *WallTable* and *Plant* can be understood as real-life parking scenarios, scenario *Table* serving as benchmark.

test scenario – called *Table* – contains objects located close to the camera system. Moreover, this scenario contains well-textured objects and serves as a reference scenario to evaluate the generated disparity maps and bounding boxes of the other scenarios. Figure 4.23 illustrates the obtained disparity maps for the four test scenarios using the semi-global matching algorithm

and different penalties P_1, P_2 where $P_1 < P_2$ (see Section 4.4.3). The penalties P_1 and P_2 are introduced to distinguish between large discontinuities and small discontinuities caused by slanted or curved objects. The disparity maps contain very detailed information and fine disparities for small values of the penalties P_1 and P_2. However, the disparity maps also contain more gaps and disturbances caused by noise. Increasing the penalties P_1 and P_2 leads to disparity maps being more robust against outliers or gaps, but fine disparities obtained from small objects are also suppressed. In other words, high penalties P_1 and P_2 introduce some kind of smoothness and are suitable for suppressing noise and disturbances, but they also suppress disparities of fine objects at the same time. In this thesis, one way to overcome this limitation is to use dynamically adapted penalties. Discontinuities in disparity maps relate to large intensity differences and cause edges in panoramic images. Therefore, edges are detected in panoramic images in a first step. The semi-global matching algorithm then decreases the penalties P_1, P_2 by $P_2 = P_2/10$, $P_1 = P_1/10$ when detecting edges in order to maintain disparities of small objects. This way, dynamically adapted penalties maintain disparities of small objects but suppress noise and close gaps in disparity maps at the same time.

Rank transformation with a block size of 13×13 pixels along with a pixel by pixel difference computation is used to compute the local matching costs. For this reason, the maximum difference d between two values in a rank transformed image and hence the maximum value for matching costs $C(x, y)$ is $d = C_{max} = N^2$ – in this application $d = C_{max} = 169$. N represents the size of the transformation window used for rank transformation. Experiments demonstrated that the penalties should be chosen in such a way that $P_2 \approx 1.5 \cdot C_{max}$ and $P_1 = P_2/4$ in order to obtain dense disparity maps. Figure 4.23 illustrates the disparity maps for the test scenarios *Table*, *Wall*, *WallTable* and *Plant* using different dynamic penalties P_1 and P_2. It cam be seen that disparity maps contain many disturbances in low-textured regions such as floor for small penalties. These disturbances are successfully removed by using larger penalties whereas disparities of fine structures are maintained.

In further experiments, the disparity maps obtained from the semi-global matching stereo algorithm [136] are compared to the disparity maps obtained from a stereo algorithm based on dynamic programming. Georg Passig [144] provides a toolbox used for computing disparity maps based on dynamic programming. Figure 4.24 illustrates the comparison results for the four test scenarios. Streaking effects are the limitation of stereo algorithms based on dynamic programming due to their optimization of disparities in a certain direction by incorporating constraints for unambiguous matching. In particular, the streaking effects occur in low-textured regions such as floor regions for all scenarios. Semi-global-matching overcomes the streaking artifacts by global optimizing path costs for all directions equally. However, disparities obtained for regions from less-textured objects such as table legs or the elevator frame benefit from the streaking effects that fill gaps in disparity maps.

4.7.4 Ambiance reconstruction

Once disparity maps are obtained, bounding boxes are generated to represent the geometric properties of the environment close to the car door. Figure 4.25 shows 3D point clouds (bounding boxes) computed from the disparity maps for the scenarios *Table*, *Wall*, *WallTable* and *Plant*.

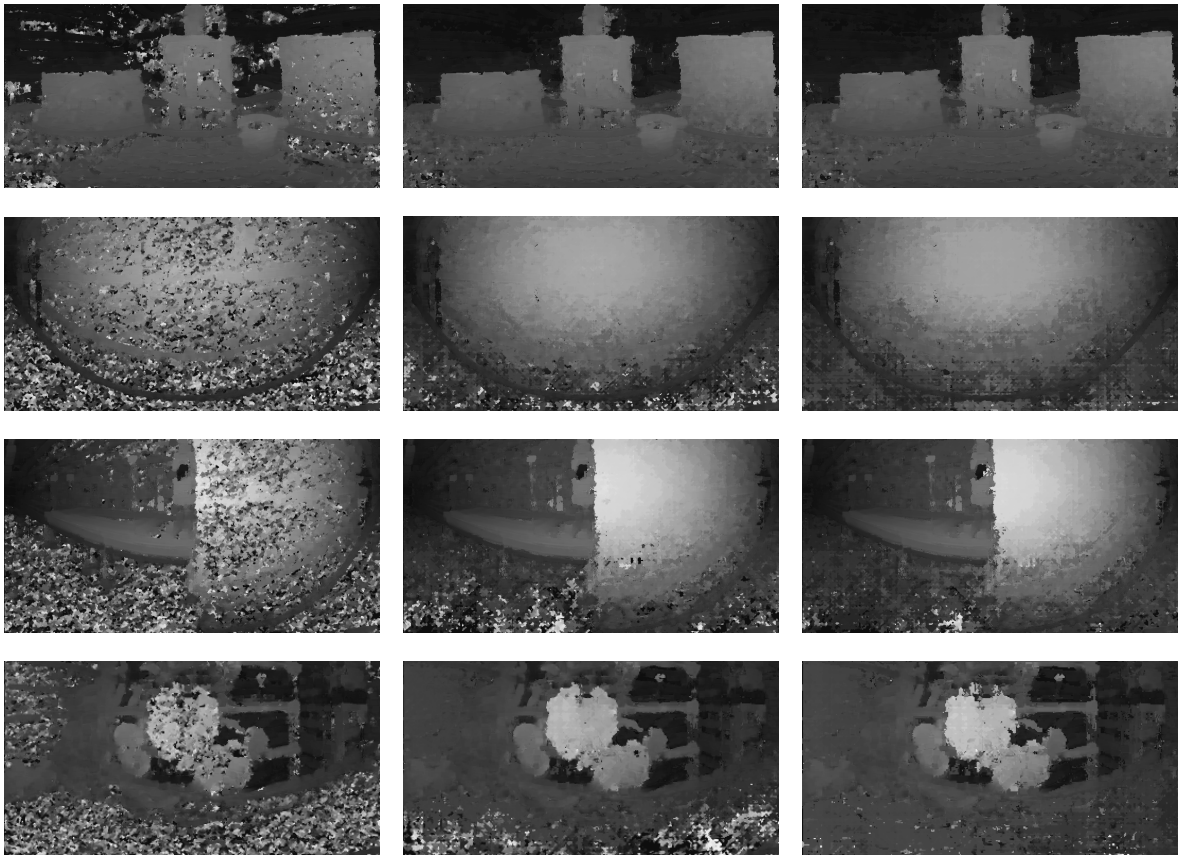


Figure 4.23: Disparity maps computed for the for test scenarios with the penalties $P1 = 20, P2 = 80$ (left), $P1 = 40, P2 = 160$ (middle) and $P1 = 65, P2 = 260$ (right).

The origin of the omnidirectional camera is specified as the reference coordinate system (world coordinate system). Bounding boxes obtained from objects located close to the camera system can be precisely determined and match the surfaces of objects very well. Figure 4.25(a) illustrates the bounding boxes obtained for the scenario *Table*. Due to well-textured objects located close to the camera system, there are fewer outliers and inaccuracies in the 3D data.
Figure 4.25(b) also demonstrates a good approximation of the wall for the test scenario *Wall*. Additionally, bounding boxes are generated for modeling the geometric structure of the scenarios *WallTable* and *Plant*. In these scenarios, the disparity maps contain many wrongly determined disparities caused by poorly textured objects such as table legs or floor regions.

Figure 4.25(c) shows a good representation of the front wall in scenario *WallTable* with bounding boxes. However, there are vast gaps in the bounding boxes representing the rear wall caused by missing texture or inaccurate disparities. Figure 4.25(d) shows bounding boxes obtained for the fourth scenario *Plant*. The camera was located close to the leaves of the plant to obtain good correspondences in panoramic images. Additionally, well-textured regions can be found in the left background and on the floor. However, the floor or the elevator cannot be precisely modeled by bounding boxes due to missing texture and noise (see Figure 4.25(d)).

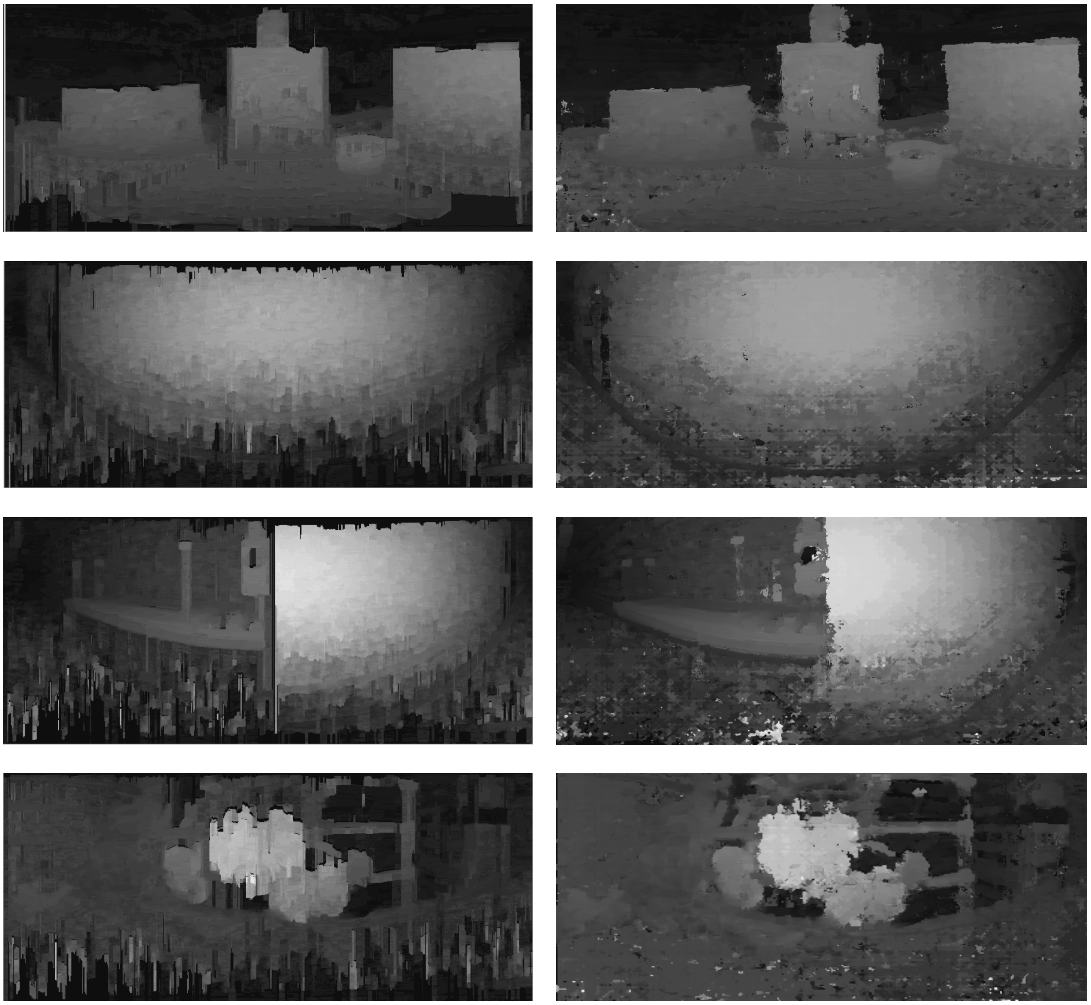
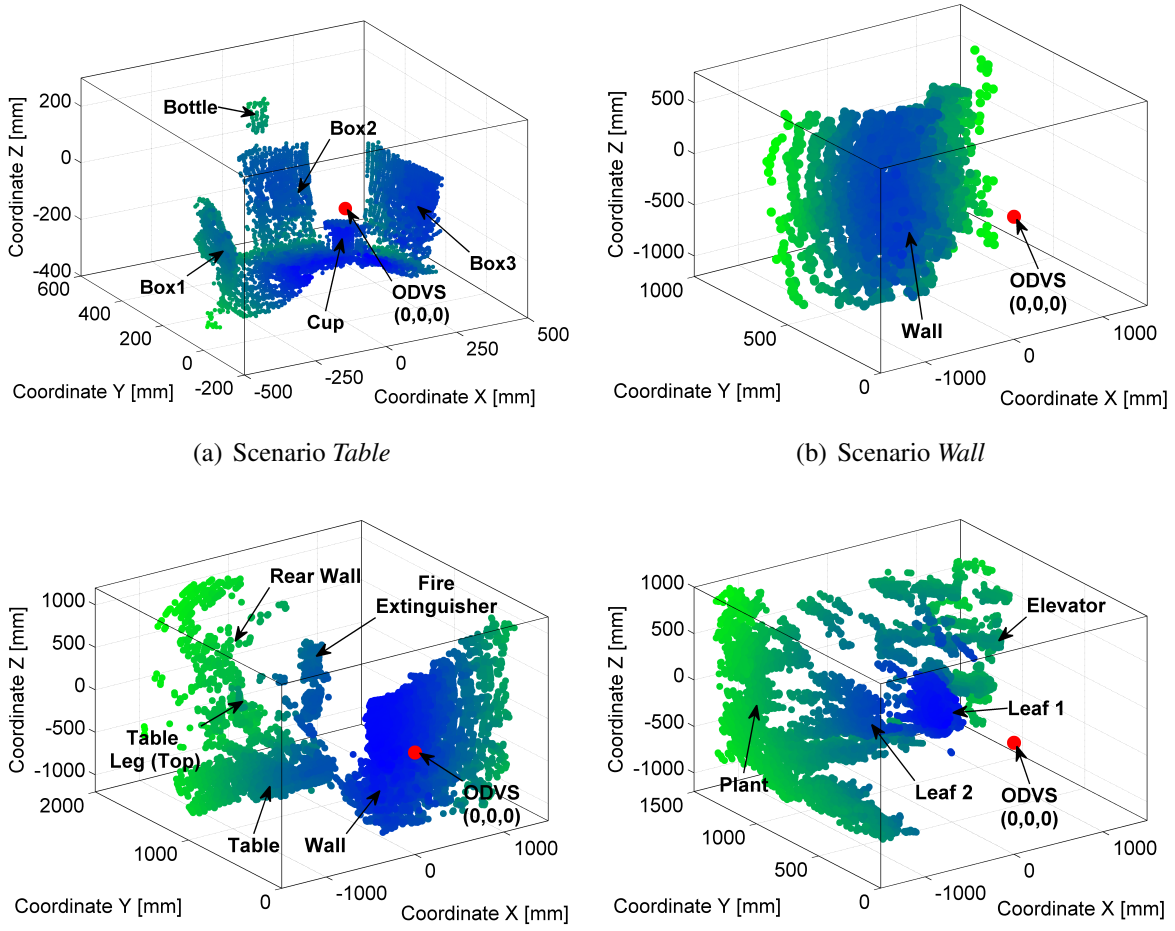


Figure 4.24: Dynamic programming (left) vs. semi-global matching (right). Streaking effects occur in low-textured regions such as the floor regions for all scenarios. Semi-global matching overcomes the streaking effects by optimizing path costs in all directions equally.

Ground-truth data

Further experiments were conducted to compare the location of bounding boxes obtained from the disparity maps and triangulation with ground truth data. The scenarios *Wall* and *WallTable* were chosen as test scenarios, since they are similar to potential parking situations. The first scenario represents a parking scenario where a car is parked close to a wall (see Figure 4.26(a)). A more complex environment was chosen for the second parking scenario (see Figure 4.26(b)) where a car is parked close to a wall and other objects. Ground truth data for both scenarios were generated in order to evaluate the accuracy of the position of the bounding boxes. Figure 4.26(c) and Figure 4.26(d) illustrate ground truth data for the parking scenarios. Bounding boxes are generated along arbitrary scanlines and are compared with the corresponding ground truth data. Figure 4.26(a) and Figure 4.26(b) show three arbitrary scanlines for the parking scenarios. The



(a) Scenario *Table*

(b) Scenario *Wall*

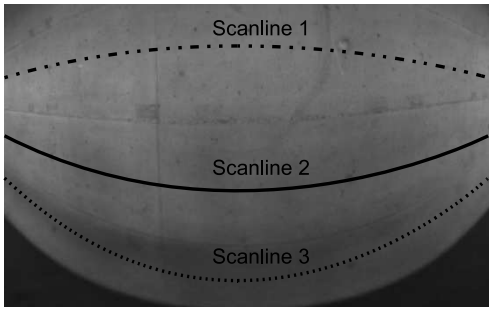
(c) Scenario *WallTable*

(d) Scenario *Plant*

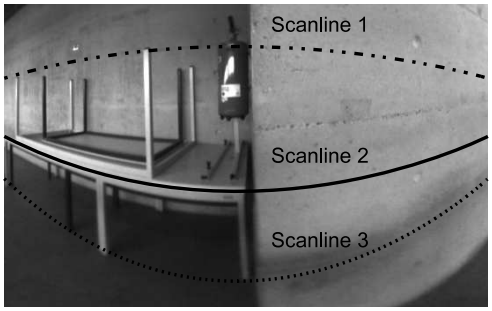
Figure 4.25: Scenario and resulting bounding boxes for the scenarios *Table* (a), *Wall* (b), *WallTable* (c) and *Plant* (d).

scanlines represent a 2D cut through the real scenario and also a cut through the computed bounding boxes. For all scenarios, the origin of the world coordinate system is located in the center of the omnidirectional camera system at $(0, 0, 0)$ and the camera is placed at the lowest position of the camera platform.

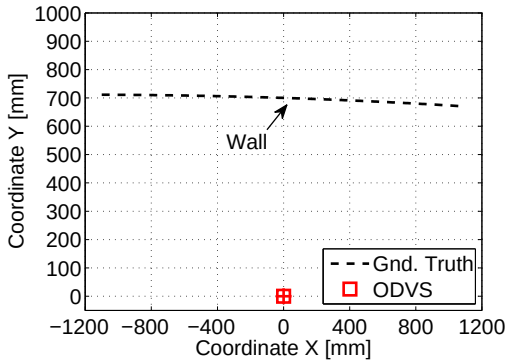
Figure 4.27 illustrates the locations of bounding boxes computed for the scenarios *Wall* and *WallTable*. The locations of the bounding boxes matches the ground truth data for scanline one and scanline two in the first scenario (see Figure 4.27(a) and Figure 4.27(b)). The position error of bounding boxes increases at scanlines in the lower regions of panoramic images. These regions are noisy and less textured than the upper regions so that wrong disparities and wrong 3D locations of bounding boxes are determined. Figure 4.27(c) displays bounding boxes that are computed for the third scanline in scenario *Wall*. For the second scenario, Figure 4.27(d) and Figure 4.27(e) illustrate the locations of bounding boxes for scanline 1 and scanline 2. It can be seen that the bounding boxes are obtained for disparities found at the top of the front



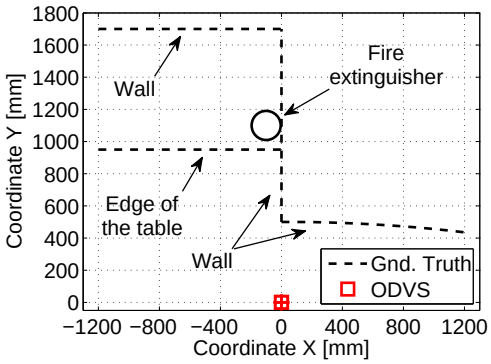
(a) Scenario 1 and scanlines



(b) Scenario 2 and scanlines



(c) Ground Truth



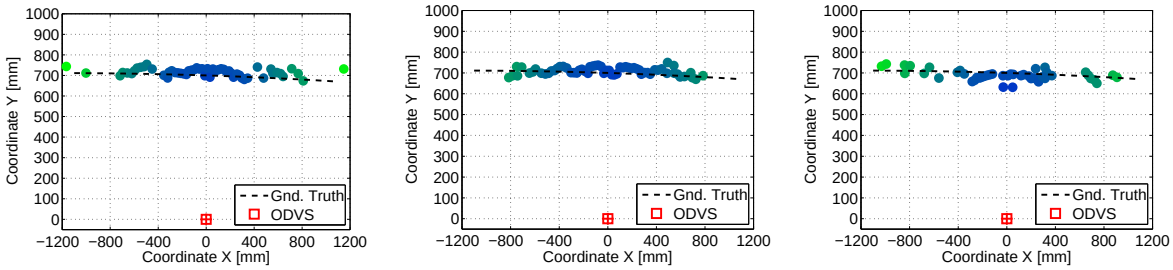
(d) Ground Truth

Figure 4.26: Scenario *Wall* (a) and scenario *WallTable* (b) and the chosen scanlines used to compare the bounding boxes with ground truth data of the environment (c,d).

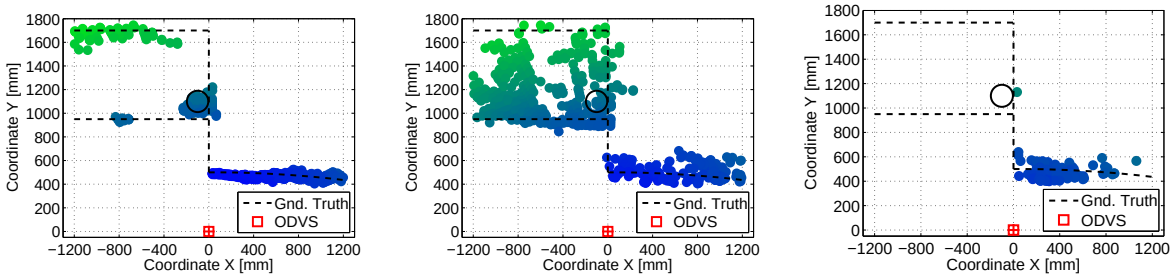
table leg, for the fire extinguisher and for the rear wall (see Figure 4.27(d)). The bounding boxes generated for scanline 2 model the surface of the table and the front wall and match the ground truth data (see Figure 4.27(e)) very well. Figure 4.27(f) illustrates the bounding boxes obtained from the third scanline. The locations of the boxes are computed and model the front wall. A direct comparison with the ground truth data shows a position error up to 10 cm for the bounding boxes obtained from disparities in lower image regions.

Bounding box refinement

In Section 4.5.3, a method has been proposed to refine the bounding boxes of objects whose surface is perpendicular to the ground. Therefore, 3D information is determined from the edge appearing in an image when an object is in contact with the floor. Figure 4.28(a) and Figure 4.28(d) show the detected edges (solid line) for scenario *Wall* and scenario *WallTable* within a specified region (dashed line). The refinement stage uses this information to remove outliers and inaccuracies in the generated bounding boxes. Figure 4.28 illustrates the bounding boxes for the scenarios *Wall* and *WallTable*. These boxes are generated by means of the stereo algorithm and contain many outliers and noise. The refinement stage can be used for improving the bounding boxes by removing outliers and noise. Figure 4.28(c) and Figure 4.28(f) illustrate the result after the refinement stage.



(a) Scen. 1, Scanline 1 (b) Scen. 1, Scanline 2 (c) Scen. 1, Scanline 3



(d) Scen. 2, Scanline 1 (e) Scen. 2, Scanline 2 (f) Scen. 2, Scanline 3

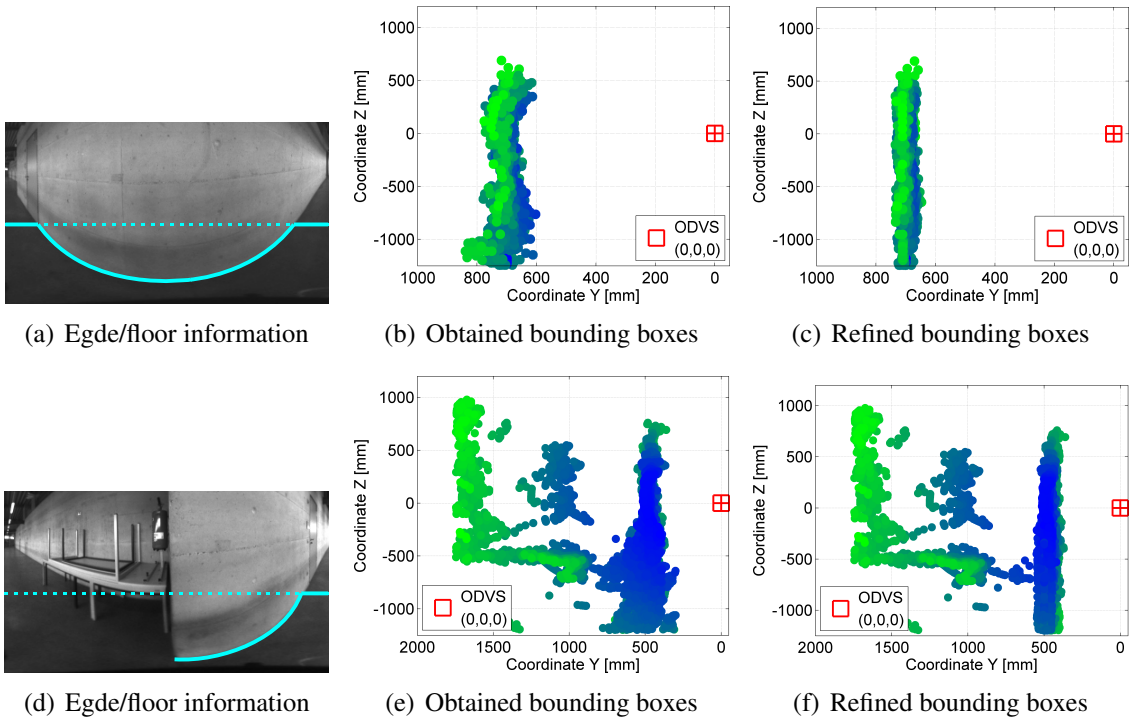
Figure 4.27: Bounding boxes obtained at the scanlines for the scenarios *Wall* and *WallTable* and compared with ground the truth data.

Scenario	No Refinement	Refinement
Wall	$45 \pm 31mm$	$21 \pm 11mm$
Wall Table	$157 \pm 47mm$	$78 \pm 29mm$

Table 4.2: Refinement result

It can be seen, that the refinement stage leads to good scene approximations for the scenario *Wall*. However, the refinement stage only improves the bounding boxes of objects that touch the floor and that are perpendicular to the ground. This is guaranteed by restricting the search area $(x \pm \Delta, y \pm \Delta)$ around objects suitable for the refinement stage (see Section 4.5.3). The bounding boxes of the front wall in scenario *WallTable* are suitable for refinement only, whereas all other bounding boxes remain unchanged – i.e. the bounding boxes of the table and the fire extinguisher (see Figure 4.28(f)).

Further experiments have been conducted to compare the accuracy of improved bounding boxes with the accuracy of unchanged bounding boxes. The scenarios *Wall* and *WallTable* were chosen as benchmarks and the results obtained were compared to ground truth data. Table 4.2 illustrates the mean position errors and the standard deviations for the benchmark scenarios. It can be seen that the refinement stage reduces the mean position errors for both the scenarios *Wall* and *WallTable*. However, the refinement stage must not improve bounding boxes that represent the rear wall, the table and the fire extinguisher. This leads to a higher mean localization error in scenario *WallTable* than in scenario *Wall* across all bounding boxes. However, outliers and disturbances in the front wall of scenario *WallTable* are also detected and removed by means of the refinement stage.



(a) Egde/floor information (b) Obtained bounding boxes (c) Refined bounding boxes

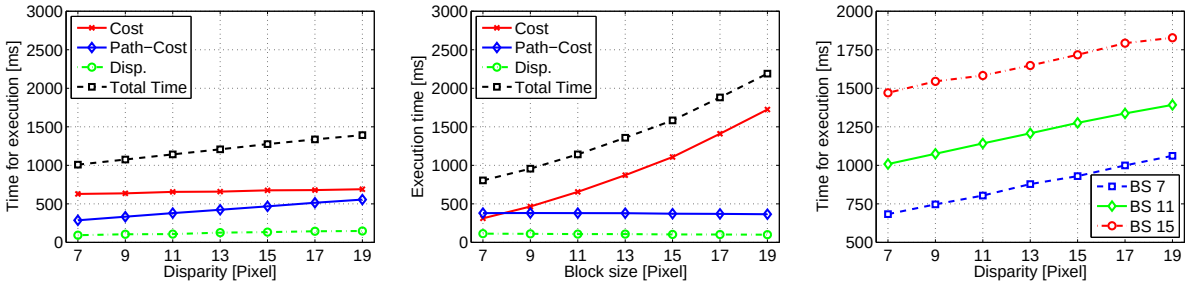
(d) Egde/floor information (e) Obtained bounding boxes (f) Refined bounding boxes

Figure 4.28: Unchanged and refined bounding boxes of the scenarios *Wall* and *WallTable* that obtained by a stereo algorithm and improved by an additional refinement stage.

Execution time

In a last experiment, the time for execution has been determined for the proposed ambiance detection algorithm. Figure 4.29 illustrates the execution time for disparity map generation by means of the semi-global matching stereo algorithm. The disparity map generation process is subdivided into three stages. In a first step, matching costs based on rank transformation followed by a pixel by pixel difference computation are determined. Secondly, the path costs for the semi global matching algorithm are computed. In a last step, the path costs are summed up to determine the minimum path costs for disparity computation. Figure 4.29(a) illustrates the execution time for computing rank-based matching costs and for determining path costs and disparity maps over the numbers of disparities for fixed search windows (size 11×11 pixels). It can be seen that the execution time for computing matching costs slowly rises when increasing the disparity range, but strongly rises when increasing the size of the search windows for constant disparities $\delta = 11$ (see Figure 4.29(b)). Only changes in the disparity ranges lead to changes in the execution times for path costs and disparity computation. By contrast, the execution time for path costs and disparity computation is independent of the window size for matching cost computation (see Figure 4.29(b)) with regard to constant disparities δ.

Figure 4.29(c) illustrates the total execution time for computing disparity maps over the disparity range. These execution times are determined by using different block sizes for matching cost computation. Table 4.3 gives a brief overview of the total execution time for the entire ambiance reconstruction process.



(a) Fixed block size 11×11 pixels (b) Fixed disparity $\delta = 11$ pixel (c)

Figure 4.29: Execution time for the semi-global matching-based stereo algorithm (SGM) for increasing disparities (a) and increasing the size of search windows for matching cost computation (b). Total execution time for the SGM-based stereo algorithm using different search windows over increasing disparities (c). Legend: *Cost*: matching cost computation, *Path*: path cost computation, *Disp*: Summed path costs and disparity computation, *BS* block size of search windows.

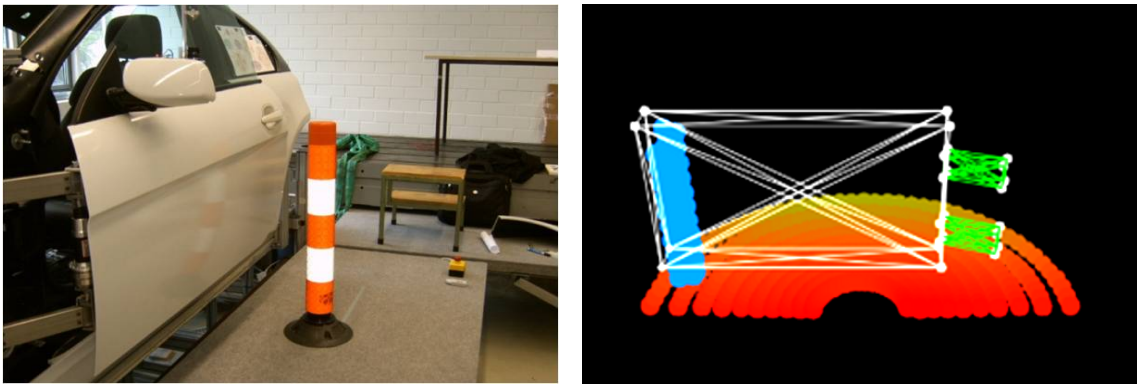
Task	Execution Time [ms]
Camera Motion $(35mm)$	412ms
Pose Refinement, Rectification	227 ms
Disparity Map Generation, BS: 11×11 pix, Disp.: $\delta = 19$ pix.	1425 ms
Bounding Boxes (2500)	242 ms
Total Time	2306 ms

Table 4.3: This table illustrates the total execution time for reconstructing the ambiance close to the car door.

Finally, Figure 4.30 shows the bounding boxes (ambiance information) of a test scenario where a pillar is located next to the car door. The ambiance information is represented by the blue spherical bounding boxes, whereas ground information is modeled by red bounding boxes. This figure also illustrates the wire-frame model of the smart car door.

4.8 Conclusion

This chapter presents a new stereo application in the automotive domain for generating 3D ambiance information in a smart car door. This application will prevent collisions with static obstacles next to the car door when opening the car door. For this purpose, an omnidirectional camera is integrated with the side-view mirror of a car and monitors the surroundings in theirs entirety. 3D ambiance information is obtained by means of a single omnidirectional camera and a motion-stereo algorithm. While other stereo-based applications such as telepresence require detailed 3D scene models, a rough modeling of the environment based on bounding boxes is sufficient for safely performing door operations. The key problem addressed in this chapter is the generation of solid 3D ambiance information from low-textured and low-resolution



(a) Test Scenario (b) Generate Ambiance Information

Figure 4.30: Real-life testing scenario (a) and 3D ambiance information obtained from the proposed stereo algorithm (b). Obstacles are represented by the blue bounding boxes, and the ground is represented by the red bounding boxes.

panoramic images.

First, this chapter introduces the fundamentals of stereo vision using omnidirectional cameras and presents a parallel panoramic configuration allowing for a correspondence search along 1D epipolar lines. In contrast to stereo setups with perspective cameras, two epipoles exist in stereo panoramic images captured by stereo setups with panoramic cameras. For stereo setups with horizontally arranged cameras, panoramic images do not preserve straight epipolar lines, and epipolar curves are obtained instead. However, epipolar lines can be obtained when using a stereo setup with vertically arranged omnidirectional cameras. This setup is called a parallel panoramic configuration. A mechanical device is attached to the side-view mirror and positions the camera vertically to generate a vertical stereo configuration. The mechanical device is also equipped with a position sensor for determining the translation between two camera positions. Inaccuracies in the positions due to clearances in the mechanical device cannot be detected with the sensor. Therefore, an egomotion estimation algorithm is described for refining the relation between two camera poses using image correspondences only. Based on the relation between two camera poses, this chapter proposes a method for rectifying pairs of panoramic images to obtain a parallel panoramic configuration.

Many low-textured objects such as white walls or flower boxes exist in typical parking scenarios. For this reason, the stereo algorithm must be suitable for generating disparity maps from low-textured panoramic images. Therefore, the semi-global matching stereo algorithm is presented that is suitable for producing dense disparity maps even from pairs of low-textured and low-resolution panoramic images. Other stereo algorithms, i.e. based on dynamic programming, use strong stereo constraints in one search direction but none or weak constraints in other search directions. Semi-global matching, however, aggregates matching costs across the whole panoramic image and considers stereo constraints for all search directions equally. For this reason, the semi-global matching stereo algorithm can compute dense disparity maps even from poorly textured regions in panoramic images.

Based on the disparity maps, 3D ambiance information in the form of bounding boxes is generated using triangulation. But these disparity maps contain inhomogeneities and outliers leading to wrongly determined positions of the bounding boxes. In particular, low-textured regions and image noise lead to many wrongly determined disparities and hence to wrongly computed locations of bounding boxes. These disturbances in the bounding boxes must be removed in order to increase the robustness of ambiance modeling. For this reason, a refinement stage is introduced in Section 4.5.3 to remove outliers in the bounding boxes. Outliers are detected using edge/floor information from objects that touch the floor and whose surfaces are perpendicular to the floor. Experiments demonstrated a good modeling surroundings next to the car door with a single omnidirectional camera and the motion-stereo algorithm.

Additionally, a new method is proposed to determine the position error of 3D-data depending on the quantization error (see Section 4.6.1) and the calibration error (see Section 4.6.2) of omnidirectional cameras. For both the quantization and calibration error, distance error intervals are computed for analyzing the accuracy of 3D data obtained from omnidirectional camera-based stereo setups. In literature, the geometry of the imaging device is adapted to the properties of the cylindric projection in order to obtain best resolution of panoramic images. For this reason, accuracy analysis of 3D data obtained from panoramic stereo setups is limited to the cylindric projection. In this thesis, a new study is presented that analyzes the accuracy of 3D data generated from cylindrical, conical and spherical panoramic images using distance-based calibration and quantization error intervals (see Section 4.7.1). The ratio between the calibration and quantization error intervals can be used as a measurement value to evaluate projections in terms of accuracy and usability of stereo setups. Finally, the measurement ranges and blind zones are analyzed for cylindrical, conical and spherical panoramic images.

5 Concluding remarks

In this thesis, a smart car door system has been presented that consists of an actuated two-hinge kinematic system allowing for situation-dependent door opening in tight parking lots. The smart car door system also includes a device to adjust the car seat according to the driver's body height. Therefore, a sensor subsystem provides ambiance information and body heights of approaching drivers for the control unit. The sensor system used for this application consists of two omnidirectional cameras, one attached to each side-view mirror of the car.

This thesis focuses on the sensor subsystem of the smart car door and on algorithms for body height estimation and ambiance monitoring. First, methods and techniques were presented to transform original images obtained by omnidirectional cameras into panoramic images. These methods are also suitable for evaluating panoramic images in terms of applicability for specific camera/mirror configurations. Additionally, novel methods were proposed to robustly extract approaching drivers in low-resolution images and to measure their absolute body heights using a single omnidirectional camera only. Finally, this thesis presented a new application that uses a motion stereo-based algorithm to generate 3D-information of the surroundings next to the door. This chapter summarizes the contributions of the thesis and outlines some directions for future research and future work.

Chapter 2 introduced the geometry of omnidirectional cameras and described the properties of central projection cameras, viz. omnidirectional cameras with a single point of view. The single point of view enables the generation of perspective panoramic images from original images to facilitate the determination of geometric properties of objects in a scene. Moreover, the single point of view property is a prerequisite for applying the known epipolar geometry to omnidirectional cameras to generate true 3D information of a scene. This chapter described the mathematical formalism of the omnidirectional camera model and presented a calibration process. A toolbox for calibrating omnidirectional cameras required manually selected chessboard corners to obtain the intrinsic and extrinsic parameters of omnidirectional cameras. However, the calibration of (omnidirectional) cameras must be performed automatically in the automotive domain: Therefore, this thesis presented an extension to the current calibration process to automatically detect chessboard corners in calibration images. The proposed algorithm is suitable for extracting calibration patterns in low resolution images and was tested with images taken under different illumination conditions. It is shown through a large number of experiments that the proposed algorithm is strongly robust against illumination changes and noise.

Additionally, the pixel density was introduced as a measurement value to compare different camera configurations and to evaluate different projections used for image transformation. The pixel density indicates the distribution and the utilization of sensor pixels in panoramic images. The best utilization of sensor pixels is achieved by values of the pixel density close to unity.

(a) Constant Gain Mirror (b) Resolution Invariant Mirror

Figure 5.1: Images of omnidirectional cameras using optimized mirrors [166].

This means a direct mapping of intensity values of sensor pixels to the corresponding pixel positions in panoramic images. A pixel density less than unity denotes poor resolution since the intensity of a sensor pixel is required at several pixel positions. Larger values than unity denote a waste of sensor pixels since intensities of many sensor pixel positions are projected onto the same pixel position in the panoramic image. The pixel density and its characteristics can help to adjust the projection type and the projection parameters to obtain best utilization of sensor pixels in panoramic images. This chapter also demonstrates that the commonly used cylindric projection is not the best projection for image transformation due to the large variance of the values of the pixel density across the projection area.

Future directions: The calibration process described in this chapter requires calibration patterns to determine the extrinsic and intrinsic parameters of the camera. An interesting direction to explore would be the design of a self calibration algorithm that extracts features such as lines and corners from the surroundings of the camera to determine the calibration parameters. Examples of such features include road markings or arbitrary features, e.g. of passing cars. Additional information such as the mounting position on the car or the speed together with the epipolar constraint for omnidirectional cameras would lead to precise calibration results. A self calibration would also overcome costly additional calibration stages in car manufacturing processes. First research addressing these issues is presented by Geyer *et al.* [113] and could be extended to design a robust self-calibration algorithm to be used in the domain of automotive system engineering.

In this thesis, a hyperbolic mirror was used for the omnidirectional camera. Hyperbolic mirrors lead to varying pixel densities over the transformed panoramic images and, hence, to a varying resolution in panoramic images. Another interesting direction to explore would be the development of optimized mirrors to obtain constant pixel densities and constant resolutions across panoramic images. Such mirrors were proposed in [166] and would lead to a better utilization

of sensor pixels in panoramic images (see Figure 5.1). The proposed pixel density could be used for computing optimized mirrors particularly targeting the need for cameras in the automotive domain. Additionally, correspondence search and distance estimation could be facilitated by using panoramic images with approximately constant resolutions.

Chapter 3 describes a novel method for estimating the body height of approaching drivers to automatically adjust the driver seat position. An adjusted seat position increases the comfort for ingress and egress in tight parking lots. This chapter focuses on robust extraction of approaching drivers in low-resolution panoramic images and on absolute body height estimation using a single omnidirectional camera. A Kalman-based background model is presented that separates foreground objects from background and detects approaching drivers. A background model is suitable for extracting complete regions even in low-contrast panoramic images. Therefore, it has been chosen instead of alternative detection methods such as optical flow or template matching. In particular, drivers that are far away from the car occupy only few pixels on each video frame and are, thus, not easy to differentiate from the background.

Optical flow is a powerful tool for detecting moving objects, but it cannot detect static objects such as standing drivers. Moreover, it is difficult to detect approaching drivers in our setup due to poor contrast and poorly textured regions in low-resolution panoramic images. The use of optical flow would then lead to incomplete image regions containing the drivers. These regions, however, must be determined precisely since they serve as input for the height estimation algorithm. Foot and head points of approaching drivers are identified in these regions to estimate the body heights.

The background algorithm is extended to record background images in parking scenarios with high traffic volumes. This chapter also presents robust shadow detection especially targeting shadow elimination in gray-scaled images and compensation of illumination changes. The position of a camera relative to the ground is the key feature for absolute body height estimation using a single camera only. This camera position varies for each parking scenario and must be determined from image data only. Camera position estimation is based on a set of extracted foot and head points that is recorded from a driver as he/ she approaches. In Chapter 3, a novel, model-based camera-ground function is proposed that models the relation between the camera and the ground based on image data from extracted head and foot points of approaching drivers. This function explicitly considers camera tilt caused by inclined parked cars and has a global minimum when the estimated camera position relative to the ground best matches the real camera position. With the estimated camera position, the absolute body height of approaching drivers can be computed from the sets of foot and head points.

Experiments demonstrated a good extraction of approaching drivers in panoramic images even if they were far away from the car. The proposed shadow detection and illumination compensation algorithm proved to be a powerful extension for the precise extraction of drivers in low-resolution panoramic images. Further experiments showed a good estimation of the camera position relative to the ground. Moreover, the body height of approaching drivers could be estimated with an accuracy of up to $3cm$ to $4cm$ using several sets of foot and head points.

Future directions: The proposed algorithm uses foot and head points of approaching drivers for estimating their body heights. The limitation of this technique is the influence of high-heeled shoes or large hair styles on the body height estimation. In other words, it is not possible to extract real foot points of approaching drivers when they wear high-heeled shoes. One way to overcome this limitation is using an additional anthropometric model that explicitly extracts and considers body proportions such as the length of legs, arms and torso. Such anthropometric models are commonly used in the domain of ergonomics and in performance analyses of athletes in sports and athletics.

Chapter 4 presents a new stereo application with a single omnidirectional camera and for generating 3D ambiance information of the surroundings next to a car door. The 3D information serves as input to a control unit that computes opening paths to avoid collisions when opening the door. In contrast to other applications such as telepresence that require detailed 3D scene models, a rough model of the surroundings is sufficient to safely open a car door.

This chapter introduces the fundamentals of stereo vision with omnidirectional cameras. In this application, the omnidirectional cameras are vertically arranged to each other to obtain a parallel panoramic configuration. The parallel panoramic configuration enables a correspondence search along 1D epipolar lines. Therefore, a mechanical device is attached to the side-view mirror of the car and vertically positions the camera to obtain a stereo setup. The mechanical device is equipped with a position sensor to determine the vertical translation between two camera positions. Small rotations and additional translations caused by mechanical clearances cannot be detected with the sensor. Therefore, an egomotion estimation algorithm is described that refines the relation between two camera poses using image correspondences only. With the known relation, pairs of panoramic images can be rectified to obtain a parallel panoramic configuration for facilitating the correspondence search.

The semi-global matching algorithm is presented that is suitable for generating dense disparity maps from pairs of low-textured and low-resolution panoramic images. Based on these disparity maps, 3D ambiance information – in form of bounding boxes – is produced using triangulation. Moreover, a refinement stage is introduced to remove outliers in the 3D data by using edge/floor information of objects that touch the floor and whose surface is perpendicular to the ground. Chapter 4 also describes a new method to determine the position error of 3D-data. This position error depends on the calibration error of omnidirectional cameras and on the quantization error incurred by the resolution of panoramic images. In a further study, the measurement ranges and blind zones of stereo setups with omnidirectional cameras are analyzed. The measurement ranges and blind zones depend on the baseline length and on the projection in panoramic images. Further experiments demonstrated a good modeling of the surroundings next to the car door with an omnidirectional camera-based stereo setup.

Future directions: Stereo setups providing a parallel panoramic configuration have very large dimensions due to large packages of today's omnidirectional cameras and are, hence, not easy to integrate into vehicles. A new design of omnidirectional cameras – such as folded omnidirectional cameras – would reduce the dimension of the camera packages and lead to smaller stereo setups and would also fulfill the needs of automotive system design. Figure 5.2(a) displays a folded omnidirectional camera [168] that could be used as an alternative to standard omnidirec-

(a) Folded ODVS (b) Mirror and Lens (c) Example Image

Figure 5.2: Folded omnidirectional camera [168] (a). Stereo configuration with a single omnidirectional camera and a lens (b) and the resulting stereo image [169] (c).

tional cameras. In this device, the perspective camera is integrated within the hyperbolic mirror. An extra mirror reflects the light from the hyperbolic mirror into the perspective camera.

In this thesis, a mechanical device has been attached to the side-view mirror to vertically position the camera. In this manner, a stereo setup based on motion stereo has been obtained. When the camera is integrated with the side-view mirror of a car, the required camera motion could be provided by a fold-in and fold-out movement of the side-view mirror to obtain a stereo setup. Moreover, several images could be captured during the fold-in and fold-out movement of the mirror in order to obtain a multi-baseline stereo system. The relation between a reference image and the other images captured, which are captured at different poses, could be estimated from image data and from known motion paths. Another interesting direction to explore would be the use of stereo systems with single omnidirectional cameras. Yi *et al.* [169] proposed an omnidirectional camera that is equipped with an extra lens to obtain two views of a scenario. Figure 5.2(b) shows the proposed camera system and Figure 5.2(c) illustrates a stereo image captured by this camera. This camera system was not suitable for this application due to its very poor resolution, but the increasing resolution of common camera sensors makes such system more and more attractive for stereo applications in the automotive domain.

Bibliography

[1] Q. Strolz, M.and Mühlbauer, C. Scharfenberger, G. Färber, and M. Buss. Towards a generic control system for actuated car doors with arbitrary degrees of freedom. In *In Procedings of IEEE Intelligent Vehicles Symposium (IV)*, pages 391 –397, Eindhoven, Netherlands, june 2008.

[2] M. Fischer, S. Braun, D. Hellenbrand, C. Richter, O. Sabbah, C. Scharfenberger, M. Strolz, and G. Färber. Multidisciplinary developement of new door and seat concepts as part of an ergonomic ingress/egress support system. In *FISITA 2008 World Automotive Congress*, Munich, Germany, 2008.

[3] C. Scharfenberger, F. Böhm, and G. Färber. Image rectification: Evaluation of various projections for omndirectional vision sensors using the pixel density. In *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, Lisbon, Portugal, feb. 2009.

[4] C. Scharfenberger, M. Fischer, S. Braun, D. Hellenbrand, C. Richter, O. Sabbah, M. Strolz, and G. Färber. Multidisziplinäre Entwicklung von neuen Türkonzepten als ein Teil einer ergonomisch optimierten Ein/Ausstiegsunterstützung. In *VDI-OEM Forum Fahrzeugtüren und -klappen*, Sindelfingen, Germany, may 2009.

[5] C. Scharfenberger, S. Chakraborty, and G. Färber. Robust image processing for an omnidirectional camera-based smart car door. In *Proceedings of the 7th IEEE Workshop on Embedded Systems for Real-time Multimedia (ESTIMedia)*, Grenoble, France, oct. 2009.

[6] C. Scharfenberger. Multi disciplinary development of novel door concepts. In *Invited Speaker at the International Automotive Conference on Doors and Closures in Car Body Engineering*, Bad Nauheim, Germany, nov. 2009.

[7] C. Scharfenberger and G. Färber. Robuste Erkennung von Verkehrsteilnehmern zur Vermeidung von Unfällen beim Türöffnen für ein Fahrerassistenzsystem. In *4. Tagung Sicherheit durch Fahrerassistenzsysteme, TUEV-Sued*, Munich, Germany, apr. 2010.

[8] C. Scharfenberger, S. Chakraborty, and G. Färber. Driver body-height prediction for an ergonomically optimized ingress using a single omnidirectional camera. In *20th International Conference on Pattern Recognition (ICPR)*, pages 298 –301, Istanbul, Turkey, aug. 2010.

[9] C. Scharfenberger, P. Hoffmann, J. Leupold, and G. Färber. Vorrichtung zum Verhindern einer Verklemmung eines Fremdkörpers in einem Fahrzeugzugang. In *German Patent Disclosure, No. DE 102009040994 A1*, apr. 2010.

[10] C. Scharfenberger, M. Eibl, J. Leupold, and G. Färber. Vorrichtung zum Vermessen einer sich einem Fahrzeug annähernden Person. In *German Patent Disclosure, No. DE 102009040995 A1*, apr. 2010.

[11] C. Scharfenberger, F. Böhm, J. Leupold, and G. Färber. Vorrichtung zur Überwachung eines Raumbereiches ausserhalb eines Fahrzeuges. In *German Patent Disclosure, No. DE 102009057336 A1*, july 2010.

[12] M. Almstätter. Design und prototypische Realisierung einer positionsgeregelten Kameraplattform für ein optisches Sensorsystem. Master's thesis, TU Munich, Institute for Real-Time Computer Systems, aug. 2007.

[13] T. Friederich. Bewegungsdetektion anhand eines adaptierten Referenzbildes mit Kompensation der Kameraeigenbewegung. Studienarbeit, TU Munich, Institute for Real-Time Computer Systems, feb. 2008.

[14] F. Böhm. Tiefenschätzung mit omnidirektionalem Kamerasystem. Master's thesis, TU Munich, Institute for Real-Time Computer Systems, apr. 2008.

[15] D. Carton. Fahrzeugtracking und Gefahreneinschätzung unter Verwendung eines omnidirektionalen Kamerasystems. Studienarbeit, TU Munich, Institute for Real-Time Computer Systems, oct. 2008.

[16] L. Hong. Vollautomatische Detektion von Schachbrettmustern zur Kalibrierung omnidirektionaler Kamerasysteme. Bachelor's thesis, TU Munich, Institute for Real-Time Computer Systems, oct. 2008.

[17] M. S. Trigui. Person Height Measurement with Unknown Camera Orientation to the Ground. Master's thesis, TU Munich, Institute for Real-Time Computer Systems, apr. 2009.

[18] J. Huang. Weiterentwicklung und Verbesserung einer automatischen Kalibration für omnidirektionale Kameras., sep. 2009.

[19] A.-O. Sabbah. *Entwicklung eines dynamischen Diskomfortmodell am Beispiel des Ein- und Ausstiegs*. PhD thesis, Lehrstuhl für Ergonomie – Technische Universität München, 2010.

[20] C. Richter. *Ein Kinematikentwurfsarbeitsplatz für Fahrzeugtüren*. PhD thesis, Lehrstuhl für Mikro und Medizingerätetechnik – Technische Universität München, 2011.

[21] M. Fischer. *Anwendungsspezifische Kinematikentwicklung unter Einsatz modularer Prototypen*. PhD thesis, Lehrstuhl für Mikro und Medizingerätetechnik – Technische Universität München, 2010.

[22] M. Strolz. *Control, Simulation and Evaluation of Actuated Car Doors*. PhD thesis, Institute of Automatic Control Engineering – Technische Universität München, 2010.

[23] M. Murphy. Passenger car driver assistance systems, technologies and trends to 2015. Technical Report, Automotive World News, 2007.

[24] T. Ehlgen, T. Pajdla, and D. Ammon. Eliminating blind spots for assisted driving. *IEEE Transactions on Intelligent Transportation Systems*, 9(4):657–665, Dec 2008.

[25] L. Matuszyk, A. Zelinsky, L. Nilsson, and M. Rilbe. Stereo panoramic vision for monitoring vehicle blind-spots. In *Proceedings of the Intelligent Vehicles Symposium (IV)*, june 2004.

[26] D. Scaramuzza, A. Martinelli, and R. Siegwart. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *Proceedings of the Conference on Computer Vision Systems (ICVS)*, jan. 2006.

[27] A. Martinelli, D. Scaramuzza, and R. Siegwart. A toolbox for easily calibration omnidirectional cameras. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, oct. 2006.

[28] J. P. Barreto and K. Daniilidis. Unifying image plane liftings for central catadioptric and dioptric cameras. In *Proceedings of the Workshop on Omnidirectional Vision (OMNIVIS)*, may 2004.

[29] B. Micusik. *Two wiew geometry of omnidirectional Cameras*. PhD thesis, Center for Machine Perception, Czech Technical University in Prague, 2004.

[30] D. Scaramuzza. *Omnidirectional vision: from calibration to robot motion estimation.* PhD thesis, ETH ZURICH, 2008.

[31] S. Nayar and S. Baker. Catadioptric omnidirectional camera. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, june 1997.

[32] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* 2004.

[33] R. Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. 1986.

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

[35] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[36] S. Derrien and K. Konolige. Approximating a single viewpoint in panoramic imaging devices. In *Proceedings of the IEEE Workshop on Omnidirectional Vision (OMNIVIS)*, june 2000.

[37] M. D. Grossberg and S. K. Nayar. A general imaging model and a method for finding its parameters. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2001.

[38] R. Swaminathan, M. D. Grossberg, and S. K. Nayar. Caustics of catadioptric cameras. In *Proceedings of the International Conference on Computer Vision (ICCV)*, july 2001.

[39] D. Strelow, J. Mishler, D. Koes, and S. S. Precise omnidirectional camera calibration. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, dec. 2001.

[40] R. Descartes and D. Smith. *The geometry of Rene Descartes.* Dover, New York, originally published in Discours de la Methode, 1637.

[41] R. P. Feynman, R. Leighton, and M. Sands. *The Feynman Lectures on Physics*. Addison-Wesley, 1 ed., 1963.

[42] D. W. Rees. Panoramic television viewing system, 1970. US Patent US3505465.

[43] H. Ishiguro. *Development of Low-Cost Compact Omnidirectional Vision Sensors and their applications.* Springer, 1998.

[44] S. Baker and S. Nayar. A theory of catadioptric image formation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, jan. 1998.

[45] S. Baker and S. Nayar. A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35:175–196, 1999.

[46] C. A. Poynton. *Digital Video and HDTV: Algorithms and Interfaces.* 2003.

[47] J. Canny. A computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[48] D. B. Williams and V. Madisetti. *Digital signal processing handbook.* 1999.

[49] A. Waibel and K. F. Lee. *Readings in Speech Recognition*. 1990.

[50] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference (AVC)*, apr. 1988.

[51] K. Levenberg. A method for the solution of certain problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

[52] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.

[53] Burger W. und Burge, M. J. *Digitale Bildverarbeitung.* Springer Verlag Berlin, Heidelberg, New York, 2., überarb. Aufl. 2006.

[54] C. Ishii, Y. Sudo, and H. Hashimoto. An image conversion algorithm from fish eye image to perspective image for human eyes. In *Proceedings of the International Conference on Advanced Intelligent Mechatronics (AIM)*, july 2003.

[55] J. Fernandes, M. Ferreira, J. Neves, and C. Couto. Fast correction of lens distortion for image applications. In *Proceedings of the International Symposium on Industrial Electronics (ISIE)*, july 1997.

[56] W. Yu and Y. Chung. An embedded camera lens distortion correction method for mobile computing applications. In *Proceedings of the International Conference on Consumer Electronics (ICCE)*, dec. 2003.

[57] M. Rufli, D. Scaramuzza, and R. Siegwart. Automatic detection of checkerboards on blurred and distorted images. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, sep. 2008.

[58] D.S. Breed and W.E. DuVall. Automatic vehicle seat adjuster, 1998. US Patent 5.748.473.

[59] K. Kamijo and H. Nagashima. Seat position automatic adjusting device for an automotive vehicle, 1983. US Patent 4.401.928.

[60] S. Saito, S. Kanai, et al. Height adjusting device for automotive seat, 1989. US Patent 4.834.333.

[61] M.D. White and R. Freund. Horizontal seat position adjuster, 1993. US Patent 5.222.402.

[62] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and Cybernetics*, 34:334–352, 2004.

[63] H.-Y. Shum. Interactive construction of 3d models from panoramic mosaics. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, june 1998.

[64] R. Cipolla, T. Drummond, and D. Robertson. Camera calibration from vanishing points in images of architectural scenes. In *Proceedings of the British Machine Vision Conference (BMVC)*, sep. 1999.

[65] R. L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[66] R. Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice.* 2009.

[67] O. Achler and M. M. Trivedi. Real-time traffic flow analysis using omnidirectional video network and flatplane transformation. Computer Vision and Robotics Research Laboratory - University of California, San Diego, 2002.

[68] T. Ghandi and M. M. Trivedi. Motion based vehicle surround analysis using an omnidirectional camera. In *Proceedings of the Intelligent Vehicle Symposium (IV)*, june 2004.

[69] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: Who? when? where? what? a real-time system for detecting and tracking people. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition (FGR)*, apr. 1998.

[70] N. Friedman and S. Russel. Image segmentation in video sequences: A probalistic approach. In *Proceedings of the International Conference on Uncertainity in Artificial Intelligence (UAI)*, aug. 1997.

[71] K. P. Karmann and A. von Brandt. *Moving object recognition using an adaptive background memory*, volume 2. 1990.

[72] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, 2002.

[73] D. Stenger, V. Ramesh, N. Paragios, F. Coetzec, and J. M Buhmann. Topology free hidden markov models: application to background modeling. In *Proceedings of the International Conference on Computer Vision (ICCV)*, july 2001.

[74] Z. Zivkovic and F. V. Heijden. Efficient adaptive density estimation per image pixel for task of background subtraction. *Pattern Recognition Letters.*, 27:773–780, 2006.

[75] Z. Tang and Z. Miao. Fast background subtraction and shadow elimination using improved gaussian mixture model. In *Proceedings of the International Workshop on Haptic, Audio and Visual Environments and Games (HAVE)*, oct. 2007.

[76] H. Ardoe and R. Berthilsson. Adaptive background estimation using intensity independent features. In *Proceedings of the British Machine Vision Conference (BMVC)*, sep. 2006.

[77] H. Bhaskar, L. Mihaylova, and S. Maskell. Background modeling using adaptive cluster density estimation for automatic human detection. *In Lecture Notes in Informatics, GI Jahrestagung (2)*, pages 130–134, 2007.

[78] J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In *Proceedings of the International Conference on Computer Vision (ICCV)*, oct. 2003.

[79] C. Ridder, O. Munkelt, and H. Kirchner. Adaptive background estimation and foreground detection using kalman-filtering. In *Proceedings of the International Conference on recent Advances in Mechatronics (ICRAM)*, 1995.

[80] R. Zabih and J. Woodfill. Non-parametric local trans-forms for computing visual correspondence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, may 1994.

[81] B. Dinkar and S. K. Nayar. Ordinal measures for visual correspondence. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, june 1996.

[82] J. Jacques, C. Jung, and S. Musse. Background subtraction and shadow detection in grayscale video sequences. In *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, oct. 2005.

[83] W.E.L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, june 1998.

[84] T. Gandhi and M. M. Trivedi. Vehicle mounted wide fov stereo for traffic and pedestrian detection. In *Proceedings of the International Conference on Image Processing (ICIP)*, sep. 2005.

[85] L. Spinello, R. Triebel, and R. Siegwart. Multimodal detection and tracking of pedestrians in urban environments with explicit ground plane extraction. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, sep. 2008.

[86] R. Moe-Nilson. A new method for evaluating motor control in gait under real-life environment conditions. *International Journal of Clinical Biomechanics*, 13:328–335, 1998.

[87] C. Barron and I.A. Kakadiaris. Estimating anthropometry and pose from a single image. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, june 2000.

[88] De A. Danilo, S. Remo, A. Cantatore, P. Poppa, M. Dufour, M. Grandi, and C. Cattaneo. New method for height estimation of subjects represented in photograms taken from video surveillance systems. *International Journal of Legal Medicine*, 121(6):489–492, 2007.

[89] R. Korpelainen, H. Kaikkonen, V. Kampman, and J. T. Korpelainen. Reliability of an inclinometric method for assessment of body sway. *Technology Health Care*, 2005.

[90] F. Cupillard, F. Bremond, and M. Thonnat. Tracking groups of people for video surveillance. Press of University of Kingston (London), 2001.

[91] R. T. Collins, R. Gross, and J. Shi. Silhouette-based human identification from body shape and gait. In *Proceedings of the Conference on Face and Gesture Recognition (FGR)*, may 2002.

[92] C. BenAbdelkader, R. Cutler, and L. Davis. Person identification using automatic height and stride estimation. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, aug. 2002.

[93] T. Izumi and Y. Uerki. Improvement of precision by introducing vitality degree to the height measurement system using stereo vision. In *Proceedings of the SICE 2004, Annual Conference*, aug. 2004.

[94] A. Bovyrin and K. Rodyushkin. Human height prediction and roads estimation for advanced video surveillance systems. In *Proceedings of the Conference on Advanced Video and Signal Based Surveillance (AVSS)*, sep. 2005.

[95] A. Criminisi, A. Zisserman, and L. Van Gool. A new approach to obtain height measurement from video. In *Proceedings of SPIE, Boston, Massachussets, USA*, dec. 1998.

[96] D. Farin, P. H. N. de With, and W. Effelsberg. Robust background estimation for complex video sequences. In *Proceedings of the International Conference on Image Processing (ICIP)*, sep. 2003.

[97] M. Massey and W. Bender. Salients stills: Process and practice. *IBM Systems Journal*, 35(3.4):557–573, 1993.

[98] B. D. O. Anderson and J. B. Moor. Optimal filtering. *Information and System Science Series, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, USA*, 1979.

[99] M.-B. Dillencourt, H. Samet, and M. Tamminen. A general approach to connected-component labeling for arbitrary image representations. *Journal of the Association for Computing Machinery*, 39(2):253–280, 1992.

[100] L. Shapiro and G. Stockman. *Computer Vision*. Prentice Hall, 2002.

[101] F.W. Gembicki. *Vector Optimization for Control with Performance and Parameter Sensitive Indices*. PhD thesis, Case Western Reserve University, Cleveland, Ohio, 1974.

[102] V. Chvatal. Linear programming. W. H. Freeman and Company, 1983.

[103] R. E. Bixby. Implementing the simplex method: The initial basis. *ORSA Journal on Computing*, 4(3):267–284, 1992.

[104] E D. Andersen and K.D Andersen. Presolving in linear programming. *Mathematical Programming*, 71:221–245, 1995.

[105] M.I.A. Lourakis. levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++. [web page] http://www.ics.forth.gr/ lourakis/levmar/, Jul. 2004. [Lastly accessed on 27 Jun. 2010.].

[106] H Hirschmueller. Stereo processing by semi-global matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:328–341, 2008.

[107] P. Steingrube, S. Gehrig, and U. Franke. Performance evaluation of stereo algorithms for automotive applications. *Computer Vision Systems*, 5815:285–294, 2009. 10.1007/978-3-642-04667-4-29.

[108] R. Kawanishi, A. Yamashita, and T. Kaneko. Estimation of camera motion with feature flow model for 3d environment modeling by using omni-directional camera. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, oct. 2009.

[109] T. Svoboda, T. Pajdla, and V. Hlavac. Epipolar geometry for panoramic cameras. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 1998.

[110] J. Gluckman and S. Nayar. Egomotion and omnidirectional cameras. In *Proceedings of the International Conference on Computer Vision (ICCV)*, jan. 1998.

[111] S. Kang. Catadioptric self-calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, june 2000.

[112] P. Chang and M. Hebert. Omnidirectional structure from motion. In *Proceedings of the IEEE Workshop on Omnidirectional Vision (OMNIVIS)*, june 2000.

[113] C. Geyer and K. Daniilidis. Structure and motion from uncalibrated catadioptric views. In *Proceedings on the International Conference on Computer Vision and Pattern Recognition (CVPR)*, dec. 2001.

[114] A. W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proceedings on the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, dec. 2001.

[115] R. Bunschoten and B. Kröse. Range estimation from a pair of omnidirectional images. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 2001.

[116] R. Bunschoten and B. Kröse. Visual odometry from an omnidirectional vision system. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003.

[117] R. Bunschoten and B. Kröse. Robust scene reconstruction from an omnidirectional vision system. *IEEE Transactions on Robotics and Automation*, 19(2):351–357, apr. 2003.

[118] R. Bunschoten and B. Kröse. 3d scene reconstruction from cylindrical panoramic images. *IEEE Transactions on Robotics and Autonomous Systems*, 41:111–118, 2002.

[119] B. Micusik and T. Pajdla. Estimation of omnidirectional camera model from epipolar geometry. In *Proceedings of the International Conference on Computer Vision and Patter Recognition (CVPR)*, june 2003.

[120] B. Micusik and T. Pajdla. Para-catadioptric camera autocalibration from epipolar geometry. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, jan. 2004.

[121] B. Micusik and T. Pajdla. Structure from motion with wide circular field of view cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1135–1149, jul. 2006.

[122] A. Makadia, D. Gupta, and K. Daniilidis. Planar ego-motion without correspondences. In *Proceedings of the Workshop on Motion and Video Computing*, 2005.

[123] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, sep. 2008.

[124] S.-H. Jung and C.J. Taylor. Camera trajectory estimation using inertial sensor measurements and structure from motion results. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, dec. 2001.

[125] I. Bauermann. *Acquisition and streaming of image-based scene representations*. PhD thesis, Technische Universität München, Lehrstuhl fü Medientechnik, 2008.

[126] D. Scaramuzza, F. Fraundorfer, M. Pollefeys, and R. Siegwart. Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints. In *Proceedings of the International Conference on Computer Vision (ICCV)*, sep. 2009.

[127] M. Bertozzi, A. Broggi, and A. Fascioli. A stereo vision system for real-time automotive obstacle detection. In *Proceedings of the International Conference on Image Processing*, sep. 1996.

[128] K. Kohara, N. Suganuma, T. Negishi, and T. Nanri. Obstacle detection based on occupancy grid maps using stereovision system. *IEEE Transactions on Intelligent Transportation Systems*, 8:85–95, 2010.

[129] S. Krotosky and M. M. Trivedi. *Registering multimodal imagery with occluding objects using mutual information: Application to stereo tracking of humans.* Advances in Pattern Recognition. Springer London, 2009.

[130] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Bota, M.-M. Meinecke, and M.-A. Obojski. *Stereovision-based sensor for intersection assistance.* VDI-Buch. Springer Berlin Heidelberg, 2009.

[131] C. Caraffi, S. Cattani, and P. Grisleri. Off-road path and obstacle detection using decision networks and stereo vision. *IEEE Transactions on Intelligent Transportation Systems*, 8(4):607–618, dec. 2007.

[132] C. Töpfer and T. Ehlgen. A unifying omnidirectional camera model and its applications. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1–5, oct. 2007.

[133] U. Franke and S. Heinrich. Fast obstacle detection for urban traffic situations. *IEEE Transactions on Intelligent Transportation Systems*, 3(3):173–181, sep. 2002.

[134] U. Franke, S. Gehrig, H. Badino, and C. Rabe. *Towards optimal stereo analysis of image sequences.*, volume 4931 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2008.

[135] C. Keller, D. Llorca, and D. Gavrila. *Dense stereo-based ROI generation for pedestrian detection*, volume 5748 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2009.

[136] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information., june 2005.

[137] R. Klette. *Stereo-vision-support for intelligent vehicles - The need for quantified evidence.*, volume 5360 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2008.

[138] S. Gehrig, C. Rabe, and L. Krüger. 6d vision goes fisheye for intersection assistance. In *Proceedings of the Canadian Conference on Computer and Robot Vision (CCRV)*, may 2008.

[139] S. Gehrig, F. Eberli, and T. Meyer. *A real-time low-power stereo vision engine using semi-global matching.*, volume 5815 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2009.

[140] T. Ehlgen, M. Thorn, and M. Glaser. Omnidirectional cameras as backing-up aid. In *Proceedings of the International Conference on Computer Vision (ICCV)*, oct. 2007.

[141] T. Ehlgen and T. Paidla. Maneuvering aid for large vehicle using omnidirectional cameras. In *Proceedings of the Workshop on Applications of Computer Vision (WACV)*, feb. 2007.

[142] T. Gandhi and M.-M. Trivedi. Dynamic panoramic surround map: motivation and omni

video based approach. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, june 2005.

[143] T. Gandhi and M.-M. Trivedi. Vehicle surround capture: survey of techniques and a novel omni-video-based approach for dynamic panoramic surround maps. *IEEE Transactions on Intelligent Transportation Systems*, 7(3):293–308, sep. 2006.

[144] G. Passig. *Umgebungsmodellierung auf Basis von Stereo-Kamerabildern fü eine Telepräsenzanwendung*. PhD thesis, Technische Universität München, Lehrstuhl fuer Realzeit-Computersysteme, 2006.

[145] H. von Sanden. *Die Bestimmung der Kernpunkte in der Photogrammetrie*. PhD thesis, Universität Göttingen, 1908.

[146] O. Schreer. *Stereoanalyse und Bildsynthese*. Springer Verlag Berlin, Heidelberg, New York, 2005.

[147] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint.* MIT Press, Cambridge, MA, USA, 1993.

[148] R. Bunschoten. *Mapping and localization from a panoramic vision sensor.* PhD thesis, Universiteit van Amsterdam, 2003.

[149] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, jun. 1997.

[150] R. I. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, ISBN: 0521540518, second edition edition, 2004.

[151] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the DARPA Image Understanding Workshop*, 1981.

[152] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, june 1994.

[153] T. Svoboda and T. Pajdla. Epipolar geometry for central catadioptric cameras. *International Journal of Computer Vision*, 49:23–37, 2002.

[154] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. New York, NY, USA, 1987.

[155] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300, 1997.

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

[157] J. Banks and T. Corke. Quantitative evaluation of matching methods and validity measures for stereo vision. *International Journal of Robotics Research*, 20(7):512–532, 2001.

[158] R. Zabih and Woodfill J. Non-parametric local transforms for computing visual correspondence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, june 2001.

[159] R Bellman. *Dynamic programming*. Princeton University Press, 1957.

[160] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming, 1985.

[161] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. jan. 1998.

[162] G. Van Meerbergen, M. Vergauwen, M. Pollefeys, and L. Van Gool. A hierarchical symmetric stereo algorithm using dynamic programming. *International Journal of Computer Vision*, 47:275–285, 2002.

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

[164] C. Banz, H. Flatt, H. Blume, and P. Pirsch. Hardware-Architektur zur echtzeitfähigen Berechnung dichter Disparitätskarten. In *ITG Fachtagung für Elektronische Medien S̈ysteme, Technologien, AnwendungenÏ3. Dortmunder Fernsehseminar*, 2009.

[165] Bronstein I.N., Semendajew K.A., Musiol G. und Mühlig H. *Taschenbuch der Mathematik*. Wissenschaftlicher Verlag Harri Deutsch GmbH, Frankfurt am Main, 6. vollständig überarbeitete und ergänzte Ausgabe 2005.

[166] T Conroy. *Resolution invariant surfaces for panoramic vision systems*. Chapter 5 of Topics in Nonlinear Signal Processing, Doctoral Thesis, Australian National University, 2001.

[167] J. Bernd. *Digitale Bildverarbeitung*. Springer Verlag Berlin, Heidelberg, New York, 6., überarb. Aufl. 2005.

[168] S. Nayar and V. Peri. Folded catadioptric cameras. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, june 1999.

[169] S. Yi and N. Ahja. An omnidirectional stereo vision system using a single camera. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, aug. 2006.