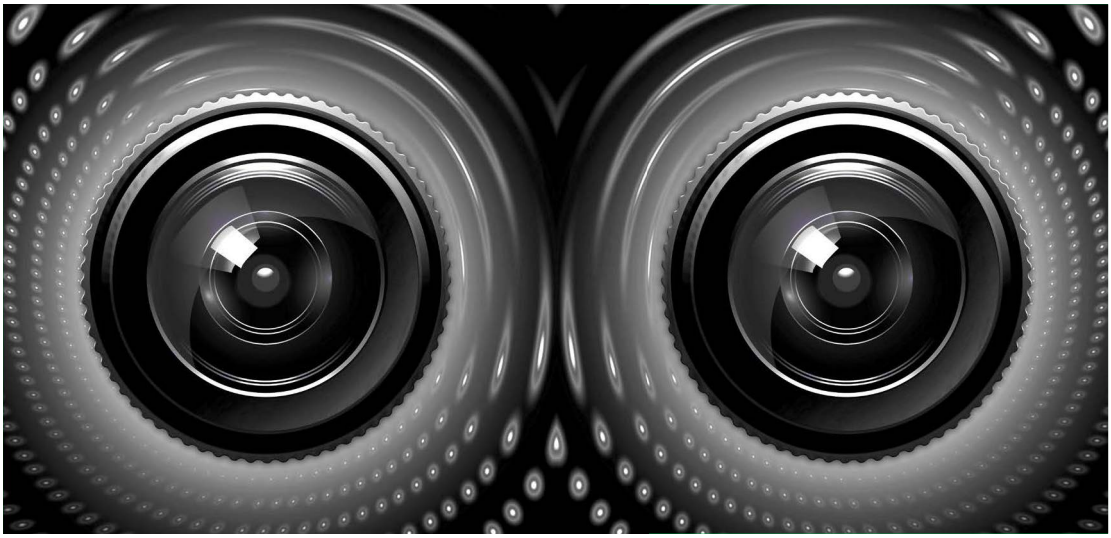
Advances in Automation Engineering Band 9

Editor: Clemens Gühmann

Hauke Brunken

Stereo Vision-Based Road Condition Monitoring



Universitätsverlag der TU Berlin

Technische Universität Berlin

Hauke Brunken
Stereo Vision-Based Road Condition Monitoring

The scientific series *Advances in Automation Engineering*
is edited by Prof. Dr.-Ing. Clemens Gühmann.

Hauke Brunken

Stereo Vision-Based Road Condition Monitoring

Universitätsverlag der TU Berlin

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the internet at http://dnb.dnb.de.

Universitätsverlag der TU Berlin, 2021
http://verlag.tu-berlin.de

Fasanenstr. 88, 10623 Berlin
Tel.: +49 (0)30 314 76131 / Fax: -76133
E-Mail: publikationen@ub.tu-berlin.de

Zugl.: Berlin, Techn. Univ., Diss., 2021
Gutachter: Prof. Dr.-Ing. Clemens Gühmann
Gutachter: Prof. Dr.-Ing. Olaf Hellwich
Gutachter: Prof. Dr.-Ing. Uwe Stilla (Technische Universität München)
Die Arbeit wurde am 18. Januar 2021 an der Fakultät IV unter Vorsitz von Prof. Dr.-Ing. Uwe Schäfer erfolgreich verteidigt.

Cover image:
geralt | https://pixabay.com/de/illustrations/objektiv-kamera-linse-digital-3541734/ | free for commercial use, no attribution required | modified

Print: docupoint GmbH
Layout/Typesetting: Hauke Brunken

ORCID iD Hauke Brunken: 0000-0002-2665-1122
https://orcid.org/0000-0002-2665-1122

ISBN 978-3-7983-3205-8 (print)
ISBN 978-3-7983-3206-5 (online)

ISSN 2509-8950 (print)
ISSN 2509-8969 (online)

Published online on the institutional repository of the Technische Universität Berlin:
DOI 10.14279/depositonce-11487
http://dx.doi.org/10.14279/depositonce-11487

Abstract

When planning road construction measures, it is essential to have up-to-date information on road conditions. If this information is not to be obtained manually, it is currently obtained using laser scanners mounted on mobile mapping vehicles, which can measure the 3D road profile. However, a large number of mobile mapping vehicles would be necessary to record an entire road network on a regular basis. Since 2D road damages can be found automatically on monocular camera images, the idea was born to use a stereo camera system to capture the 3D profile of roads. With stereo camera systems, it would be possible to equip a large number of vehicles and regularly collect data from large road networks.

In this thesis, the potential application of a stereo camera system for measuring road profiles, which is mounted behind the windshield of a vehicle, is investigated. Since this requires a calibration of the stereo camera system, but the effort for the user should be kept low, the camera self-calibration for this application is also examined.

3D reconstruction from stereoscopic images is a well-studied topic, but its application on road surfaces with little and repetitive textures requires special algorithms. For this reason, a new stereo method was developed. It is based on the plane-sweep approach in combination with semi-global matching. It was tested with different measures for pixel comparison. Furthermore, the plane-sweep approach was implemented in a neural network that solves the stereo correspondence problem in a single step. It uses the stereoscopic images as input and provides an elevation image as output.

A completely new approach was developed for the self-calibration of mono cameras and stereo camera systems. Previous methods search for feature points in several images of the same scene. The points are matched between the images and used for the calibration. In contrast to these methods, the proposed method uses feature maps instead of feature points to compare multiple views of one and the same plane. To estimate the unknown parameters, the backpropagation algorithm is used together with the gradient descent method.

The measurements obtained by stereoscopic image processing were compared with those obtained by industrial laser scanners. They show that both measurements are very close to each other and that a stereoscopic camera system is in principle suitable for capturing the surface profile of a road.

Experiments show that the proposed self-calibration method is capable of estimating all parameters of a complex camera model, including lens distortion, with high precision.

Kurzfassung

Bei der Planung von Straßenbaumaßnahmen ist es unabdingbar, über aktuelle Informationen über den Straßenzustand zu verfügen. Sollen diese Informationen nicht manuell gewonnen werden, werden derzeit Messfahrzeug mit Laserscannern verwendet, welche das 3D-Straßenprofil vermessen können. Für die regelmäßige Erfassung eines gesamten Straßennetzes wäre jedoch eine große Anzahl von Messfahrzeugen erforderlich. Da 2D-Straßenschäden automatisch auf monokularen Kamerabildern gefunden werden können, entstand die Idee, ein Stereokamerasystem zur Erfassung des 3D-Profils zu verwenden. Eine große Anzahl von Fahrzeugen könnte damit ausgerüstet werden und es könnten regelmäßig Daten von großen Straßennetzen erfasst werden.

In dieser Arbeit werden die Einsatzmöglichkeiten eines Stereokamerasystems zur Messung von Straßenprofilen untersucht, dass sich hinter der Windschutzscheibe eines Fahrzeugs befindet. Da hierzu das Stereokamerasystems kalibriert sein muss, der Aufwand für den Anwender aber geringgehalten werden soll, wird außerdem die Selbstkalibrierung für diesen Einsatzzweck untersucht.

Die 3D-Rekonstruktion aus stereoskopischen Bildern ist ein viel untersuchtes Thema, aber ihre Anwendung auf Straßenoberflächen mit wenig und sich wiederholenden Texturen erfordert spezielle Algorithmen. Aus diesem Grund wurde ein neues Stereoverfahren entwickelt. Es basiert auf dem Plane-sweep-Ansatz in Kombination mit Semi-global Matching. Es wurde mit verschiedene Maßen für den Vergleich von Pixeln getestet. Darüber hinaus wurde der Plane-sweep-Ansatz in einem neuronalen Netzwerk implementiert, das das Stereo-Korrespondenzproblem in einem einzigen Schritt löst. Es verwendet die stereoskopischen Bilder als Eingabe und liefert als Ausgabe ein Höhenbild.

Für die Selbstkalibrierung von Monokameras und Stereokamerasystemen wurde ein völlig neuer Ansatz entwickelt. Bisherige Methoden suchen nach Merkmalspunkten in mehreren Bildern der gleichen Szene. Die Punkte werden zwischen den Bildern zugeordnet und für die Kalibrierung verwendet. Die vorgeschlagene Methode verwendet anstelle von Merkmalspunkten Feature-Maps um mehrere Ansichten derselben Ebene zu vergleichen. Zur Schätzung der unbekannten Parameter wird der Backpropagation-Algorithmus zusammen mit dem Gradientenabstiegsverfahren verwendet.

Die durch stereoskopische Bildverarbeitung erhaltenen Messungen wurden mit Messungen von industriellen Laserscannern verglichen. Sie zeigen, dass beide sehr nahe beieinander liegen und dass ein Stereokamerasystem für die Erfassung des Oberflächenprofils einer Straße grundsätzlich geeignet ist.

Experimente zeigen, dass die neue Selbstkalibrierungsmethode in der Lage ist, alle Parameter eines komplexen Kameramodells, einschließlich der Linsenverzerrung, mit hoher Präzision abzuschätzen.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

API Application programming interface 89, 90

BilSub Background subtraction by bilateral filtering 17, 45, 97–99, 102–110, 115

CNN Convolutional neural network 3, 17, 39, 58, 97–99, 102–110, 114–116, 141, 144

CPU Central processing unit 90, 144

Census Census transform 17, 45, 97–99, 102–110, 115

DLSC Deep learning self-calibration 62, 63, 68, 69, 117, 118, 121–131

DLT Direct linear transform 30, 33

GPU Graphics processing unit 26, 121, 144, 145

GUM Guide to the expression of uncertainty in measurement 93, 94

HMI Hierarchical mutual information 17, 21, 45, 97–99, 101–110, 114, 116

ICP Iterative closest point 93, 95

LIDAR Laser light detection and ranging 95

MAP Maximum a posteriori 22, 25

MI Mutual information 18, 19, 25, 45, 48

MPT Movable planar target 34, 90, 117, 120–132

MRE Mean relative error 121, 125

MRF Markov random field 22, 23, 25

MdRE Median relative error 121, 125

NCC Normalized cross-correlation 50, 51

PCA Principal component analysis 47–49

RANSAC Random sample consensus 47, 49, 66, 67

RMS Root mean square 94, 96, 98–100, 115, 116, 120, 143

SAD Sum of absolute differences 18

SGM Semi-global matching 27, 46, 48, 50, 97, 144

SIFT Scale invariant feature transform 49, 66

SIMD Single input multiple data 27, 144

SVD Singular value decomposition 30, 33

SfM Structure from motion 49, 59

Notations

Notation	Description
$\mathbf{x}$	Vector $(x_1, x_2, x_3)^T$ of homogeneous 2D point coordinates
$\underline{x}$	Vector of inhomogeneous 2D point coordinates
$\mathbf{X}$	Vector $(X_1, X_2, X_3, X_4)^T$ of homogeneous 3D point coordinates
$\underline{X}$	Vector of inhomogeneous 3D point coordinates
$\underline{\mathbf{X}}$	Matrix or array

Symbols

Symbol	Description	Unit
A_L	Lens aperture size	m^2
A_R	Road segment size	m^2
A_s	Image sensor size	m^2
B_e	Irradiation	$\frac{\mathrm{W\,s}}{\mathrm{m}^2}$
$\underline{C}$	Coordinates of the camera center	m
C	Clique in a graphical model	-
D_L	Aperture diameter	m

Symbol	Description	Unit
$E(x,y), E(x)$	Energy function	-
$\underline{\mathbf{F}}$	Fundamental matrix	-
F_L	Focal point of a lens	m
$\underline{\mathbf{H}}$	Homography	-
H	Image height	px
H_L	Principal point of a lens	m
H_X / $H_{X,Y}$	Entropy of a random variable X / joint entropy of random variables X, Y	-
$\underline{\mathbf{I}}$	Unit matrix	-
I	Image – the intensity value of the pixel with coordinates (u,v) is found by $I[u,v]$	-
$\underline{\mathbf{K}}$	Camera matrix that outputs homogeneous pixel coordinates	-
$\underline{\mathbf{K}}_m$	Camera matrix that outputs homogeneous metric coordinates	-
$K_{1...6}$	Parameters of the distortion model	-
K_s	Smoothing term constant	-
L_R	Radiance of the road	$\frac{\mathrm{W}}{\mathrm{m}^2\,\mathrm{sr}}$
$\underline{M}$	Object point	m
$\underline{\mathbf{M}}$	Matrix of object points	m
$\mathcal{M}_x$	Set of inhomogeneous image point coordinates of the image I_x	px
$\underline{\bar{\mathbf{M}}}$	Mean subtracted matrix of object points	m
$\underline{\bar{M}}$	Central point of a point cloud	m
$\mathcal{N}_{NB}$	Set of indices of all neighboring pixels.	-
O	Label image	-
$\underline{\mathbf{O}}$	Output volume of the 3D network	-
P	Number of plane hypotheses	-
$\underline{\mathbf{P}}$	Projection matrix – Mapping from metric 3D point coordinates to homogeneous 2D pixel coordinates	-
$\underline{\mathbf{P}}_m$	Projection matrix – Mapping from metric 3D point coordinates to metric homogeneous 2D point coordinates	-
P_1, P_2	Parameters of the distortion model	-
P_P	Probability of correctly matching an isolated pixel	-
$P_X(x)$	Discrete probability distribution of the random variable X	-

Symbol	Description	Unit
Q	Normalizing constant to make a probability distribution integrate to one.	-
$\underline{\mathbf{R}}$	Rotation matrix that describes the orientation of a camera	-
$\underline{R}_1, \underline{R}_2, \underline{R}_3, \underline{R}_4$	Object coordinates of the road segment's corners	m
$\underline{T}$	Rotated coordinates of the camera center	m
W	Image width	px
$\underline{\mathbf{W}}$	Rotation matrix of the principal component analysis	-
$\mathbf{X}$	Object point	$\equiv$ m
$\mathbf{X}_{\text{cam}}$	Object point in the camera coordinate system	$\equiv$ m
$\underline{\mathbf{Z}}$	Feature map	-
Z	Number of features	-
Z_{LS}	Z-component of a point from the laser scan point cloud as a random variable	m
Z_{SV}	Z-component of a point from the stereo vision point cloud as a random variable	m
b	Baseline – Distance between stereo camera centers	m
b_x, b_y	Dimensions of a road patch, which is captured by a pixel	px
$\underline{c}$	Coordinates of the center of distortion	px
$\mathbf{c}_{1,2}^{i}$	Coordinates of the circular points in the i'th view	-
d	Disparity – Distance between two corresponding pixels in two rectified images	px
$\underline{d}$	Distortion vector	px
d_L	Distance between the lens and a road surface element	m
d_{PC},$d_{PC,Z}$	Distance between nearest neighbors of two point clouds and its Z-component	m
$\hat{\underline{d}}$	Estimated distortion vector	px
$\underline{e}$	Distortion error vector	px
$\bar{\underline{e}}$	Relative distortion error vector	-
f	Focal length of the pinhole camera model	px
f_{L}	Focal length of a lens	m
f_k	F-number	F/

Symbol	Description	Unit
$g_\psi\left(\underline{x}-\underline{\mu}\right)$	Gaussian probability distribution with covariance matrix ψ and expectation $\underline{\mu}$.	-
h	Height above ground of the stereo camera	m
$h_{I_x,I_y}\left(\underline{m}\right)$	Portion of the joint entropy of images I_x, I_y that is added by a single pixel $\underline{m}$	-
h_i	Image height in the lens model	m
h_o	Object height in the lens model	m
i_x	Single intensity value of the image I_x	-
$\underline{m}$	Image point	px
m_x, m_y	Pixel dimensions	m
$\underline{\hat{m}}$	Estimated image point	px
$\underline{\tilde{m}}$	Distorted image point	px
$\underline{\bar{m}}$	Normalized image point	px
$\underline{\breve{m}}$	Normalized, distorted image point	px
$\underline{\grave{m}}$	Centered image point	px
n_Q	Number of quantization levels in a search path	-
n_sp	Number of pixels in a search path	-
o	A label	-
$\underline{p}$	Principal point coordinates	m
$\underline{p}_\mathrm{Im}$	Image point	m
$p_X(x), p(x)$	Continuous probability distribution of the random variable X	-
$\underline{p}_\mathrm{i}$	Image point coordinates in the 2D example	m
$\underline{p}_\mathrm{o}$	Object point coordinates in the 2D example	m
$\underline{r}_1, \underline{r}_2, \underline{r}_3, \underline{r}_4$	Image coordinates of the road segment's corners	px
r_B	Range of the terrestrial laser scanner beam	m
s	Skew paramter of the camera matrix	-
s_L	Lens thickness	m
s_i	Image distance in the lens model	m
s_o	Object distance in the lens model / focusing distance	m
s_u	Image sensor width	px
s'_v	Usable image sensor height	px
s_y	Image sensor height	m
s'_y	Usable image sensor height	m
t_E	Exposure time	s
t_{PHL}, t_{PLH}	Propagation delay between a trigger signal and the reaction of the interal line status	s

Symbol	Description	Unit
t_f	Transition time of a square wave signal from high to low	s
u_0, v_0	Principal point coordinates	px
$u_{D_{PC,Z}}$	Uncertainty or standard deviation of the Z-component of the difference between laser scan and stereo vision measurement	m
$u_{Z_{LS}}$	Uncertainty or standard deviation of the Z-component of the laserscan measurement	m
$u_{Z_{SV}}$	Uncertainty or standard deviation of the Z-component of the stereo vision measurement	m
u_a	Uncertainty or standard deviation of the beam position caused by the angular uncertainty of the terrestrial laser scanner	m
u_{r_B}	Uncertainty or standard deviation of the range measurement of the terrestrial laser scanner	m
u_α	Uncertainty or standard deviation of the beam angle of the terrestrial laser scanner	rad
v	Vehicle speed	$\frac{\mathrm{m}}{\mathrm{s}}$
$\mathbf{x}$	Image point	$\equiv$ px
$\mathbf{x}_m$	Image point	$\equiv$ m
$\hat{\mathbf{x}}$	Estimated image point	$\equiv$ px
$\tilde{\mathbf{x}}$	Distorted image point	$\equiv$ px
$\bar{\mathbf{x}}$	Normalized image point	$\equiv$ px
$\check{\mathbf{x}}$	Normalized, distorted image point	$\equiv$ px
r_d	Distance above ground from the center of the stereo rig to the center of the front edge of the road segment	m
r_l	Road segment length	m
r_r	Distance from either heads of the stereo camera to the center of the front edge of the road segment	m
r_w	Road segment width	m
$\Theta()$	Operator that applies the distortion function	-
Φ_e	Radiant flux	W
Ψ	Factor in a Markov random field	-
$\underline{\mathbf{\Omega}}_\infty$	Absolute conic	-
$\underline{\mathbf{\Omega}}^*_\infty$	Dual absolute quadric	-
α	Beam angle of the terrestrial laser scanner	rad
β_L	Angle of a light ray entering the lens measured against the optical axis	rad

Symbol	Description	Unit
β_R	Angle of a light ray measured against the normal of the road segment	rad
β_y	Camera opening angle in vertical direction	rad
$\delta()$	Operator that transforms homogeneous to inhomogeneous coordinates	-
θ	Rotation around the Y-axis	rad
$\kappa()$	Operator that applies the camera matrix	-
λ	Wavelength	m
$\underline{\pi}_\infty$	Plane at infinity	-
$\sigma_{\mathrm{SM}}()$	Softmax operator	-
σ_d	Parameter of the Gaussian domain kernel function	-
σ_r	Parameter of the Gaussian range kernel function	-
φ	Rotation around the X-axis	rad
ψ	Rotation around the Z-axis	rad
$\underline{\omega}$	Image of the absolute conic	-
$\underline{\omega}^*$	Dual image of the absolute conic	-
$\phi()$	Operator that maps pixel coordinates from one image to the other	-

1 Introduction

In the field of road construction, knowledge about the condition of roads is of great importance. During road maintenance, surface defects that impair driving safety must be quickly detected and eliminated. This concerns surface defects[1] that can occur without prior notice, such as potholes and blow-ups [87]. Preventive maintenance is carried out to extend the life of pavements and reduces the need for more extensive repairs. The decision as to when and how it is carried out depends not only on the individual surface defect but on the overall condition of the road, including the surface shape [76]. For planning road renewals at a network level, individual surface defects are not of interest. Instead, road construction authorities use indices that describe the overall condition of road segments [87, 93]. When determining the indices, the smoothness of the surface is taken into account in addition to the defects. These applications require up-to-date information, which means that the roads need to be measured and recorded regularly, and for example, in Germany, federal roads are visited every four years [81]. In the process, it is not enough to record defects; the surface shape must also be recorded.

1.1 Motivation

A common approach to gathering information on road conditions is the use of mobile mapping vehicles equipped with laser scanners, laser triangulation devices, and cameras [20, 24, 31, 81]. In this way, the road profile can be recorded, but operating these vehicles is expensive. Prices are in the region of 100 € per km [14, 103]. Furthermore, to maintain up-to-date information about entire road networks, a large number of mobile mapping vehicles is required. An alternative method for road condition monitoring is, therefore, being sought, which ideally would not require special vehicles.

Surface defects, for whose detection knowledge about surface deformations is not necessary, can be detected by analyzing monocular camera images. These are, among others, cracks, potholes, patches, and open joints [31]. For surface defects, like depressions and rutting, however, knowledge about the surface

[1] A catalog of damages occurring on asphalt concrete and paved roads used by German authorities can be found in [34].

deformation is necessary and they cannot be detected on monocular images [24, 85]. That raises the question of whether stereo camera systems could be suitable for measuring the shape of the surface. Especially if the cameras were installed behind the windshield of a vehicle, a large number could be equipped with it. They could be mounted on garbage trucks, for example, so that a large part of the road network would be regularly surveyed without additional effort.

1.2 Objectives

To answer the question

1. *Is a stereo camera system that is mounted behind the windshield of a moving vehicle suitable for measuring the 3D profile of road surfaces in the context of road condition monitoring?*

is the main objective of this thesis. To give an answer, a stereo camera system is required to record test data, and an algorithm is needed that can extract depth information. Stereoscopic image processing is a widely studied topic, but the reconstruction of some scenes pose a challenge. The reconstruction of road surfaces is one of them. The low texture, which is also repetitive, is problematic for matching pixels in the reconstruction process. Furthermore, many algorithms assume fronto-parallel scenes, i. e. scenes consisting of planes parallel to the image plane of a reference camera [89]. They rely on rectified stereoscopic images, which are processed into disparity maps [88, 96]. The disparity, in turn, corresponds to a distance from the cameras, such that the scene's depth is sampled in the viewing direction of the cameras. In the case of cameras behind a windshield, which are tilted in relation to the road, this is unproductive, since the representation of a perfectly flat road requires many disparity levels. The dimension of interest, namely the elevation of the road, can be resolved with much fewer levels. Therefore, two intermediate objectives must be achieved:

1. a) *A stereo camera system must be designed and put into operation.*
1. b) *A specialized algorithm for depth estimation from stereoscopic images of low-textured slanted planes has to be developed.*

To extract depth information from a stereo camera system, it must be calibrated. A thorough calibration requires calibration targets that need to be produced. This prevents the easy application of a stereo camera system. Self-calibration methods that do not depend on special calibration targets exist but are limited in the mathematical camera model that can be used. Therefore, another question arises:

2. *Is it possible to automatically calibrate the stereo camera system installed in a vehicle behind the windshield with a sufficiently high accuracy for road condition monitoring?*

which shall be answered in this thesis.

1.3 Contributions

The research discussed in this thesis makes two important contributions:

1. The suitability of stereo imaging for road condition monitoring is investigated. In the course of this investigation, several side contributions are produced:
   a) It is examined how a stereo camera system can be designed that is ideal in terms of resolution and expected reconstruction performance.
   b) Two algorithms for the dense reconstruction of low-textured slanted planes are introduced.
   c) Experiments to assess the suitability of stereo imaging in this context are carried out.

   The dense reconstruction algorithms both use the plane-sweep approach, as this naturally solves the problems arising in this context. The first one uses traditional methods for the comparison of pixels and for solving the optimization problem. The second one uses a convolutional neural network (CNN) to solve the stereo correspondence problem.

2. A novel method for the automatic calibration of mono and stereo cameras is introduced. It uses an entirely new approach compared to previous ones. Instead of relying on feature points, entire calibration images are compared using feature maps. It uses the backpropagation algorithm with gradient descent to infer the unknown camera parameters.

1.4 Overview

Following the introduction, the basics of stereo cameras and camera calibration are given in Chapter 2. It starts with the image formation of a single camera, introduces the projective camera model, and continues with how this model is used in the depth estimation from stereoscopic images. Afterward, an overview of camera calibration techniques is given.

In Chapter 3, after the plane-sweep method is explained, the developed algorithms for depth estimation in the case of low-textured slanted planes are introduced.

In Chapter 4 the shortcomings of the established self-calibration techniques are discussed. They are solved by a novel approach for camera self-calibration. The method is introduced for use with mono cameras and is then extended for use with stereo cameras.

In order to evaluate the stereo algorithms in the target application of road condition monitoring, a setup is required to acquire stereoscopic images of road surfaces from inside the vehicle. Therefore, in Chapter 5, considerations are made about which cameras and lenses should be used and how they should be aligned. The effects of motion blur and blur caused by defocusing are also investigated.

In Chapter 6 road surfaces are recorded with the developed stereo camera system and scanned for comparison with industrial laser scanners. With the recorded data, the stereo camera system is evaluated in combination with the presented stereo methods.

In Chapter 7 the self-calibration method is evaluated. Cameras are calibrated by a well-established method and by the proposed method and are compared to each other.

In Chapter 8 the suitability of the stereo camera system in combination with the developed algorithms for road condition monitoring is demonstrated. The stereo camera system is used to record a section of a road. From the measured data, a condition index is calculated, which is commonly used in road construction in Germany.

Finally, in Chapter 9, conclusions are drawn and suggestions for future work are made.

1.5 Publications

Parts of this thesis have been published in the following journal articles and conference papers:

1. Hauke Brunken and Clemens Gühmann. “Deep learning self-calibration from planes”. In: *Twelfth International Conference on Machine Vision (ICMV 2019)*. Vol. 11433. 2020, p. 114333L.

2. Hauke Brunken and Clemens Gühmann. “Incorporating Plane-Sweep in Convolutional Neural Network Stereo Imaging for Road Surface Reconstruction”. In: *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP*. 2019, pp. 784–791. ISBN: 978-989-758-354-4. DOI: 10.5220/0007352107840791.

3. Hauke Brunken and Clemens Gühmann. “Pavement distress detection by stereo vision/Straßenzustandserkennung durch stereoskopische Bildverarbeitung”. *tm-Technisches Messen* 86 (s1 2019), pp. 42–46.

4. Hauke Brunken and Clemens Gühmann. “Road Surface Reconstruction by Stereo Vision”. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 88.6 (Dec. 1, 2020), pp. 433–448. ISSN: 2512-2819. DOI: 10.1007/s41064-020-00130-z.

2 Basics

In this chapter, the basics about image formation of cameras, depth reconstruction from stereoscopic images, and camera calibration are explained. An understanding of these basics is required for the following chapters.

A stereo camera system consists of two or more individual cameras, which are firmly connected and mounted at a distance from each other. By comparing their images, the distance of an object to the cameras can be estimated from geometric relationships. The estimation of distances is based on a camera model that is derived in Sections 2.1 and 2.2. Since real camera lenses are not perfect elements, real images are distorted compared to perfect images in terms of the camera model. The distortion is discussed in Section 2.3. A method to remove the distortion is described in Section 2.4. Section 2.5 handles the relationship of imaged points in a stereo camera and shows how this is used to estimate the distance of a point from the camera. The parameters of a camera model are determined by camera calibration. Various methods for this are presented in Section 2.6.

2.1 Image formation

A digital camera consists of two basic elements: an image sensor and a camera lens, whereby the camera lens usually is equipped with an adjustable aperture. The camera lens itself is composed of several individual lenses with different focal lengths. Mathematically, they can be treated as a single lens with an effective focal length [46]. A single lens is modeled by the Gaussian Lens Formula [50]

$$\frac{1}{f_\mathrm{L}} = \frac{1}{s_\mathrm{i}} + \frac{1}{s_\mathrm{o}}\,. \tag{2.1}$$

The rays from an object at object distance s_o passing a lens with focal length f_L that form an image at image distance s_i are shown in Figure 2.1. The Gaussian Lens Formula can be used to calculate the image distance s_i from the object distance s_o and focal length f_L. Alternatively, the image of an object can be constructed geometrically. Rays entering the lens parallel to the optical axis are deflected at the rear principal plane and pass the focal point F_L on the exit side. Vice versa, rays passing through the focal point before entering the

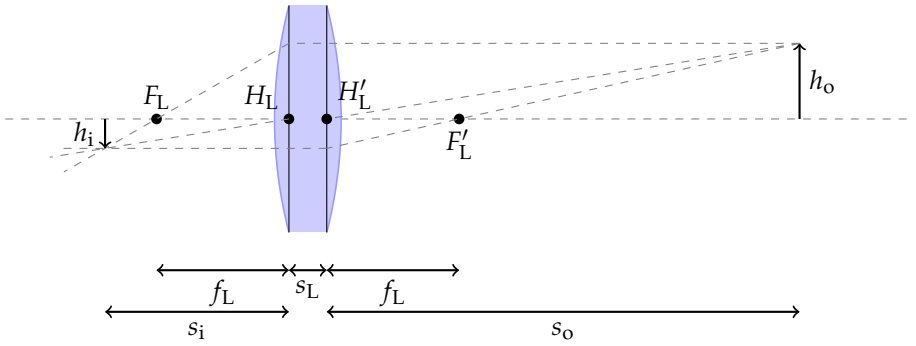


Figure 2.1: Rays entering the thick lens parallel to the optical axis are deflected at the rear principal plane and pass the focal point F_L on the exit side. Vice versa rays passing through the focal point before entering the lens are deflected at the front principal plane and leave the lens parallel to the optical axis.

lens are deflected at the front principal plane and leave the lens parallel to the optical axis. The nodal points of a lens are those through which a ray entering the lens on one side leaves the lens in parallel on the other side. In air they are located at the principal points H_L,H'_L, which are the intersections of the principal planes with the optical axis [50]. Rays through the center of the lens are not deflected.

Note that in computer vision literature, the axis through the camera center perpendicular to the image plane is called the principal axis, and their intersection is called the principal point. This is in contrast to the optics literature, where the active planes of a lens are called principal planes, and their intersection with the optical axis is called the principal point. In this section, the terms are used interchangeably as they fit. The meaning should be clear from the context.

The image of an object point that emits rays of light circularly in every direction is constructed by following the ray parallel to the optical axis, the ray passing through the object-side focal point F'_L, and the ray passing through the lens center. All three rays intersect in a single point, which is the image point. The height of the image is calculated from similar triangles

$$h_i = h_o \frac{s_i}{s_o} . \tag{2.2}$$

In order to photograph an object at s_o the image sensor must be placed at s_i. If the object moves further away, the sensor must be moved closer to the lens until it is at the focal plane at the distance f_L, and the object is infinitely far away [50].

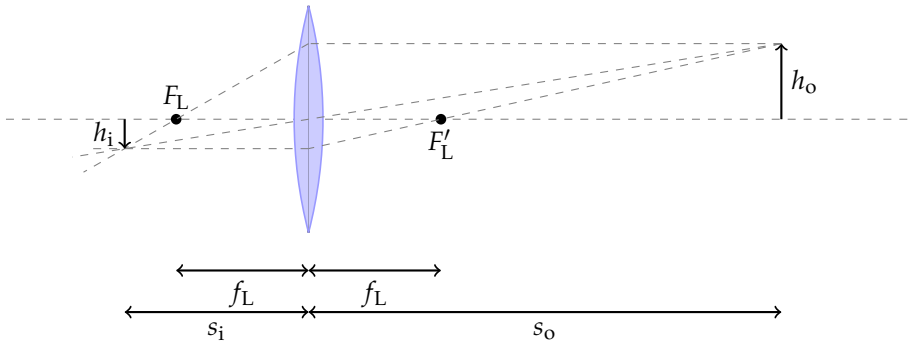


Figure 2.2: The thin lens neglects the thickness s_L of the lens.

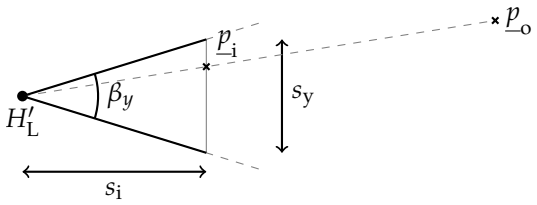


Figure 2.3: The pinhole camera model in 2D. The object point $\underline{p}_o$ is imaged at the image point $\underline{p}_i$ at the intersection of the ray from object point to the center of projection and the image sensor. The field of view β_y is determined by the image distance s_i and the size of the image sensor s_y.

To construct the image, all distances can be measured from the principal planes. Thus, the lens can be replaced by a thin lens, as shown in Figure 2.2, where the lens thickness s_L is neglected.

2.2 Pinhole camera model

A sharp image only is created if the image sensor is placed at s_i, but due to a finite aperture and a finite sensor resolution, objects at different distances can also appear sharp. By inspecting an image, it is only known that light that is caught by the image sensor must have passed the principal point H'_L, which is the lens center in case of a thin lens. This point, therefore, is called the center of projection.

With the center of projection, an image is constructed by placing the image plane or image sensor in front of the lens, as shown in Figure 2.3. The object point $\underline{p}_o = (s_o, h_o)^T$ is imaged at the image point $\underline{p}_i = (s_i, h_i)^T$ where the ray

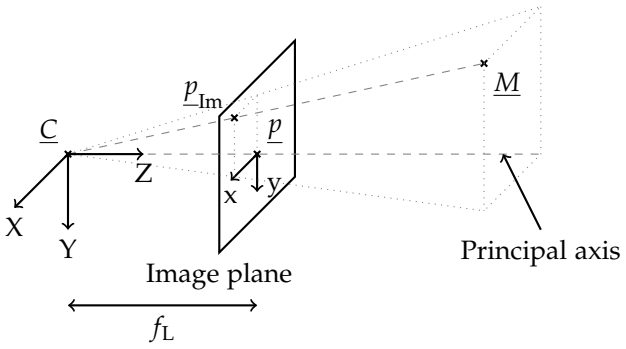


Figure 2.4: The pinhole camera model in 3D. The origin is placed at the center of projection and the image plane or sensor at $Z = f_{\mathrm{L}} \approx s_{\mathrm{i}}$.

from object point to the center of projection intersects the image sensor. This simple image formation model is referred to as the pinhole camera model [48].

In regular cameras $s_{\mathrm{i}} \approx f_{\mathrm{L}}$, which can be seen from the dimensions of f_{L} (mm) and s_{o} (m). Therefore, in the pinhole camera model, s_{i} often is replaced by f_{L}, but one has to keep in mind that s_{i} really is the distance between the back principal plane and the image sensor and that distance is changed while focusing the camera [50]. The field of view β_y of the pinhole camera model is the opening angle and it is calculated by

$$\beta_y = 2 \arctan \frac{s_{\mathrm{y}}}{2 f_{\mathrm{L}}}, \tag{2.3}$$

with the image sensor height s_{y}.

Figure 2.4 shows the pinhole camera model in 3D, where the image plane is placed at $Z = f_{\mathrm{L}}$ and the the origin is placed at the center of projection. In a camera model, the center of projection is also called the camera center $\underline{C}$ or optical center. The axis through the camera center perpendicular to the image plane is called the principal axis, and their intersection is called the principal point $\underline{p}$[1]. The image $\underline{p}_{\mathrm{Im}} = (x, y)^T$ of an object point $\underline{M} = (X, Y, Z)^T$ is the projection of the point onto the image plane through the camera center. From similar triangles the mapping from $\underline{M}$ to $\underline{p}_{\mathrm{Im}}$ is found:

$$(X, Y, Z)^T \mapsto (f_{\mathrm{L}} X/Z, f_{\mathrm{L}} Y/Z)^T. \tag{2.4}$$

[1] Note the ambiguous meaning of the two terms principal point and principal plane (Section 2.1).

In homogeneous coordinates the projection can be expressed by a linear equation [48]

$$\mathbf{x}_m = \begin{pmatrix} f_\mathrm{L} X \\ f_\mathrm{L} Y \\ Z \end{pmatrix} = \begin{pmatrix} f_\mathrm{L} & & & 0 \\ & f_\mathrm{L} & & 0 \\ & & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{2.5}$$

where $\mathbf{x}_m$ are homogeneous coordinates of the 2D image point.

The origin of image coordinates so far has been placed at the principal point. With digital cameras, the origin is in a corner, so the coordinates must be shifted. That extends Equation 2.4 to

$$(X, Y, Z)^T \rightarrow (f_\mathrm{L} X/Z + p_x, f_\mathrm{L} Y/Z + p_y), \tag{2.6}$$

which in homogeneous coordinates is

$$\mathbf{x}_m = \begin{pmatrix} f_\mathrm{L} X + p_x Z \\ f_\mathrm{L} Y + p_y Z \\ Z \end{pmatrix} = \begin{pmatrix} f_\mathrm{L} & & p_x & 0 \\ & f_\mathrm{L} & p_y & 0 \\ & & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \tag{2.7}$$

By writing

$$\underline{\mathbf{K}}_m = \begin{pmatrix} f_\mathrm{L} & & p_x \\ & f_\mathrm{L} & p_y \\ & & 1 \end{pmatrix} \tag{2.8}$$

Equation 2.7 becomes [48]

$$\mathbf{x}_m = \underline{\mathbf{K}}_m \, [\mathbf{I}|\underline{0}] \, \mathbf{X}, \tag{2.9}$$

where $\mathbf{X} = (X_1, X_2, X_3, X_4)^T$ is the object point in homogeneous 3D coordinates.

Usually, object coordinates are given in the world coordinate system rather than in the camera coordinate system. The position of the camera is given by the camera center $\underline{C}$ and its orientation by the rotation matrix $\underline{\mathbf{R}}$. Homogenous world coordinates are then transformed into homogenous camera coordinates by

$$\mathbf{X}_\mathrm{cam} = \begin{bmatrix} \underline{\mathbf{R}} & -\underline{\mathbf{R}}\underline{C} \\ \underline{0}^T & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} \underline{\mathbf{R}} & -\underline{\mathbf{R}}\underline{C} \\ \underline{0}^T & 1 \end{bmatrix} \mathbf{X}. \tag{2.10}$$

Equation 2.10 is used in 2.9 and gives

$$\mathbf{x}_m = \underline{\mathbf{K}}_m \left[\underline{\mathbf{R}} \middle| -\underline{\mathbf{R}}\underline{C}\right] \mathbf{X} = \underline{\mathbf{K}}_m \left[\underline{\mathbf{R}} \middle| \underline{T}\right] \mathbf{X}\,, \tag{2.11}$$

with

$$\underline{T} = -\underline{\mathbf{R}}\underline{C}\,. \tag{2.12}$$

$$\underline{\mathbf{P}}_m = \underline{\mathbf{K}}_m \left[\underline{\mathbf{R}} \middle| \underline{T}\right] \tag{2.13}$$

is called the projection matrix.

A digital camera measures the position of image points in pixel coordinates. If the dimension of pixels in x- and y-direction is given by m_x and m_y, pixel coordinates are given by $\underline{m} = (u, v)^T = (x/m_x, y/m_y)^T$. This can be implemented in the camera matrix

$$\underline{\mathbf{K}} = \begin{pmatrix} f_x & & u_0 \\ & f_y & v_0 \\ & & 1 \end{pmatrix} \tag{2.14}$$

with $f_x = f_\mathrm{L}/m_x$, $f_y = f_\mathrm{L}/m_y$, $u_0 = p_x/m_y$ and $v_0 = p_y/m_y$, so that image coordinates are given in pixel units

$$\mathbf{x} = \underline{\mathbf{P}}\mathbf{X} = \underline{\mathbf{K}} \left[\underline{\mathbf{R}} \middle| \underline{T}\right] \mathbf{X}\,. \tag{2.15}$$

The camera matrix $\underline{\mathbf{K}}$ describes the internal camera geometry, and its components are called intrinsic camera parameters. $\underline{\mathbf{R}}$ and $\underline{T}$ describe the location and pose of a camera and are referred to as extrinsic camera parameters.

A general projection matrix has the dimension 3×4. The left hand 3×3 submatrix of any finite camera has to be non-singular [33]. Thus, it can be uniquely decomposed into $\underline{\mathbf{P}} = \underline{\mathbf{K}} \left[\underline{\mathbf{R}} \middle| \underline{T}\right]$ by RQ matrix decomposition [48], where $\underline{\mathbf{K}}_m$ is an upper triangular matrix. That is the expected shape from Equation 2.13, but with an additional skew parameter s in the camera matrix [48]

$$\underline{\mathbf{K}} = \begin{pmatrix} f_x & s & u_0 \\ & f_y & v_0 \\ & & 1 \end{pmatrix}\,. \tag{2.16}$$

s is only non-zero for cameras with non-perpendicular pixel arrays and for some other special cases [48].

2.3 Image distortion

The projective camera model is valid only for cameras without a lens and with an infinitely small aperture, since the model assumption is that all light passes through one single point (Figure 2.4). In that case, homogeneous image coordinates are found by Equation 2.15, which are then converted into inhomogeneous coordinates $\underline{m} = (u, v)^T$ by

$$u = \frac{x_1}{x_3} \tag{2.17}$$

$$v = \frac{x_2}{x_3}\,. \tag{2.18}$$

Real cameras, however, have a lens and a finite aperture. The result is that the rays do not hit the image plane at the predicted position, but are displaced. This effect is called lens distortion and can be modeled after the perspective projection in pixel coordinates. A widely used model is Brown's model [13]

$$\begin{aligned}\tilde{u} = u &+ \breve{u}(K_1|\underline{\breve{m}}|^2 + K_2|\underline{\breve{m}}|^4 + K_3|\underline{\breve{m}}|^6) \\ &+ 2P_1\breve{u}\breve{v} + P_2(|\underline{\breve{m}}|^2 + 2\breve{u}^2)\end{aligned} \tag{2.19}$$

$$\begin{aligned}\tilde{v} = v &+ \breve{v}(K_1|\underline{\breve{m}}|^2 + K_2|\underline{\breve{m}}|^4 + K_3|\underline{\breve{m}}|^6) \\ &+ 2P_2\breve{u}\breve{v} + P_1(|\underline{\breve{m}}|^2 + 2\breve{v}^2)\,,\end{aligned} \tag{2.20}$$

with $\underline{\breve{m}} = (\breve{u}, \breve{v})^T = \underline{m} - \underline{c}$, where $\underline{c}$ is the center of distortion. $K_{1\ldots3}$ and P_1, P_2 are the distortion parameters and $\underline{\tilde{m}} = (\tilde{u}, \tilde{v})^T$ are the distorted image coordinates.

This work follows the camera model from [9] that is implemented in the OpenCV [11] and MATLAB [99] libraries. It is based on Brown's model but differs in that the distortion is calculated in normalized camera coordinates and that the center of distortion coincides with the principal point. Calculating the distortion in normalized coordinates has the advantage that the same distortion parameters can be used for different image resolutions. Combining the principal point and the center of distortion to a single point is a simplification. In OpenCV, the camera model is extended by a rational function, which is also used in this thesis. The final model is given by the following Equations

$$\bar{\mathbf{x}} = [\mathbf{R}|\underline{T}]\,\mathbf{X} \tag{2.21}$$

$$\bar{u} = \frac{\bar{x}_1}{\bar{x}_3} \tag{2.22}$$

$$\bar{v} = \frac{\bar{x}_2}{\bar{x}_3} \tag{2.23}$$

$$\begin{aligned}\breve{u} =& \bar{u}\frac{1 + K_1|\underline{\bar{m}}|^2 + K_2|\underline{\bar{m}}|^4 + K_3|\underline{\bar{m}}|^6}{1 + K_4|\underline{\bar{m}}|^2 + K_5|\underline{\bar{m}}|^4 + K_6|\underline{\bar{m}}|^6} \\ &+ 2P_1\bar{u}\bar{v} + P_2(|\underline{\bar{m}}|^2 + 2\bar{u}^2)\end{aligned} \tag{2.24}$$

$$\begin{aligned}\breve{v} =& \bar{v}\frac{1 + K_1|\underline{\bar{m}}|^2 + K_2|\underline{\bar{m}}|^4 + K_3|\underline{\bar{m}}|^6}{1 + K_4|\underline{\bar{m}}|^2 + K_5|\underline{\bar{m}}|^4 + K_6|\underline{\bar{m}}|^6} \\ &+ 2P_2\bar{u}\bar{v} + P_1(|\underline{\bar{m}}|^2 + 2\bar{v}^2)\end{aligned} \tag{2.25}$$

$$\tilde{u} = f_x\breve{u} + u_0 \tag{2.26}$$

$$\tilde{v} = f_y\breve{u} + v_0\,, \tag{2.27}$$

where $\underline{\bar{m}} = (\bar{u}, \bar{v})^T$ are normalized undistorted image coordinates and $\underline{\breve{m}} = (\breve{u}, \breve{v})^T$ are normalized distorted image coordinates.

If the parameters $K_{1\ldots6}$ and P_1, P_2 are all equal to zero, i. e. without distortion, the model collapses to Equation 2.15. In case only the distortion model without the projective camera model is needed, the normalized undistorted image coordinates are calculated by

$$\bar{u} = \frac{u - u_0}{f_x} \tag{2.28}$$

$$\bar{v} = \frac{v - v_0}{f_y} \tag{2.29}$$

instead of applying Equations 2.22 and 2.23. This way, the distorted and undistorted images have the same camera matrix. In homogeneous coordinates Equation 2.26 and 2.27 are expressed as

$$\tilde{\mathbf{x}} = \underline{\mathbf{K}}\breve{\mathbf{x}} \tag{2.30}$$

and Equation 2.28 and 2.29 as

$$\bar{\mathbf{x}} = \underline{\mathbf{K}}^{-1}\mathbf{x}\,. \tag{2.31}$$

For later use, the equations are abbreviated. Homogeneous to inhomogeneous variables (Equations 2.22 and 2.23):

$$\underline{m} = \delta(\mathbf{x}) \, ; \tag{2.32}$$

application of the distortion function (Equations 2.24 and 2.25), which can be applied in normalized and non-normalized coordinates:

$$\breve{\underline{m}} = \Theta_{K_{1\ldots6},P_1,P_2} \left(\bar{\underline{m}} \right) \, ; \tag{2.33}$$

and the application of the camera matrix (Equations 2.26 and 2.27), which can be applied in distorted and undistorted coordinates:

$$\underline{m} = \kappa \left(\bar{\underline{m}} \right) \, . \tag{2.34}$$

The estimation of the parameters of the distortion model is described in Section 2.6.

2.4 Inverse warping

If a pixel-wise mapping from one view to another is available, an image can be transformed to the other by inverse warping[2] [96]: For each pixel of a destination image, the corresponding location in the source image is calculated. From there, pixel values are copied to the destination image. Since the calculated locations are generally non-integer values, bilinear interpolation is used to estimate the value of a pixel in the source image. Since this method requires the mapping from target to source, it is called inverse warping. It is used to undistort images based on the distortion model (Equations 2.24 and 2.25), where the source is the distorted image, and the target is the undistorted image. For that reason, the distortion model does not need to be inverted. It is also used for warping images by plane homographies later in this thesis.

2.5 3D reconstruction

From Figure 2.4 it can be seen that every image point corresponds to exactly one ray. With the knowledge of the geometrical properties of a projective camera, encoded in the projection matrix $\underline{\mathbf{P}}$, the direction of these rays is known, but the position of an object point on a ray is not known. This issue is solved by using at least two cameras and by geometrically triangulating two corresponding

[2] The transformation of an image is also called warping. The terms are used interchangeably.

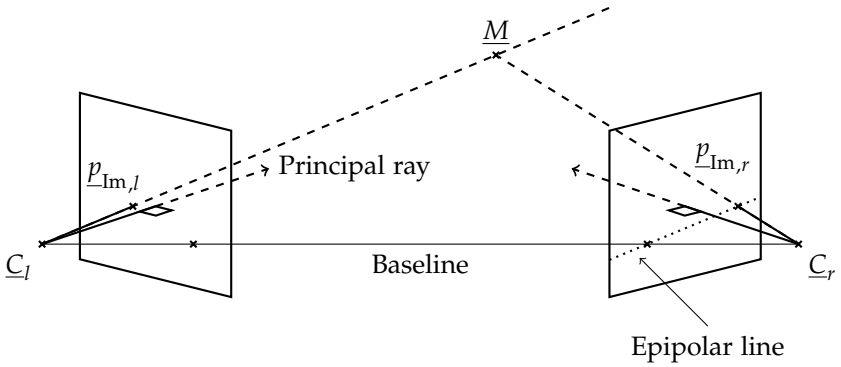


Figure 2.5: The position of an object point $\underline{M}$ is determined by finding the intersection of the rays belonging to the image points $\underline{p}_{\mathrm{Im},l}$ and $\underline{p}_{\mathrm{Im},r}$ of a camera pair.

rays, as shown in Figure 2.5, where the intersection of the rays is at the object point.

Algebraically, the problem is solved by setting up a system of linear equations. The object point is transformed to left and right camera coordinates by their projection matrices

$$\mathbf{x}_l = \underline{\mathbf{P}}_l \mathbf{X} \tag{2.35}$$

$$\mathbf{x}_r = \underline{\mathbf{P}}_r \mathbf{X}\,. \tag{2.36}$$

These equations can be solved for the object point $\mathbf{X}$. Since small errors in the location of the image points lead to non-intersecting rays, in practice it cannot be solved directly, but a best fitting solution has to be found [48].

The difficulty with a 3D reconstruction from stereoscopic images is to find matching pixels in both images, i. e. to find tie points. Fortunately, it is not necessary to search the entire image for a match. The image point $\underline{p}_{\mathrm{Im},l}$ in the left image in Figure 2.5 is back-projected at a ray shown by a dashed line. The image of that ray is a line in the right image – the epipolar line – which is shown as a dotted line. Thus, the image of the object must lie on the epipolar line in the right image, and only that line has to be searched for the corresponding point.

2.5.1 Image rectification

The problem becomes particularly easy if the cameras are aligned horizontally and in parallel, and both cameras have identical focal lengths. The epipolar

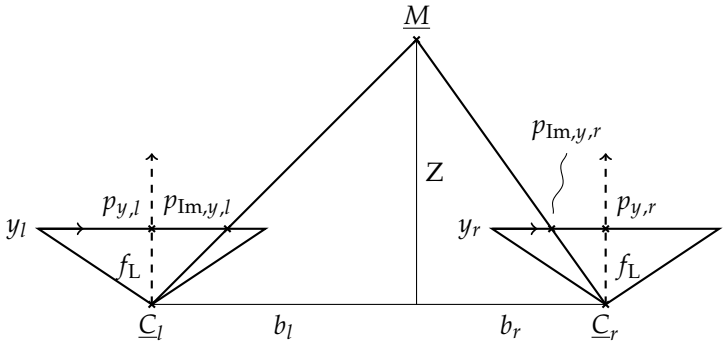


Figure 2.6: The triangulation of object points becomes particularly easy if the cameras are aligned horizontally and in parallel, and both cameras have identical focal lengths. The corresponding pixel to a pixel in one image is then found at the same height on a horizontal line in the other image.

lines then become parallel, and the corresponding pixel to a pixel in one image is found at the same height on a horizontal line in the other image.

Figure 2.6 shows the resulting geometry. By similar triangles the following relations are found

$$b_l = \frac{Z}{f_\mathrm{L}}(p_{\mathrm{Im},y,l} - p_{y,l}) = \frac{Z}{f_y}(v_l - v_{0,l}) \tag{2.37}$$

$$b_R = \frac{Z}{f_\mathrm{L}}(p_{y,r} - p_{\mathrm{Im},y,r}) = \frac{Z}{f_y}(v_{0,r} - v_r) \tag{2.38}$$

$$\Rightarrow b_l + b_r = b = \frac{Z}{f_y}\left((v_l - v_{0,l}) + (v_{0,r} - v_r)\right) . \tag{2.39}$$

With $v_{0,l} = v_{0,r}$ and the disparity defined as $d = v_l - v_r$ a connection between the disparity and the distance is found

$$Z = \frac{bf_y}{d} . \tag{2.40}$$

By determining the depth from the disparity, the problem of non-intersecting rays is avoided, and the search for matching pixels is simplified. Therefore, many stereo reconstruction algorithms work with such stereo camera setups. To use those algorithms for other camera setups, images are artificially rectified as shown in Figure 2.7, where the original images are projected onto image planes that are aligned horizontally and in parallel. In this case, the new virtual cameras have different principal points, and Equation 2.39 must be used. The

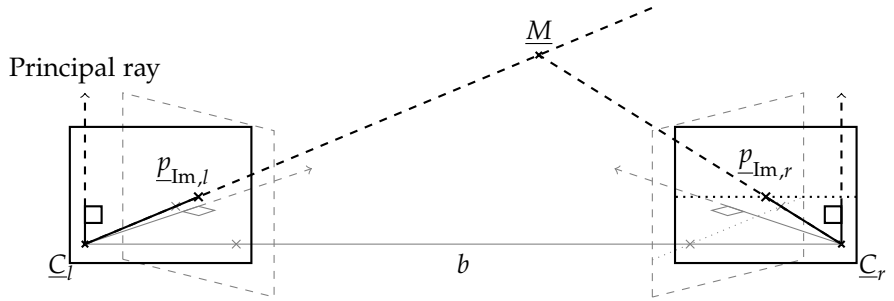


Figure 2.7: If images are taken with cameras in general orientation, the images can be rectified. After that, they look as if the cameras were aligned horizontally and in parallel and had the same focal lengths.

transformations from real to artificial cameras are homographies that can be calculated from the in- and extrinsic camera parameters [48].

2.5.2 Image features

One way of finding tie points in an image pair is to search for feature points in both individual images and to compare them. The tie points can then be triangulated. This approach is useful if the point correspondences are also used to gain more information about the scene, such as to infer the camera parameters and the 3D locations at the same time. The drawback is that the result is only a sparse reconstruction. If the camera parameters are already known, images can be rectified so that the corresponding pixel for a pixel in the first image must lie on the same horizontal line in the other image. The pixel from the first image can then be compared to each pixel on that line from the other image with the help of a similarity measure. The similarity measure is used to calculate a score for each comparison. Neglecting noise and ambiguities, the comparison that gives the highest score indicates the correct match.

The most straightforward measure compares pixels of greyscale images by their intensities. The shortcoming of this approach is that it does not consider radiometric differences between images. These appear due to slight differences between the individual cameras of the stereo camera system, but more importantly, due to non-Lambertian surfaces [55]. In the target application of road surface reconstruction, a stereo camera with a large baseline is used. This leads to different angles between the light source, the road, and the two cameras and, thus, to radiometric differences.

In [55] different similarity measures in combination with different optimization algorithms (Section 2.5.3) are compared, and three particularly interesting similarity measures are identified. According to the results in [55], background subtraction by bilateral filtering (BilSub) performs well for low radiometric differences. Hierarchical mutual information (HMI) is particularly suitable if strong image noise is present. The Census transform (Census) has a good overall performance, except for the case of strong image noise.

A more recent and widely used choice for estimating the similarity of rectangular pixel patches is a CNN [109]. At the time of working on this part, all top-ranking algorithms on the Middlebury stereo evaluation benchmark [1] used CNNs. Nevertheless, they are not used for this purpose in this work. The problem of matching pixels of low-textured road surfaces seems to be a different task than in public data sets, which mostly consist of scenes of arranged everyday objects, and the advantage of a CNN in comparing low-textured patches seems limited compared to traditional methods. Instead, in Section 3.3, a CNN for one-step depth reconstruction is introduced, which also solves the optimization problem that is introduced in Section 2.5.3. BilSub, HMI, and Census are described in the following.

Background subtraction by bilateral filtering (BilSub)

Before comparing the pixel values between two images, radiometric differences are removed with the help of a bilateral filter. The bilateral filter is a lowpass filter and weights pixels depending on their neighborhood $\mathcal{N}_{\underline{m}}$ around the pixel $\underline{m}$ [96]

$$I^{\mathrm{LP}}\left[\underline{m}\right] = \frac{\sum_{\underline{n}\epsilon\mathcal{N}_{\underline{m}}} I\left[\underline{n}\right] w_I(\underline{m},\underline{n})}{\sum_{\underline{n}\epsilon\mathcal{N}_{\underline{m}}} w_I(\underline{m},\underline{n})}. \tag{2.41}$$

I is an intensity image and the intensity of a single pixel with coordinates $\underline{m}$ is given by $I\left[\underline{m}\right]$. The weighting factor $w_I(\underline{m},\underline{n}) = d(\underline{m},\underline{n}) \cdot r_I\left(\underline{m},\underline{n}\right)$ is the product of a domain kernel $d(\underline{m},\underline{n})$, which depends on the pixel distance, and a range kernel $r_I\left(\underline{m},\underline{n}\right)$, which depends on the pixel data. In the case of Gaussian kernels these are [96]

$$d(\underline{m},\underline{n}) = \exp\left(\frac{-\left|\underline{m}-\underline{n}\right|^2}{2\sigma_d^2}\right) \tag{2.42}$$

$$r_I\left(\underline{m},\underline{n}\right) = \exp\left(-\frac{\left(I\left[\underline{m}\right]-I\left[\underline{n}\right]\right)^2}{2\sigma_r^2}\right), \tag{2.43}$$

with parameters σ_r and σ_d. The filter smoothes an image by removing high frequencies without blurring edges [5], and essentially captures the radiometric

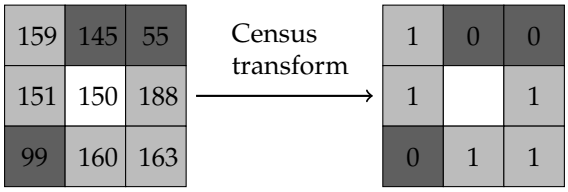


Figure 2.8: The Census transform generates a bit string from a set of pixels by comparing each pixel to the central pixel.

differences. The lowpass filtered image is subtracted from the original image

$$I^{\mathrm{BS}} = I - I^{\mathrm{LP}}, \tag{2.44}$$

which results in a highpass filter that highlights the texture.

After radiometric differences have been removed, pixel intensities are compared by calculating their differences. Alternatively, patches of both images are compared by the sum of absolute differences (SAD), which makes the measure more robust

$$SAD\left[\underline{m}\right] = \sum_{\underline{n}\epsilon\mathcal{N}_{\underline{m}}} \left| I_1^{\mathrm{BS}}\left[\underline{n}\right] - I_2^{\mathrm{BS}}\left[\underline{n}\right] \right| . \tag{2.45}$$

Census transform

The Census transform [108] transforms a set of pixels surrounding a central pixel into a bit string. A pixel is represented by a binary 0 if its value is less than that of the central pixel. Otherwise, it is represented by a binary 1. The bits are then brought into a canonical order, such as $\{1,0,0,1,1,0,1,1\}$ in Figure 2.8. In this way, a bit string is calculated for every pixel. The similarity between two pixels of Census transformed images is calculated by the Hamming distance, i. e. the differing bits between the two bit strings are counted [108].

Hierarchical mutual information

The mutual information (MI) of two random variables measures the average amount of information that one variable conveys about the other [70]. By regarding a pair of images as probability distributions of pixel intensities, their MI can be utilized as a measure for image similarity [53, 64]. The larger the MI is, the more similar two images are. A stereoscopic reconstruction establishes a mapping ϕ of pixels $\underline{m}_b$ from a base camera image to the pixels $\underline{m}_t$ of a target camera image $\phi : \underline{m}_b \rightarrow \underline{m}_t$. It can be derived from Equations 2.35 and 2.36. For rectified images, the mapping corresponds to the disparity image, i. e. the

disparity value of each pixel. With the help of ϕ the pixels of I_t are rearranged to

$$I_{tr} = \{ I_t [\phi(\underline{m}_b)] \mid \underline{m}_b \in \mathcal{M}_b \} \ , \tag{2.46}$$

where $\mathcal{M}_b$ is the set of all pixels of the base image. With the correct mapping ϕ, $I_b \approx I_{tr}$, apart from radiometric differences, different camera gain, offset, and noise. I_b and I_{tr} are regarded as random variables and their MI is calculated by

$$MI_{I_b,I_{tr}} = H_{I_b} + H_{I_{tr}} - H_{I_b,I_{tr}} \tag{2.47}$$

where $H_{I_b}, H_{I_{tr}}$ are the entropies of individual images, and

$$H_{I_x} = - \int_{I_x} p_{I_x}(i_x) \log p_{I_x}(i_x) \, di_x \ \text{ for } x = b, tr \, . \tag{2.48}$$

The joint entropy of two images is

$$H_{I_b,I_{tr}} = - \int_{I_b} \int_{I_{tr}} p_{I_b,I_{tr}}(i_b, i_{tr}) \log p_{I_b,I_{tr}}(i_b, i_{tr}) \, di_b \, di_{tr} \ . \tag{2.49}$$

p_{I_b} and $p_{I_{tr}}$ are the probability distributions of pixel intensities of I_b and I_{tr}. $p_{I_b,I_{tr}}$ is their joint probability distribution.

To make the MI usable for matching pixels, it has to be formulated as a sum over pixels, so it can be used for each individual pixel. Furthermore, the probability distribution $p_{I_b,I_{tr}}$ is unknown because the mapping ϕ is not yet available. In [64] these issues are solved by approximating Equation 2.49 by a Taylor expansion and by using Parzen estimation. The derivation from [64] is repeated below:

The employed Taylor expansion is

$$x \log(x) \simeq -x^0 + \left(1 + \log(x^0)\right) x \, , \tag{2.50}$$

which turns Equation 2.49 into

$$H_{I_b,I_{tr}} \simeq - \int_{I_b} \int_{I_{tr}} \log \left(p^0_{I_b,I_{tr}}(i_b, i_{tr}) \right) p_{I_b,I_{tr}}(i_b, i_{tr}) \, di_b \, di_{tr} \ . \tag{2.51}$$

$p^0_{I_b,I_{tr}}$ is a probability distribution that is similar to $p_{I_b,I_{tr}}$. $p_{I_b,I_{tr}}$ is estimated by Parzen estimation

$$p_{I_b,I_{tr}}(i_b, i_{tr}) \approx \frac{1}{|\mathcal{M}_b|} \sum_{\underline{m}_b} g_\psi \left((i_b, i_{tr})^T - (I_b(\underline{m}), I_{tr}(\underline{m}))^T \right) , \tag{2.52}$$

with the two-dimensional Gaussian distribution $g_\psi\left(\underline{x}-\underline{\mu}\right)$, where $\underline{\mu}$ is the expected value, and ψ is a diagonal covariance matrix. Equation 2.51 can now be written as

$$H_{I_b,I_{tr}} \approx -\sum_{\underline{m}_b} \frac{1}{|\mathcal{M}_b|} \int_{I_b}\int_{I_{tr}} \log\left(p^0_{I_b,I_{tr}}(i_b,i_{tr})\right) g_\psi\left((i_b,i_{tr})^T-(I_b(\underline{m}),I_{tr}(\underline{m}))^T\right) di_b\, di_{tr}\ . \quad (2.53)$$

The portion that a single pixel adds to $H_{I_b,I_{tr}}$ is

$$h_{I_b,I_{tr},\phi}(\underline{m}_b) = -\frac{1}{|\mathcal{M}_b|} \int_{I_b}\int_{I_{tr}} \log\left(p^0_{I_b,I_{tr}}(i_b,i_{tr})\right) g_\psi\left((i_b,i_{tr})^T-(I_b(\underline{m}_b),I_t\left[\phi(\underline{m}_b)\right])^T\right) di_b\, di_{tr}\,, \quad (2.54)$$

where I_t was reinserted. $h_{I_b,I_{tr},\phi}(\underline{m}_b)$ now explicitly depends on the mapping of every pixel and the goal is reached.

$p^0_{I_b,I_{tr}}$ is estimated by Parzen estimation from an initial mapping ϕ^0. In the implementation, it is computed by convolving the 2D histogramm of I_b and $I^0_{tr} = \left\{I_t\left[\phi^0(\underline{m}_b)\right] \mid \underline{m}_b \in \mathcal{M}_b\right\}$ with a Gaussian function

$$p^0_{I_b,I_{tr}} = P_{I_b,I^0_{tr}}(i_b,i_{tr}) \otimes g_\psi(i_b,i_{tr})\,. \quad (2.55)$$

The convolution is indicated by $\otimes$. The histogram is given by

$$P_{I_b,I^0_{tr}}(i_b,i_{tr}) = \frac{1}{|\mathcal{M}_b|}\sum_{\underline{m}_b} T\left[(i_b,i_{tr})) = \left(I_b(\underline{m}_b),I_t\left[\phi^0(\underline{m}_b)\right]\right)\right], \quad (2.56)$$

with $T[\cdot]=1$ if its argument is true, otherwise $T[\cdot]=0$. Equation 2.54 is also efficiently computed by a convolution. The result is an array for every possible combination (i_b,i_{tr}). Its entries are given by

$$h^{\text{lookup}}_{I_b,I_{tr}}(i_b,i_{tr}) = -\frac{1}{|\mathcal{M}_b|}\log\left(p^0_{I_b,I_{tr}}(i_b,i_{tr})\right) \otimes g_\psi(i_b,i_{tr}) \quad (2.57)$$

$$= -\frac{1}{|\mathcal{M}_b|}\log\left(P_{I_b,I^0_{tr}}(i_b,i_{tr}) \otimes g_\psi(i_b,i_{tr})\right) \otimes g_\psi(i_b,i_{tr}) \quad (2.58)$$

and it is used as a lookup table in

$$h_{I_b,I_{tr},\phi}(\underline{m}_b) = h^{\text{lookup}}_{I_b,I_{tr}}\left(I_b\left[\underline{m}_b\right], I_t\left[\phi(\underline{m}_b)\right]\right)\ . \quad (2.59)$$

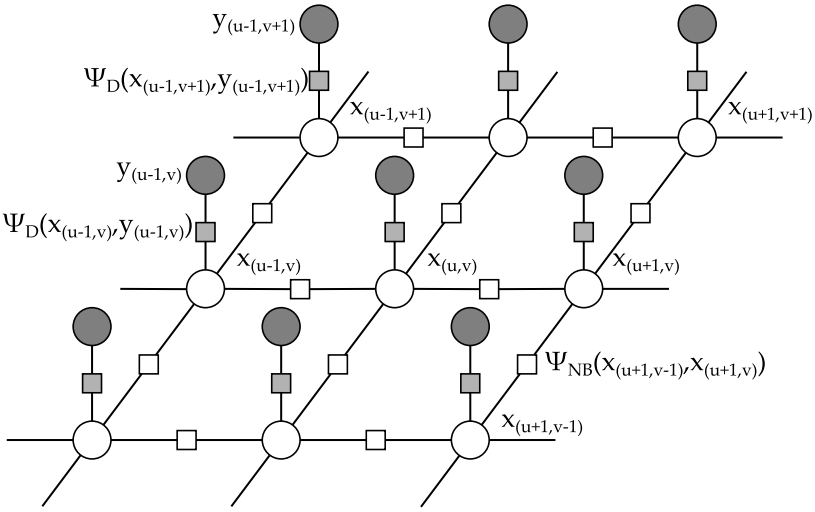


Figure 2.9: The problem of assigning an appropriate label to each pixel is expressed as a Markov random field, which is shown here as a factor graph. $x(u,v)$ is the label assigned to the pixel with coordinates (u,v) and $y(u,v)$ is the data at that pixel. $\Psi_{NB}(\cdot,\cdot)$ and $\Psi_D(\cdot,\cdot)$ are functions that describe the compatibility of adjacent labels and the compatibility of the label with the data.

H_{I_b} and $H_{I_{tr}}$ are equally converted into sums over pixels, whose summands are given by h_{I_b} and $h_{I_{tr}}$. In [64] only $H_{I_b,I_{tr}}$ is used as the pixelwise similarity measure and $H_{I_b}, H_{I_{tr}}$ are neglected. In [53] it is reported that not neglecting theses terms improves the performance of mutual information, such that the final similarity measure is

$$mi_\phi(\underline{m}_b) = h_{I_b}(\underline{m}_b) + h_{I_{tr},\phi}(\underline{m}_b) - h_{I_b,I_{tr},\phi}(\underline{m}_b)\,. \tag{2.60}$$

ϕ^0 is initialized in some way, e. g. randomly. ϕ then is refined iteratively hierarchically on an image pyramid [53], as shown in Section 3.2.1. Hence the name HMI.

2.5.3 Optimization problem

One difficulty in matching pixel pairs is that the highest similarity value does not necessarily correspond to the correct match. Low-textured surfaces might have multiple pixels with equal similarity scores, e. g. if an object is placed in front of a white background. The same problem occurs in textures with repetitive patterns. Radiometric differences due to the distance of the stereo

camera heads to each other also lead to inconsistent pixel intensities. Another reason is sensor noise. If the disparity is chosen based on similarity values alone, noisy disparity maps will be the result.

To solve this problem, it is assumed that neighboring pixels mostly belong to the same physical object and, therefore, must be related [7, 77, 96]. For the sake of clarity, this section assumes rectified stereoscopic images. Under this assumption, the disparity value is the mapping of a pixel from one image to the other and must be in accordance with the two images, i. e. with the given data. These relationships can be represented in an undirected graph.

By regarding the disparity values and the data given by the images as two collections of random variables $X = \{X_{\underline{m}} : \underline{m} \in \mathcal{M}\}$, $Y = \{Y_{\underline{m}} : \underline{m} \in \mathcal{M}\}$, the undirected graph corresponds to a Markov random field (MRF) [77]. The goal is to infer the most probable set of disparity values x from the observed data y, i. e. $\arg\max_x p(x|y)$, the maximum a posteriori (MAP) estimate. This is described in the following.

According to the Hammersley-Clifford theorem, a probability distribution that is described as a MRF can be factorized into [77]

$$p(z) = \frac{1}{Q} \prod_{c \in C} \Psi_c(z_c)\,. \tag{2.61}$$

Ψ_c is any positive function, called factor, and C is the set of cliques of the graph.

$$Q = \sum_z \prod_{c \in C} \Psi_c(z_c) \tag{2.62}$$

is a normalizing constant, which ensures that $p(z)$ is a proper density function that integrates to ones [77]. In the case at hand $Z = X \cup Y$, and the cliques each consist of the disparities of two neighboring pixels $x_{\underline{m}_i}, x_{\underline{m}_j}$, or of the disparity of a pixel and the data that belongs to the pixel $y_{\underline{m}_i}$, hence

$$p(x,y) = \frac{1}{Q} \prod_i \Psi_D(x_{\underline{m}_i}, y_{\underline{m}_i}) \prod_{i,j \in \mathcal{N}_{NB}} \Psi_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j})\,, \tag{2.63}$$

where $\mathcal{N}_{NB}$ is the set of indices of neighboring pixels. The MRF that is described by Equation 2.63 is shown as a factor graph in Figure 2.9. By choosing $\Psi_c(z_c) = \exp(-E_c(z_c))$, where E is an energy function, Equation 2.63 is turned into

$$p(x|y) = \frac{1}{Q} \exp\left(\sum_i -E_D(x_{\underline{m}_i}, y_{\underline{m}_i}) + \sum_{i,j \in \mathcal{N}_{NB}} -E_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j}) \right) \tag{2.64}$$

with the constant

$$Q = \sum_x \exp\left[\sum_i -E_D(x_{\underline{m}_i}, y_{\underline{m}_i}) + \sum_{i,j\in\mathcal{N}_{NB}} -E_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j})\right]. \tag{2.65}$$

In order to maximize $p(x|y)$ with a given configuration y, the negative log likelihood can be minimized, which is

$$-\log p(x,y) = \sum_i -E_D(x_{\underline{m}_i}, y_{\underline{m}_i}) + \sum_{i,j\in\mathcal{N}_{NB}} -E_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j}) + \log Q\,, \tag{2.66}$$

and therefore, to solve the problem, which is to find the most probable x, the energy

$$E = \sum_i E_D(x_{\underline{m}_i}, y_{\underline{m}_i}) + \sum_{i,j\in\mathcal{N}_{NB}} E_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j}) \tag{2.67}$$

must be minimized. E_D is interpreted as the energy that is related to the data, and E_{NB} is a regularization or smoothness term.

Another way of looking at the problem is by using Bayes' theorem [96]

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\sum_x p(y|x)p(x)} \tag{2.68}$$

$$\Leftrightarrow -\log p(x|y) = -\log p(y|x) - \log p(x) + \log \sum_x p(y|x)p(x)\,. \tag{2.69}$$

Comparing Equations 2.66 and 2.69, the following relations are found

$$p(y|x) = \frac{1}{Q_D} \exp\left[-\sum_i E_D(x_{\underline{m}_i}, y_{\underline{m}_i})\right] \tag{2.70}$$

$$p(x) = \frac{1}{Q_{NB}} \exp\left[-\sum_{i,j\in\mathcal{N}_{NB}} E_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j})\right]. \tag{2.71}$$

$$Q_D = \sum_x \exp\left[-\sum_i E_D(x_{\underline{m}_i}, y_{\underline{m}_i})\right] \tag{2.72}$$

$$Q_{NB} = \sum_x \exp\left[-\sum_{i,j\in\mathcal{N}_{NB}} E_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j})\right] \tag{2.73}$$

are normalizing constants to ensure that $p(y|x)$ and $p(x)$ are proper density functions. They cancel out in Equation 2.69. From Bayes' theorem interesting properties are found. The data term E_D corresponds to the conditional

probability $p(y|x)$, and E_{NB} corresponds the prior distribution of X

$$p(x) = \frac{1}{Q_{NB}} \prod_{i,j \in \mathcal{N}_{NB}} \exp\left[-E_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j})\right] \tag{2.74}$$

and is a MRF itself (compare to Equation 2.63).

2.5.4 Smoothing term

$E_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j})$ controls the smoothness of the prior distribution $p(x)$. In standard regularization techniques $\sum_{i,j \in \mathcal{N}_{NB}} E_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j})$ in Equation 2.67 is called a regularization term. A common choice is the p-norm $E_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j}) = |x_{\underline{m}_i} - x_{\underline{m}_j}|^p$ [96] and often $p = 2$. The implications on the MRF prior can be seen in the following way:

In a 4-connected neighborhood, as is shown in Figure 2.9, $E_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j}) = |x_{\underline{m}_i} - x_{\underline{m}_j}|^2$ corresponds to the discretized membrane model $\int\int (x_u^2 + x_v^2)\, du\, dv$. The resulting MRF is investigated in [95]. It turns out to be equivalent to a correlated Gaussian field with spectral distribution $S(f) \propto |2\pi f|^{-2}$. It shows that under this prior low spatial frequencies are much more probable than high frequencies, which prevents discontinuities. This prior is well suited for removing noise in otherwise flat surfaces, but depth discontinuities are obscured [8].

In [8] the relationship between regularization terms and line-processes is investigated. A line-process makes discontinuities possible by downweighting the regularization term if a discontinuity is present

$$E_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j}) = |x_{\underline{m}_i} - x_{\underline{m}_j}|^2 l_{\underline{m}_i, \underline{m}_j} + \Psi(l_{\underline{m}_i, \underline{m}_j})\,. \tag{2.75}$$

$l_{\underline{m}_i, \underline{m}_j}$ is the line-process and indicates the presence ($l_{\underline{m}_i, \underline{m}_j} \to 0$) or absence ($l_{\underline{m}_i, \underline{m}_j} \to 1$) of a discontinuity. $\Psi(l_{\underline{m}_i, \underline{m}_j})$ is the penalty for a discontinuity to be present. The disadvantage of the line-process is that it has to be included in the estimation of x, but it can be shown, that line-processes and regularization terms are related. E. g. the L1 norm $E_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j}) = |x_{\underline{m}_i} - x_{\underline{m}_j}|$ corresponds to Equation 2.75 with $\Psi = (4 l_{\underline{m}_i, \underline{m}_j})^{-1}$, where l has been estimated optimally [8]. The smoothing energy

$$E_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j}) = |x_{\underline{m}_i} - x_{\underline{m}_j}| \tag{2.76}$$

therefore is discontinuity preserving. It will be used later in this thesis.

2.5.5 Data term

For the data energy term $E_D(x_{\underline{m}_i}, y_{\underline{m}_i})$, one of the negative similarity measures from Section 2.5.2 is used. E. g. in case of MI

$$E_D(x_{\underline{m}_i}, y_{\underline{m}_i}) = -mi_\phi(\underline{m}_i)\,, \tag{2.77}$$

where the labeling is implicity given by the mapping ϕ. $y_{\underline{m}_i}$ (the data y at pixel $\underline{m}_i$) is observed and not regarded to be random. It is used in the calculation of mi.

2.5.6 Optimization algorithms

Except for some special cases of E_{NB} and binary labeling, the problem of estimating the global optimum of the MAP estimate is NP-hard [10] and therefore algorithms that find an approximate solution are required.

Graph cuts and message passing algorithms

In [94] and [62] methods for energy minimization of MRFs are compared. Both find the expansion-move algorithm [10] and the sequential tree-reweighted message passing algorithm [104] the most useful for stereo matching. The former uses graph-cuts to infer the hidden variables of the MRF. The latter is a message-passing algorithm. Message-passing algorithms were originally introduced for trees and later used for cyclic graphs.

Variational approach

In [82] a special case of Equation 2.67 is considered, where $E_{NB}(x_{\underline{m}_i}, x_{\underline{m}_j}) = |x_{\underline{m}_i} - x_{\underline{m}_j}|$. It is reformulated as a variational problem

$$E_{var} = \int_{\mathcal{M}} |\nabla x_{\underline{m}}|\, d\underline{m} + \int_{\mathcal{M}} E_D(x_{\underline{m}}, y_{\underline{m}})\, d\underline{m}\,. \tag{2.78}$$

In [82] it is shown that the global minimum of E_{var} can be found, and an algorithm is presented that can be parallelized and implemented on a graphics processing unit (GPU). The variational approach also has the advantage that it is not necessary to specify a neighborhood for the smoothing term. Therefore, it does not suffer from a grid bias.

Dynamic programming

The optimization problem is greatly simplified if scan lines (horizontal epipolar lines in rectified images) are optimized individually, and the smoothing constraint in the vertical direction is neglected. In [27] the assumption is made that matching pixels are ordered, i. e. if two pixels in the left and right images with coordinates (u_l, v_l) and (u_r, v_r) match, then the match of the pixel $(u_l + 1, v_l)$ must satisfy $(u_r + d_r, v_r) \mid d_r \geq 0$. Furthermore, instead of a smoothness term, a penalty term is added for occluded pixels. In [88] scan lines are optimized individually with a smoothness term, but without enforcing the ordering constraint. By optimizing scan lines individually, the 2D cyclic graph, which is formed by the energy function in Equation 2.67, is transformed to separate tree-structured graphs. In that case, dynamic programming can be used, which is a method for inference for any tree-structured graphical model [96].

In [88] the neighborhood of the smoothing term is taken to be the left and right neighboring pixels on the scan line, such that the lines become independent

$$E_v(x) = \sum_u \left(E_D(x_{u,v}, y_{u,v}) + E_{NB}(x_{u,v}, x_{u-1,v}) \right). \tag{2.79}$$

The sum is equivalently calculated recursively

$$E_v(x_u) = E_D(x_{u,v}, y_{u,v}) + E_{NB}(x_{u,v}, x_{u-1,v}) + E_v(x_{u-1}). \tag{2.80}$$

To minimize $E_v(x, u)$ the recursive formulation is used, which enables dynamic programming

$$\min_x E_v(x_u) = E_D(x_{u,v}, y_{u,v}) + \min_{x_{u-1}} \left(E_{NB}(x_{u,v}, x_{u-1,v}) + E_v(x_{u-1}) \right). \tag{2.81}$$

Optimization by semi-global matching

Since dynamic programming can only be used to optimize scanlines independently, the consistency between scanlines is not checked. This leads to streaking effects, which are clearly visible in the resulting disparity map. The semi-global matching (SGM) approach [53] solves this issue by accumulating cost paths from multiple directions $\underline{r}$

$$E_{\underline{r}}(x_{\underline{m}}) = E_D(x_{\underline{m}}, y_{\underline{m}}) + \min_{x_{\underline{m}-\underline{r}}} \left(E_{NB}(x_{\underline{m}}, x_{\underline{m}-\underline{r}}) + E_{\underline{r}}(x_{\underline{m}-\underline{r}}) \right) \tag{2.82}$$

$$S(x_{\underline{m}}) = \sum_{\underline{r}} E_{\underline{r}}(x_{\underline{m}}) \,. \tag{2.83}$$

The paths are traversed in a forward direction from all pixels of all borders.

$$o_{\underline{m}} = \arg\min_{x} S(x_{\underline{m}}) \tag{2.84}$$

gives the estimated label o at the position $\underline{m}$.

SGM can be implemented very efficiently. Traversing can be done independently for every path in multiple processes concurrently. By only traversing paths that start at pixels at the same edge into identical directions at the same time, i. e. paths that are parallel, $E_{\underline{r}}$ can directly be added to S during the traversal because it is guaranteed that memory that holds S is never accessed concurrently at the same position $\underline{m}$. The min operation in Equation 2.82 requires to repeat the same calculations for all possible values of x. That can be implemented by using single input multiple data (SIMD) instructions. By using the Intel AVX2 instruction set, calculations on 256-bit registers can be performed simultaneously. By using single-precision floating-point numbers, calculations for eight labels $x_{\underline{m}-\underline{r}}$ are performed simultaneously in each process. The algorithm has complexity $\mathcal{O}(WHD^2)$.

In [53] a function E_{NB} is used that is zero if neighboring pixels have the same label. It takes a first defined value if neighboring labels differ by one, and it takes a second defined value if they differ by more than one. This way, the algorithm has complexity $\mathcal{O}(WHD)$.

2.6 Camera calibration

Geometric calibration is the process of determining a camera's geometric properties and is often simply referred to as camera calibration [56]. The geometric properties can be divided into in- and extrinsic parameters. Intrinsic parameters (or internal parameters) are those that are intrinsic to the camera itself [56]. Assuming the projective camera model (Equation 2.15), these are the ones that make up the camera matrix $\underline{\mathbf{K}}$. Extrinsic parameters (or external or pose parameters) are the ones that describe the pose of the camera $\underline{\mathbf{R}}$, $\underline{T}$. Parameters of the distortion model (Equations 2.21–2.27) are often counted to the intrinsic parameters [56]. To make a distinction possible, in this thesis, they will be called distortion parameters. A calibrated camera is a camera whose intrinsic and distortion parameters are known. With a calibrated stereo camera, the geometric relationship between the stereo heads is also known.

A calibrated camera, together with the extrinsic parameters, establishes a mapping from a 3D world to a 2D image coordinate. A projective mapping cannot be inverted, but with a calibrated camera, an image coordinate can be back-projected to a ray. The object point lies on this ray, but the absolute position is unknown. Many applications in computer vision require calibrated cameras. Examples are the reconstruction of metric properties from single images, depth reconstruction from stereo cameras, and augmented reality applications. Stitching panoramic images from multiple images requires undistorted images and thus the distortion parameters.

To calibrate a camera, a calibration target is observed from one or multiple perspectives. Depending on the shape of and the knowledge about the calibration target, camera calibration can be divided into different categories. Following [56] these are:

- Camera calibration with a 3D target with control points. The 3D coordinates of which have to be known precisely.
- Camera calibration with a planar target with control points. The 2D coordinates within the plane have to be known precisely.
- Camera self-calibration with a 3D or planar target without any control points.

Combinations of these categories are also possible.

In Chapters 3–6 a calibrated stereo camera system is required for depth estimation. Therefore the basics of different calibration techniques are laid out in the remainder of this chapter. Later, a novel self-calibration method is introduced. Since it uses a feature map extraction network that is part of Section 3.3, it is covered in Chapter 4.

2.6.1 Calibration with 3D control points

The most basic calibration technique uses a 3D target with control points. The coordinates of the control points are known and can be found in an image so that the corresponding object coordinates $\mathbf{X}_i$ and image coordinates $\mathbf{x}_i$ are available. Assuming the projective camera model, a 3D object point is mapped to 2D image coordinates by

$$\hat{\mathbf{x}}_i = \underline{\mathbf{P}}\mathbf{X}_i \,. \tag{2.85}$$

Neglecting noise and model imperfections, $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ describe the same point, but since those are homogeneous coordinates, they can differ by a factor. The vectors $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ are parallel, therefore $\mathbf{x}_i \times \hat{\mathbf{x}}_i = \underline{\mathbf{0}}$ and hence [48]

$$\mathbf{x}_i \times \underline{\mathbf{P}}\mathbf{X}_i = \mathbf{X}_i \times \begin{bmatrix} \underline{\mathbf{p}}^{1T} \\ \underline{\mathbf{p}}^{2T} \\ \underline{\mathbf{p}}^{3T} \end{bmatrix} \mathbf{X}_i = \begin{bmatrix} x_{2,i}\underline{\mathbf{p}}^{3T}\mathbf{X}_i - x_{3,i}\underline{\mathbf{p}}^{2T}\mathbf{X}_i \\ -x_{1,i}\underline{\mathbf{p}}^{3T}\mathbf{X}_i + x_{3,i}\underline{\mathbf{p}}^{1T}\mathbf{X}_i \\ x_{1,i}\underline{\mathbf{p}}^{2T}\mathbf{X}_i - x_{2,i}\underline{\mathbf{p}}^{1T}\mathbf{X}_i \end{bmatrix} \tag{2.86}$$

$$= \begin{bmatrix} \underline{\mathbf{0}}^T & -x_{3,i}\mathbf{X}_i^T & x_{2,i}\mathbf{X}_i^T \\ x_{3,i}\mathbf{X}_i^T & \underline{\mathbf{0}}^T & -x_{1,i}\mathbf{X}_i^T \\ -x_{2,i}\mathbf{X}_i^T & x_{1,i}\mathbf{X}_i^T & \underline{\mathbf{0}}^T \end{bmatrix} \begin{bmatrix} \underline{\mathbf{p}}^1 \\ \underline{\mathbf{p}}^2 \\ \underline{\mathbf{p}}^3 \end{bmatrix} = \underline{\mathbf{0}} \,. \tag{2.87}$$

The first and second lines in Equation 2.87 can be transformed into the third line, which therefore can be removed and leaves [48]

$$\begin{bmatrix} \underline{\mathbf{0}}^T & -x_{3,i}\mathbf{X}_i^T & x_{2,i}\mathbf{X}_i^T \\ x_{3,i}\mathbf{X}_i^T & \underline{\mathbf{0}}^T & -x_{1,i}\mathbf{X}_i^T \end{bmatrix} \begin{bmatrix} \underline{\mathbf{p}}^1 \\ \underline{\mathbf{p}}^2 \\ \underline{\mathbf{p}}^3 \end{bmatrix} = \underline{\mathbf{0}} \,. \tag{2.88}$$

The equation is valid for all n point correspondences, hence

$$\begin{bmatrix} \underline{\mathbf{0}}^T & -x_{3,1}\mathbf{X}_1^T & x_{2,1}\mathbf{X}_1^T \\ x_{3,1}\mathbf{X}_1^T & \underline{\mathbf{0}}^T & -x_{1,1}\mathbf{X}_1^T \\ \underline{\mathbf{0}}^T & -x_{3,2}\mathbf{X}_2^T & x_{2,2}\mathbf{X}_2^T \\ x_{3,2}\mathbf{X}_2^T & \underline{\mathbf{0}}^T & -x_{1,2}\mathbf{X}_2^T \\ \vdots & \vdots & \vdots \\ \underline{\mathbf{0}}^T & -x_{3,n}\mathbf{X}_i^T & x_{2,n}\mathbf{X}_n^T \\ x_{3,n}\mathbf{X}_n^T & \underline{\mathbf{0}}^T & -x_{1,n}\mathbf{X}_n^T \end{bmatrix} \begin{bmatrix} \underline{\mathbf{p}}^1 \\ \underline{\mathbf{p}}^2 \\ \underline{\mathbf{p}}^3 \end{bmatrix} = \underline{\mathbf{A}} \begin{bmatrix} \underline{\mathbf{p}}^1 \\ \underline{\mathbf{p}}^2 \\ \underline{\mathbf{p}}^3 \end{bmatrix} = \underline{\mathbf{0}} \,. \tag{2.89}$$

The projection matrix $\underline{\mathbf{P}}$ can now be determined by solving for the vector $\underline{\mathbf{p}} = \left[\underline{\mathbf{p}}^{1T}\underline{\mathbf{p}}^{2^T}\underline{\mathbf{p}}^{3^T}\right]^T$. The projection matrix consists of 12 values, but since it is only defined up to scale, only 11 values have to be determined. To determine

$\underline{\mathbf{P}}$ uniquely, 11 equations are necessary, which corresponds to 5 1/2 point correspondences. The half correspondence can be acquired e. g. by using only the x-component of a point pair.

In practice, measured point coordinates are subject to noise, and the camera model will not perfectly reflect reality. Therefore more points are used and $\underline{\mathbf{p}}$ is determined by the right nullspace of $\underline{\mathbf{A}}$. The right nullspace can be obtained by singular value decomposition (SVD) of $\underline{\mathbf{A}}$, where the solution is the singular value that corresponds to the smallest singular value. The method is called the direct linear transform (DLT) [48].

2.6.2 Geometric error

It is also possible to measure the distance between measured and estimated inhomogeneous image coordinates, i. e. the geometric error $|\underline{m}_i - \hat{\underline{m}}_i|$ [48]. The estimated coordinates are found by projecting the 3D point coordinates $\hat{\underline{m}}_i = \delta\left(\underline{\mathbf{P}}\mathbf{X}_i\right)$. $\underline{\mathbf{P}}$ can now be determined by minimizing the squared sum of all distances

$$\min_{\underline{\mathbf{P}}} \sum_i \left|\underline{m}_i - \delta\left(\underline{\mathbf{P}}\mathbf{X}_i\right)\right|^2, \tag{2.90}$$

with a nonlinear optimization technique, such as the Levenberg-Marquardt algorithm. For initialization, the DLT method can be used. This method has the advantage that the nonlinear distortion model can be included in the optimization

$$\min_{\underline{\mathbf{P}}, K_{1\ldots6}, P_1, P_2} \sum_i \left|\underline{m}_i - \delta\left(\Theta_{K_{1\ldots6}, P_1, P_2}\left(\underline{\mathbf{P}}\mathbf{X}_i\right)\right)\right|^2. \tag{2.91}$$

2.6.3 Self-calibration with a 3D target

The requirement for self-calibration through 3D scenes is that feature points can be found in multiple images. The starting point is a projective reconstruction, i. e. estimated 3D point coordinates $\mathbf{X}$ and projection matrices $\underline{\mathbf{P}}$ from 2D image coordinates $\mathbf{x}$. The reconstruction is then upgraded to a metric reconstruction by the rectifying homography determined by the self-calibration method.

Projective reconstruction

The projective reconstruction is obtained by the following basic steps [48, 83]:

1. Search for point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ between two views.
2. Reconstruction of the fundamental matrix that relates image point correspondences $\mathbf{x}'_i \underline{\mathbf{F}}_1 \mathbf{x}_i = 0$.

3. Decomposition of $\underline{\mathbf{F}}_1$ into the projection matrices of two cameras $\underline{\mathbf{P}}_1$ and $\underline{\mathbf{P}}_0$, where $\underline{\mathbf{P}}_0 = [I|0]$.

4. Estimation of world coordinates $\mathbf{X}_i$ by triangulation (Section 2.5).

5. Search for point correspondences between one of the previous views and further views and estimation of $\mathbf{X}_i$ and $\underline{\mathbf{P}}_j$ as described in points 2.–4.

The reconstruction can be enhanced by alternating the search for correspondences and updating the fundamental matrix [83]. Once the projection matrices $\underline{\mathbf{P}}_j$ are found, and the 3D points $\mathbf{X}_i$ are triangulated, the 3D points can be back-projected into 2D camera coordinates.

The steps of estimating the projection matrices and triangulating points have been separate so far. To obtain a consistent model, the geometric error (Section 2.6.2) is minimized by concurrently altering the projection matrices and the 3D point coordinates

$$\min_{\underline{\mathbf{P}},\mathbf{X}_i} \sum_i \left| \underline{m}_i - \delta \left(\underline{\mathbf{P}} \mathbf{X}_i \right) \right|^2 , \tag{2.92}$$

which is called bundle adjustment [101]. At this point, a distortion model can also be included [83].

Since the reconstruction in points 2.–4. is performed between two views at a time, a bundle adjustment refinement can be added after adding a view (step 5) or after all views have been added, to obtain a single consistent model [83].

Metric reconstruction

In a projective reconstruction, metric properties are not preserved. A rectifying homography $\underline{\mathbf{H}}_{\text{rect}}$ is required that upgrades the projective reconstruction to a metric reconstruction. It can be shown that it must have the form

$$\underline{\mathbf{H}}_{\text{rect}} = \begin{bmatrix} \underline{\mathbf{K}} & 0 \\ -\underline{\mathbf{p}}^T \underline{\mathbf{K}} & 1 \end{bmatrix} , \tag{2.93}$$

with $\underline{\pi}_\infty = (\underline{\mathbf{p}}^T, 0)^T$ and $\underline{\pi}_\infty$ being the plane at infinity in the projective frame [48]. It transforms point vectors and projection matrices to the metric frame by $\mathbf{X}_M = \underline{\mathbf{H}}_{\text{rect}}^{-1} \mathbf{X}$ and $\underline{\mathbf{P}}_M = \underline{\mathbf{P}} \underline{\mathbf{H}}_{\text{rect}}$. To estimate $\underline{\mathbf{H}}_{\text{rect}}$, an important property of the dual absolute quadric $\underline{\mathbf{\Omega}}^*_\infty$ is used. In a metric frame $\underline{\mathbf{\Omega}}^*_{\infty,M} = diag(1,1,1,0)$, and it is transformed to the projective frame by [48, 83]

$$\underline{\mathbf{\Omega}}^*_\infty = \underline{\mathbf{H}}_{\text{rect}} \underline{\mathbf{\Omega}}^*_{\infty,M} \underline{\mathbf{H}}_{\text{rect}}^T . \tag{2.94}$$

In a general projective frame, $\underline{\mathbf{\Omega}}^*_\infty$ is described by a 4x4 symmetric positive semi-definite matrix [83]. If $\underline{\mathbf{\Omega}}^*_\infty$ is known, $\underline{\mathbf{H}}_{\text{rect}}$ can be determined by matrix decomposition. The image of the dual absolute quadric is the dual image of the absolute conic. It is found by a projection

$$\underline{\omega}^* = \underline{\mathbf{P}}_j \underline{\mathbf{\Omega}}^*_{\infty,j} \underline{\mathbf{P}}_j^T \tag{2.95}$$

equally in every j'th view, since $\underline{\omega}^* = \underline{\mathbf{K}}\underline{\mathbf{K}}^T$ is only dependent on the camera matrix. In the application where a camera shall be calibrated, all views are created by the same camera, hence $\underline{\mathbf{K}}$ and thus $\underline{\omega}^*$ are fixed. $\underline{\mathbf{\Omega}}^*_{\infty,j}$ is inherent to the projective frame. The homogenous quantities in Equation 2.95 are defined only up to scale, and, therefore, $\underline{\omega}^*$ differs by an unknown scaling between views, but for its elements, $\frac{\omega^{*i}_{k,l}}{\omega^{*j}_{k,l}} = const$ for all k, l for each two views i, j. Due to the symmetric form of $\underline{\mathbf{\Omega}}^*_\infty$ and the form of $\underline{\omega}^*$, one can extract the equalities [48]

$$\frac{\omega^{*i}_{11}}{\omega^{*j}_{11}} = \frac{\omega^{*i}_{12}}{\omega^{*j}_{12}} = \frac{\omega^{*i}_{13}}{\omega^{*j}_{13}} = \frac{\omega^{*i}_{21}}{\omega^{*j}_{21}} = \frac{\omega^{*i}_{22}}{\omega^{*j}_{22}} = \frac{\omega^{*i}_{23}}{\omega^{*j}_{23}} = \frac{\omega^{*i}_{33}}{\omega^{*j}_{33}}\,. \tag{2.96}$$

By using Equation 2.95 in 2.96, a system of equations quadratic in the entries of $\underline{\mathbf{\Omega}}^*_{\infty j}$ is generated and can be solved numerically. $\underline{\mathbf{K}}$ is found from $\underline{\omega}^*$ by Cholesky decomposition.

In [83] a different approach is used. There, the projection matrices are normalized so that $\underline{\omega}^*$ can be approximated by a unit matrix. The result is a system of equations linear in the entries of $\underline{\mathbf{\Omega}}^*_{\infty j}$, which can be solved by linear least squares.

In [83] a final bundle adjustment step is performed to optimize the metric projection model. With the projection matrices $\underline{\mathbf{P}}_{M,j}$ of the metric frame, the locations of the cameras $\underline{\mathbf{R}}_j$,$\underline{T}_j$ are also known (Section 2.2). The projection model, which is established for every camera by Equations 2.21–2.27, is setup and is used to optimize the geometric error by bundle adjustment

$$\min_{\underline{\mathbf{K}},K_{1\ldots6},P_1,P_2,\underline{\mathbf{R}}_j,\underline{T}_j,\mathbf{X}_i} \sum_j \sum_i \left| \underline{m}_i - \delta\left(\Theta_{K_{1\ldots6},P_1,P_2}\left(\underline{\mathbf{K}}\left[\underline{\mathbf{R}}_j|\underline{T}_j\right]\mathbf{X}_i\right)\right)\right|^2. \tag{2.97}$$

2.6.4 Calibration with planar targets

Calibration from a planar target can be performed with targets containing known calibration patterns and those containing an arbitrary texture. Both

methods require multiple views of the target from different perspectives. They are described in the following sections.

Calibration with a planar target with calibration pattern

Calibration from a planar target is based on the fact that every plane has two distinct points, which are the circular points, and the circular points of all planes in 3D space lie on the absolute conic $\underline{\mathbf{\Omega}}_\infty$ [48, 110]. The images of the absolute points thus lie on the image of the absolute conic, which only depends on the calibration matrix $\underline{\omega} = \left(\underline{\mathbf{K}}\underline{\mathbf{K}}^T\right)^{-1}$ [48].

In Section 3.1.1 it is shown that a point lying on a plane is projected into a camera by a homography. Therefore, the image of the absolute points of the plane is found in the j'th view by

$$\mathbf{c}_{1,2}{}^j = \underline{\mathbf{H}}_j \begin{pmatrix} 1 \\ \pm i \\ 0 \end{pmatrix} = \underline{\mathbf{h}}_j^1 \pm i\underline{\mathbf{h}}_j^2 \,, \tag{2.98}$$

where $\underline{\mathbf{h}}_j^{1,2}$ are the first and second columns of $\underline{\mathbf{H}}_j$, and i is the imaginary unit. Since the images of the absolute points lie on the image of the absolute conic, it is found that [110]

$$\left(\underline{\mathbf{h}}_j^1 \pm i\underline{\mathbf{h}}_j^2\right)^T \underline{\omega} \left(\underline{\mathbf{h}}_j^1 \pm i\underline{\mathbf{h}}_j^2\right) = 0 \,, \tag{2.99}$$

which can be split into real and complex parts

$$\left(\underline{\mathbf{h}}_j^1\right)^T \underline{\omega} \left(\underline{\mathbf{h}}_j^2\right) = 0 \tag{2.100}$$

$$\left(\underline{\mathbf{h}}_j^1\right)^T \underline{\omega} \left(\underline{\mathbf{h}}_j^1\right) = \left(\underline{\mathbf{h}}_j^2\right)^T \underline{\omega} \left(\underline{\mathbf{h}}_j^2\right) \,. \tag{2.101}$$

It is assumed that the plane contains a known calibration pattern, which can be found in the images. The pattern is used to find point correspondences, which in turn are used to find the homographies between the plane and the images. In [110] Equation 2.101 is converted to a problem linear in the entries of $\underline{\omega}$. It is solved by SVD, which is similar to the DLT approach in Section 2.6.1. Finally, $\underline{\mathbf{K}}$ is extracted from $\underline{\omega}$.

Until now, the lens distortion was not taken into account. In [110] lens distortion is included by minimizing the geometric error

$$\min_{\underline{\mathbf{K}},K_{1\ldots6},P_1,P_2,\underline{\mathbf{R}}_j,\underline{T}_j} \sum_j \sum_i \left| \underline{m}_i - \delta \left(\Theta_{K_{1\ldots6},P_1,P_2} \left(\underline{\mathbf{K}} \left[\underline{\mathbf{R}}_j | \underline{T}_j \right] \mathbf{X}_i \right) \right) \right|^2 . \tag{2.102}$$

To do so, $\underline{\mathbf{R}}_j$ and $\underline{T}_j$ are found from $\underline{\mathbf{K}}$ and $\underline{\mathbf{H}}_j$ [71].

The method that combines the estimation of linear camera parameters with a following optimization is commonly referred to as Zhang's method [110] and is probably the most used method for camera calibration [56]. Variants of it [9] are implemented in the OpenCV [12] and MATLAB [99] libraries. Since the target is often a board with a printed pattern, which is moved around the camera, the method is referred to as the movable planar target (MPT) method in the following.

Calibration with a planar target with arbitrary pattern

Calibration from planar scenes that do not contain a known calibration pattern is similar to the previous method. The difference is that homographies cannot be estimated between planes and cameras, but only between cameras.

A 3D point $\mathbf{X}$ that is located on a plane can be described by 2D coordinates $\mathbf{x}$. Its image in the j'th view is found by a homography $\mathbf{x}^j = \underline{\mathbf{H}}_j \mathbf{x}$. Points that lie on a plane are transformed from one view to another by a plane induced homography, as is shown in Section 3.1.1. That also means that if a plane is photographed from two perspectives by moving the plane, the image of $\mathbf{X}$ is transformed from the j'th to the k'th view by the plane induced homography $\mathbf{x}^k = \underline{\mathbf{H}}_{j\to k} \mathbf{x}^j$.

Suppose the plane's circular points in the metric frame $\mathbf{c}_{1,2} = (1, \pm i, 0)^T$ are projected into a reference camera by a plane homography

$$\mathbf{c}_{1,2}{}^0 = \underline{\mathbf{H}}_0 \begin{pmatrix} 1 \\ \pm i \\ 0 \end{pmatrix} , \tag{2.103}$$

and homographies $\underline{\mathbf{H}}_{0\to j}$ have been estimated, which relate every other view to the reference view, then the circular points are found in those views by

$$\mathbf{c}_{1,2}{}^j = \underline{\mathbf{H}}_{0\to j} \mathbf{c}_{1,2}{}^0 . \tag{2.104}$$

The images of the circular points lie on the image of the absolute conic, hence

$$\left(\underline{\mathbf{H}}_{0\to j}\mathbf{c}_{1,2}{}^{0}\right)^{T}\underline{\omega}\left(\underline{\mathbf{H}}_{0\to j}\mathbf{c}_{1,2}{}^{0}\right)=0\,. \tag{2.105}$$

This equation can be solved by numerical optimization methods for the image of the circular points in the reference view and for the entries of $\underline{\omega}$, given sufficiently many $\underline{\mathbf{H}}_{0\to j}$ [100]. $\underline{\mathbf{K}}$ is recovered from $\underline{\omega}$, and with its help $\underline{\mathbf{R}}_j$ and $\underline{T}_j$ are recovered from $\underline{\mathbf{H}}_{0\to j}$ [71].

A distortion model with two radial coefficients is incorporated in the planar self-calibration problem in [52]. The starting point is the optimization of Equation 2.105. Then, the projective camera model for all cameras is set-up, and the geometric error is optimized by bundle adjustment (Equation 2.97).

Another plane based self-calibration approach is presented in [67]. There, the relationship between the images of a plane, which is enforced by the plane induced homographies, is maintained by first optimizing a distortion model. Afterward, the in- and extrinsic parameters are estimated, as was explained in this section. A similar approach is presented in [97].

3 Stereoscopic 3D Profile Reconstruction of Low-Textured Slanted Planes

In order to draw conclusions about the condition of a road surface through its deformation, its profile shall be extracted from stereoscopic images through a dense reconstruction. In this chapter, two new methods for performing depth estimation in this particular case are presented. Although they were developed with the application of road profile reconstruction in mind, they can be applied to the general case, where the underlying geometry is a low-textured, slanted plane. Parts of this chapter have previously been published in [16] and [18].

If intrinsic camera parameters and the relative orientation of the individual cameras are known, the problem of depth estimation from stereo images can be broken down into matching pixels in a pair of images (Section 2.5). Most stereo algorithms follow four basic steps to solve it:

1. Since the search through image pixels on skewed lines is difficult, as pixels are aligned on a grid, the images of a stereo camera are rectified so that corresponding pixels are located on the same horizontal epipolar line (Section 2.5.1).

2. For each pixel of a reference image, a similarity measure for each pixel in a specified disparity range on the same horizontal line of the target image is calculated (Section 2.5.2). The similarity measure is often calculated on a window around the pixel of interest.

3. A smoothing term is introduced, which penalizes jumps in the disparity image (Section 2.5.4).

4. The similarity is maximized, while the smoothing term is minimized (Section 2.5.3).

That procedure creates several problems in the target application of road profile reconstruction, where the stereo camera system is placed behind the windshield of a vehicle, and the road surface is a low-textured, slanted plane:

1. For a high depth resolution, a large baseline is used in the stereo camera system. In this case, to cover the lane width with both cameras, the cameras must have convergent viewing directions. Rectification stretches the resulting images and reduces their quality.

2. The disparity is directly linked to the distance between an object and the cameras. A perfectly flat slanted plane has a broad range of disparity values. As a result, a broad range of disparity values has to be searched.

3. The pixels that correspond to a rectangular patch in an image are generally not arranged rectangularly in the other image. It rather depends on the underlying geometry. The compared patches, therefore, do not show the same area.

4. By penalizing jumps in disparity space, a fronto-parallel scene is implied. In the target application, it is known that the underlying geometry is a plane. Since the cameras are located behind the windshield, the plane is not fronto-parallel, but slanted to the cameras.

Other authors have addressed some of these problems. In [42] prior knowledge about surface normals is integrated into the optimization procedure, with surface normals being extracted directly from intensity images. The extraction of surface normals from intensity images is described in [28]. In [106] second-order smoothness priors are used to account for slanted surfaces. Second-order smoothness can also be encoded by using 3D labels as is described in [79] and applied in [68]. Although these methods generally allow for slanted surfaces, they do not favor any particular surface. However, in the task of road surface reconstruction, it can be assumed that the surface belongs to a single plane. A drawback of 3D labels is the enlarged label search space. In [59] the task of surface reconstruction is described. Disparity values are converted to elevations from a base plane, making it possible to penalize a change in adjacent pixels' elevations, but the location of the ground plane has to be known in advance. In [90] plane priors are used in order to pre-estimate disparity values for an arbitrary scene. Afterward, only a small range around that value is searched, but the smoothing term penalizes jumps in disparity space. The algorithm is not optimized for a single plane. A similar approach is described in [32], where road surface reconstruction is targeted. A seed and grow algorithm is used, where the disparity is first calculated at the bottom of the image and then propagated to the lines above. The smoothing term penalizes jumps in disparity space. Both latter methods use sparse image features to find the ground plane. In [89] a smoothing term is used that is dependent on a local plane hypothesis. That makes it possible to favor slanted surfaces. However,

the method works with discrete disparity values and cannot fully account for fractional surface slants. In [112], in contrast to the search through the disparity space, the search is performed through the discretized elevation of a base plane. As a result, jumps in elevation can be penalized. The location of the base plane is considered prior knowledge.

Except for [68], none of the methods above account for non-corresponding rectangular patches. Recently, in [86], a method was proposed that addresses this problem by first searching for dominant planes in a scene and then transforming one image into the other's space. A smoothing term similar to the one introduced in [59] is used. Addressing non-corresponding rectangular patches is especially important if the underlying scene is a highly slanted plane, or if motion blur occurs because this requires the comparison of large patches to make the matching robust.

Road surface reconstruction by stereo vision is a special application and requires a specialized algorithm. In this chapter, two examples are proposed. Since it is known that the underlying geometry is a plane, the plane-sweep approach is a natural choice to limit the search space and to take into account the similarity in the elevation of adjacent pixels. This is discussed in Section 3.1. Both developed algorithms make use of it.

The first proposed algorithm is based on traditional methods and searches for corresponding pixels in a stereo image pair. It is described in Section 3.2 and has previously been published in [18]. The second proposed algorithm solves the stereo correspondence problem in a sinlge step by using a CNN. It is described in Section 3.3 and has previously been published in [16]. Results that were obtained with both proposed methods are shown in Chapter 6.

3.1 Plane-sweep

Plane-sweep was first introduced in [25], where a virtual plane is swept through object space and checked for compliance with the images of at least two cameras. First, features are calculated in each image and are then back-projected into 3D space as rays. A segmented virtual plane is placed in space, and the rays hitting a segment are counted. The number of hits indicate that the plane is at the position of an object point because the rays should intersect or almost intersect at this position. The plane is then swept through space. In this way, the plane-sweep approach is used to match image features and determine the corresponding object point position simultaneously. The result is a sparse reconstruction.

In this thesis, virtual planes parallel to the mean road surface are swept through space and checked for compliance with the stereoscopic images, as

is shown in Figure 3.1. This is accomplished by warping an image of one camera onto the plane and into the other camera's space (Section 3.1.1). There, both images are compared by a similarity measure (Section 3.2.1). Since the underlying geometry is a plane, adjacent pixels are expected to have similar elevations measured from a mean road surface, which means they produce high similarity values on similar virtual planes. That is taken into account by a smoothing term in Section 3.2.2. In order to perform the plane-sweep, the location of the mean road surface is approximated at first in Section 3.1.3 and refined later on in Section 3.2.4. The plane-sweep approach solves the previously listed difficulties:

1. The images do not need to be rectified.
2. With a sweeping direction orthogonal to the mean road surface, the volume that has to be searched can be limited to a few centimeters above and below the mean road surface. The reduction of the search space is a key aspect of the proposed algorithm because it reduces the ambiguity when matching pixels between the left and right images.
3. Assuming that compared image patches lie on one of the plane hypotheses, the similarity measure is calculated on correctly transformed patches. That is because the image from one camera is transformed into the space of the other camera.
4. A smoothing term that penalizes jumps in elevation is implemented seamlessly.

In this section it is assumed that the intrinsic camera parameters and the relative orientation and translation of both individual cameras of the stereo camera setup are known, and that possible lens distortions have been removed from the images.

3.1.1 Plane induced homography

If two cameras picture the same plane, a view from one camera is transformed into the view of the other camera by a plane induced homography. Suppose the plane is the XY-plane. A point from that plane is transformed into camera coordinates by (Equation 2.15)

$$\mathbf{x} = \underline{\mathbf{P}} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} \underline{\mathbf{p}}^1 & \underline{\mathbf{p}}^2 & \underline{\mathbf{p}}^4 \end{bmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \underline{\mathbf{K}} \begin{bmatrix} \underline{\mathbf{r}}^1 & \underline{\mathbf{r}}^2 & \underline{T} \end{bmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}, \quad (3.1)$$

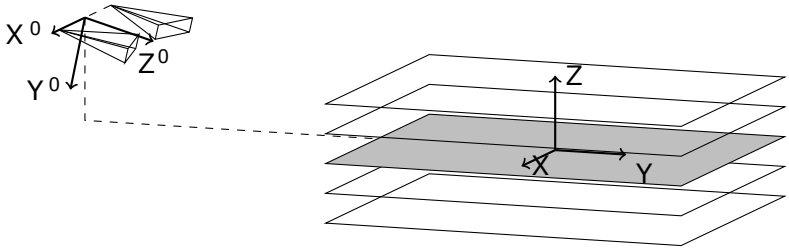


Figure 3.1: The virtual planes (unfilled rectangles) are swept in Z-direction around the average road surface (gray rectangle) located in the XY-plane of the road coordinate system.

with $\underline{T} = -\underline{\mathbf{R}}\underline{C}$ (Equation 2.13). $\underline{\mathbf{p}}^i$ is the i'th column of $\underline{\mathbf{P}}$, and $\underline{\mathbf{r}}^i$ is the i'th column of $\underline{\mathbf{R}}$. A point from a plane parallel to the XY-plane, but with distance Z, is transformed into image coordinates by shifting the camera center

$$\underline{T}_Z = -\underline{\mathbf{R}}\left(\underline{C} - \begin{pmatrix} 0 \\ 0 \\ Z \end{pmatrix}\right) = \underline{T} + \underline{\mathbf{r}}_3 Z\,. \tag{3.2}$$

A point on a plane parallel to the XY-plane with distance Z_i from the plane is projected into the left camera space by

$$\mathbf{x}_l = \underline{\mathbf{K}}_l \begin{bmatrix} \underline{\mathbf{r}}_l^1 & \underline{\mathbf{r}}_l^2 & Z_i\underline{\mathbf{r}}_l^3 + \underline{T}_l \end{bmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \underline{\mathbf{H}}_{l,i} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}, \tag{3.3}$$

where the index l denotes that the variables belong to the left camera. The transformation is a homography and can be inverted, which means an image can be projected from a camera view onto the plane.
A point from the left camera image is warped onto the plane with index i and elevation Z_i and into the right camera space by the plane induced homography

$$\underline{\mathbf{H}}_i = \underline{\mathbf{H}}_{r,i} \cdot \underline{\mathbf{H}}_{l,i}^{-1}\,, \tag{3.4}$$

where $\underline{\mathbf{H}}_{r,i}$ is the homography that maps from the plane into the right camera space.

3.1.2 Coordinate system

Two coordinate systems are defined, which are shown in Figure 3.1 The road coordinate system XYZ is placed such that the XY-plane coincides with the

mean road surface. The stereo camera system's coordinate system is set to the point in-between camera centers, such that the X^0-axis points to the right, the Y^0-axis down, and the Z^0-axis in the viewing direction. This helps to create an elevation map later on. It is accomplished by splitting the extrinsic parameters $\underline{\mathbf{R}}_{st}$ and $\underline{T}_{st}$, which relate the stereo camera heads to each other and are a result of stereo camera calibration, into two parts by the square root of a matrix

$$\begin{pmatrix} \underline{\mathbf{R}}_{st} & \underline{T}_{st} \\ 0 & 1 \end{pmatrix}^{\frac{1}{2}} = \begin{pmatrix} \underline{\mathbf{R}}_{r} & \underline{T}_{r} \\ 0 & 1 \end{pmatrix} , \tag{3.5}$$

where $\underline{\mathbf{R}}_r$ is the rotation matrix from coordinate origin to the right camera, and $\underline{T}_r$ is the corresponding translation vector. The location of the left camera is found by

$$\underline{\mathbf{R}}_l = \underline{\mathbf{R}}_r^{-1} \text{ and } \underline{T}_l = -\underline{T}_r \, . \tag{3.6}$$

The plane homographies (Equation 3.3) are then found by

$$\underline{\mathbf{H}}_{l,i} = \underline{\mathbf{K}}_l \begin{bmatrix} \underline{\mathbf{R}}_l & \underline{T}_l \end{bmatrix} \begin{bmatrix} \underline{\mathbf{r}}_P^1 & \underline{\mathbf{r}}_P^2 & z_i \underline{\mathbf{r}}_P^3 + \underline{T}_P \\ 0 & 0 & 1 \end{bmatrix} , \tag{3.7}$$

with the rotation matrix $\underline{\mathbf{R}}_P = \begin{bmatrix} \underline{\mathbf{r}}_P^1 & \underline{\mathbf{r}}_P^2 & \underline{\mathbf{r}}_P^3 \end{bmatrix}$ and translation vector $\underline{T}_P$, which relate the plane to the coordinate origin. This is equivalently done for $\underline{\mathbf{H}}_{r,i}$.

3.1.3 Mean surface approximation

In order to perform the plane-sweep, the location of a base plane, i. e. $\underline{\mathbf{R}}_P$ and $\underline{T}_P$, must be known. Therefore, the location of the mean road surface is approximated at first and refined later on. With the approximate location, dense reconstruction, as is described in Section 3.2, is performed, and a 3D point cloud is generated. A new mean road surface can then be found in that point cloud. These steps are repeated until convergence. There are two ways of determining the initial approximate location of the plane.

1. A sparse point cloud can be estimated from image features by triangulation. The mean surface is then found in the point cloud, as described in Section 3.2.4. Since robust features are hard to find on road surfaces, this method often fails. In a video sequence, it can still be used by waiting for an image pair with rich features.

2. Since the cameras are fixed in the vehicle, the relation between cameras and the road surface can be measured.

The approximate location only needs to be found once per stereo camera setup. It can be reused for subsequent images. However, since the cameras' position in relation to the base plane constantly changes due to the suspension of the vehicle, the refinement step must be repeated for each pair of images.

3.2 Plane-sweep for dense reconstruction with traditional methods

To perform dense reconstruction, the entire left camera image I_l is transformed to the space of the right camera. It is performed by inverse warping (Section 2.4) according to the plane homographies $\underline{\mathbf{H}}_i$ (Equation 3.4) for each plane hypothesis in question. The mapping function is

$$\underline{m}_l = \phi_{\underline{\mathbf{H}}_i}(\underline{m}_r) = \delta\left(\underline{\mathbf{H}}_i^{-1}\delta^{-1}(\underline{m}_r)\right), \tag{3.8}$$

and it is used to transform pixels from the left image to the space of the right camera

$$I_{lW,i} = \left\{ I_l\left[\phi_{\underline{\mathbf{H}}_i}(\underline{m}_r)\right] \mid \underline{m}_r \in \mathcal{M}_r \right\}. \tag{3.9}$$

If the virtual plane is at the true location in 3D space for parts of the images, these parts match in the warped left and unchanged right image. The right camera image therefore is the reference image. For each pixel $\underline{m}$, a virtual plane must be identified, for which this is the case. The identified plane index i for a specific pixel $\underline{m}$ is the label $o_{\underline{m}}$, and the set of all $o_{\underline{m}}$ is the label image O. With the true O, a new image I^*_{lW} can be assembled from $I_{lW,i}$, which perfectly resembles I_r.

Due to the ambiguity problem (Section 2.5.3), to obtain a robust estimate for O, the optimization problem from Equation 2.67 must be solved. The required data and smoothing terms are defined in Sections 3.2.1 and 3.2.2.

The homographies can be calculated for virtual planes of any elevation. It enables the search for correspondences in non-integer pixel coordinates, since the transformed images are obtained by interpolation. That is in contrast to a search through disparity space, where subpixel accuracy is reached through interpolation between disparity values.

3.2.1 Data term

The (dis-)similarity of pixel values is compared by the methods discussed in Section 2.5.2. Their application, in combination with the plane-sweep approach, is discussed in this Section. To emphasize that the data term is a function of

the plane index i at pixel $\underline{m}$

$$\bar{E}_D(i_{\underline{m}}) \coloneqq E_D(i_{\underline{m}}, y_{\underline{m}}) , \tag{3.10}$$

where the explicit representation of the data y is omitted.

BilSub Bilateral filtering and subtraction are performed before warping of the left image in order to prevent repetitive calculations. The background-subtracted left image I_l^{BS} is warped and a patch around a pixel of $I_{lW,i}^{\mathrm{BS}}$ is compared to the corresponding patch around the pixel of I_r^{BS} by

$$\bar{E}_D(i_{\underline{m}}) = \sum_{\underline{n} \epsilon \mathcal{N}_{\underline{m}}} \left| I_{lW,i}^{\mathrm{BS}}[\underline{n}] - I_r^{\mathrm{BS}}[\underline{n}] \right| . \tag{3.11}$$

Census Warping of the left image is performed first. Then, the Census transform is applied on a patch around each pixel of each $I_{lW,i} \mapsto I_{lW,i}^{\mathrm{CS}}$ and on a patch around each pixel of $I_r \mapsto I_r^{\mathrm{CS}}$.

It is applied after warping to account for the perspective transformation of the patches. The hamming distance is used to measure the similarity between the bit strings that belong to each pixel of every $I_{lW,i}^{\mathrm{CS}}$ and those that belong to each corresponding pixel of I_r^{CS}

$$CS(\underline{m}, i) = d_{HD}\left(I_{lW,i}^{\mathrm{CS}}[\underline{m}] , I_r^{\mathrm{CS}}[\underline{m}] \right) , \tag{3.12}$$

where $d_{HD}(\cdot,\cdot)$ is the hamming distance. To make the similarity measure more robust against noise, a rectangular patch is then summed around each pixel

$$\bar{E}_D(i_{\underline{m}}) = \sum_{\underline{n} \epsilon \mathcal{N}_{\underline{m}}} CS(\underline{n}, i) . \tag{3.13}$$

HMI With the correctly assigned plane from Equation 3.8, for each pixel a mapping $\phi : \underline{m}_r \to \underline{m}_l$ is found, which maps the pixels from the right image to the space of the left image. The correctly transformed left image in the right image space is

$$I_{lW} = \left\{ I_l[\phi(\underline{m}_r)] \mid \underline{m}_r \in \mathcal{M}_r \right\} . \tag{3.14}$$

With this, the pixel-wise MI in Equation 2.60 is written as

$$\begin{aligned} \bar{mi}(\underline{m}, i) \coloneqq mi_\phi(\underline{m}) = & h_{I_r}^{\mathrm{lookup}}(I_r[\underline{m}]) + h_{I_{lW}}^{\mathrm{lookup}}(I_{lW,i}[\underline{m}]) \\ & - h_{I_r, I_{lW}}^{\mathrm{lookup}}(I_r[\underline{m}], I_{lW,i}[\underline{m}]) . \end{aligned} \tag{3.15}$$

Just as was done for BilSub and Census, for robustness, it is summed on a patch around each pixel. Since the MI value increases as the compared images become more similar, its negative value is used as the energy

$$\bar{E}_D(i_{\underline{m}}) = -\sum_{\underline{n}\epsilon\mathcal{N}_{\underline{m}}} \bar{mi}(\underline{n}, i)\,. \tag{3.16}$$

Problematic is that $h^{\text{lookup}}_{I_r}$, $h^{\text{lookup}}_{I_{lW}}$ and $h^{\text{lookup}}_{I_r,I_{lW}}$ have to be known to calculate $\bar{mi}(\underline{m}, i)$, but $h^{\text{lookup}}_{I_{lW}}$ and $h^{\text{lookup}}_{I_r,I_{lW}}$ are calculated from I_{lW}, which is based on the correct mapping. The mapping, in turn, is the result of the optimization. The solution is to start with a mapping that corresponds to the base plane, i. e. it is initialized with $Z_i = 0$ for each pixel. Then, the optimization and assembly of I_{lW} are alternated until convergence.

For computational efficiency, HMI uses an image pyramid of downscaled input images [53]. It starts with the lowest resolution. A mapping is calculated and upscaled for use with the next level of the pyramid. In this work, the refinement of the mean road surface location (Section 3.2.4) is integrated into the process. Therefore, the mapping is warped according to the new location and upscaled afterward.

3.2.2 Smoothing term

The goal is to estimate an elevation for each pixel in the stereoscopic image. The elevation is represented by the index i that determines the elevation Z_i of the corresponding plane. Overall, the road surfaces are expected to be piecewise continuous. For this reason, the discontinuity preserving smoothing term

$$E_{NB}(i_{\underline{m}_j}, i_{\underline{m}_k}) = K_s \left| i_{\underline{m}_j} - i_{\underline{m}_k} \right| \tag{3.17}$$

is used, which was discussed in Section 2.5.4. The factor K_s is chosen depending on the size of $\bar{E}_D(i_{\underline{m}})$ and $E_{NB}(i_{\underline{m}_j}, i_{\underline{m}_k})$, in order to bring both energies to the same order of magnitude. K is not changed across images depending on pixel data, as is commonly done (e. g. in [68]) because, in the target application, a change in color or intensity is not necessarily related to a change in elevation. That can be seen in Figure 6.7, where some leaves are squeezed onto the surface and are completely flat. Another example of this is shadows, which also do not involve a change in elevation.

3.2.3 Label image, elevation image, point cloud, and elevation map

The SGM algorithm is used to approximate

$$o_{\underline{m}} = \arg\min_{i_{\underline{m}}} \sum_j \bar{E}_D(i_{\underline{m}_j}) + \sum_{i,j \in \mathcal{N}_{NB}} E_{NB}(i_{\underline{m}_j}, i_{\underline{m}_k}) \,. \tag{3.18}$$

The result is an assigned plane for each pixel of the right camera image, and every plane corresponds to an elevation. The result therefore is a label image from the right camera's perspective. By exchanging the labels for the corresponding elevation, it is transformed into an elevation image.

To find the 3D point coordinates that belong to each pixel, image coordinates are transformed to plane coordinates $\underline{m}_P$ by the appropriate inverse plane homography

$$\underline{m}_P = \delta \left(\underline{\mathbf{H}}_{r,i}^{-1} \delta^{-1} \left(\underline{m}_r \right) \right) \,, \tag{3.19}$$

and the 3D world coordinates are found by adding the elevation of the corresponding plane

$$\underline{M} = \begin{pmatrix} \underline{m}_P \\ Z_i \end{pmatrix} \,. \tag{3.20}$$

The point cloud can easily be converted to an elevation map. For this a 2D array is created, which is filled with the elevation values Z_i at the coordinates $\underline{m}_P$. Since $\underline{m}_P$ in general are non-integer values, the array is filled with interpolated values Z_i at integer coordinates.

3.2.4 Mean surface refinement

The location of the cameras in relation to the mean road surface must be known to perform the plane-sweep. First, an estimate is used, and the dense elevation image reconstruction is performed. The plane-sweep thereby must cross at least parts of the real road surface to be successful. The parts of the road that do not lie within the volume that is searched during the plane-sweep cannot be reconstructed and appear as noisy regions in the elevation image. They are filtered with a local variance filter with a threshold. The 3D coordinates of the valid pixels are found, and a plane is fitted to the resulting point cloud, as described in the following.

Let $\mathbf{M}$ be the data matrix, where each row represents the 3D coordinates of a point $\underline{M}_i$, and $\tilde{\mathbf{M}}$ be the data matrix of the same point cloud, where the sample mean $\underline{\bar{M}}$ has been subtracted. The principal component analysis (PCA) finds a

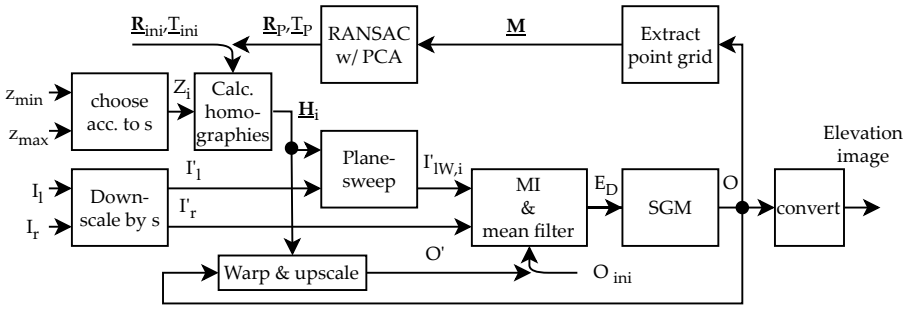


Figure 3.2: Overview of the plane-sweep algorithm. Multiple iterations are used to find the location of the base plane.

matrix $\underline{\mathbf{W}}$ that rotates $\underline{\bar{\mathbf{M}}}$

$$\underline{\bar{\mathbf{M}}}_{\mathrm{PCA}} = \underline{\bar{\mathbf{M}}}\underline{\mathbf{W}}\,, \tag{3.21}$$

such that the principal axes are aligned with the coordinate system, i. e. the greatest variance of $\underline{\bar{\mathbf{M}}}_{\mathrm{PCA}}$ is aligned with the X-axis, the second largest variance is aligned with the Y-axis and the smallest variance with the Z-axis [7]. In the case of a point cloud that has the shape of a plane, that means the XY-plane coincides with a mean plane that is fitted to the rotated point cloud. In Equation 3.7 the inverse of $\underline{\mathbf{W}}$ rotates plane coordinates into camera coordinates. Since $\underline{\mathbf{W}}$ is a rotation matrix, the inverse is replaced by the transposed

$$\underline{\mathbf{R}}_P = \underline{\mathbf{W}}^T\,. \tag{3.22}$$

The center of the stereo rig is found by the negative sample mean vector. The translation vector $\underline{T}_P$ is found by (Equation 2.12)

$$\underline{T}_P = \underline{\mathbf{R}}_P\underline{\bar{M}}\,. \tag{3.23}$$

The approximation is made more robust by using the random sample consensus (RANSAC) algorithm [38]. It uses the following steps to fit an arbitrary model to data points:

1. As many points are randomly selected as are needed to fit the model.
2. The model is fitted.
3. All points are checked to be within a defined limit of the model.
4. Steps 1 - 3 are repeated. The model with the highest number of inliers is chosen.

(5. Optionally, the model can be reestimated with the inliers of the best model.)

In the given case, three points are required in step 1, and the model is fitted by PCA. The check for compliance of points with the model is performed by checking the absolute value of the last coordinate of the transformed point cloud $\underline{\mathbf{M}}_{\mathrm{PCA}}$, which is the distance from the mean plane. PCA is repeated with all points that are within the defined limit.

Since the plane found by PCA can be arbitrarily rotated around the Z-axis, the rotation matrix is disassembled into Tait-Bryan angles, which describe three sequential rotations around the X-axis, Y-axis, and Z-axes. Rotation matrices perform the sequential rotations

$$\underline{\mathbf{R}}_P = \underline{\mathbf{R}}_{P,z}\underline{\mathbf{R}}_{P,y}\underline{\mathbf{R}}_{P,x}\,. \tag{3.24}$$

The rotation around the Z-axis is ignored, and the rotation matrix is reassembled

$$\underline{\mathbf{R}}_P = \underline{\mathbf{R}}_{P,y}\underline{\mathbf{R}}_{P,x}\,. \tag{3.25}$$

That ensures that the point cloud $\underline{\mathbf{M}}$, which consists of the road, is oriented in the correct direction. It is also possible that the normal of the plane found by PCA points downwards. In this case, an additional rotation by 180° around the x-axis is applied to the rotation matrix. Otherwise, the plane-sweep in a next iteration would take the wrong direction.

3.2.5 Algorithm overview

Figure 3.2 gives an overview of the entire plane-sweep algorithm with hierarchical mutual information as the similarity measure and shows the refinement of the plane position. The input images are downscaled by the factor s in both dimensions ("downscale by s"), and the positions of the plane hypotheses are chosen according to the reduced resolution ("Choose acc. to s"). The location of the mean road surface in relation to the stereo camera system is given by $\underline{\mathbf{R}}_P$ and $\underline{T}_P$. The virtual planes are placed below and above of it, and plane homographies are calculated ("calc. homographies"). The plane-sweep is performed with the left camera image ("plane-sweep") using the plane homographies. In order to calculate the MI similarity, the final label image is required, i. e. the plane index for each pixel. It is a result of the following SGM optimization ("SGM") and is found iteratively by alternating the SGM optimization and the MI calculation. At the same time, a point grid is extracted from the label image ("extract point grid") and the position of the road is refined by RANSAC with PCA ("RANSAC w/ PCA"). Since the scaling of the images is adjusted and

the estimated location of the mean road surface changes, the label image used to calculate the MI must be adjusted accordingly. This is done by warping and upscaling the label image by a homography ("warp & upscale"). The MI calculation is followed by summing over rectangular windows ("MI & mean filter"). By converting the indices to heights ("convert"), the label image is converted to an elevation image.

3.2.6 Distinction to other methods

The plane-sweep approach is used in several other publications for dense reconstruction [41, 107, 112]. In [107] multiple images are warped onto virtual planes, which are swept in depth direction away from a camera rig that holds multiple cameras. However, the focus lies on a real-time implementation for a multi-camera setup, and a term for smooth surfaces is not implemented. The search volume has to be known a-priori. Similarly, in [19], the plane-sweep approach is used for multi-image matching, with the sweeping direction being in the reference camera's viewing direction. In contrast to the method mentioned before, a smoothing term is implemented, and a global optimization algorithm is used. The search volume must again be known a-priori.

In [41] the dense plane-sweep approach is extended by the creation of plane hypotheses from triangulated feature points. In contrast to [107], the warped images are not compared on virtual planes, but in the camera space. Plane-sweeps are conducted orthogonally to the plane hypotheses, and by a winner takes it all principle, a best-fitting plane for each sweeping direction is chosen for each pixel. A term for smooth surfaces is not implemented in this step. Afterward, the best fitting sweeping direction is found by penalizing changes of surface normals of neighboring pixels. Due to the multiple sweeping directions, it is especially useful for scenes that contain multiple base planes. In case of only one base plane, as in this work, the approach is not expedient because the choice of a best sweeping direction is unnecessary and because of the lack of a smoothing term.

In [58] the plane-sweep approach is used for multi-image matching of aerial images in the second step of a structure from motion (SfM) pipeline. The algorithm depends on scale invariant feature transform (SIFT) feature points, which are used to compute camera poses and a sparse point cloud. The viewing direction of the reference camera is chosen for plane-sweeping. As a smoothing term, total variation is used, which enables the use of the variational optimization technique from Section 2.5.6. The volume to be searched is given by elevation data, which is specified in advance.

The work in [112] was performed concurrently with this thesis and indeed is similar. Still, there are some significant differences. A plane-sweep approach,

in combination with the SGM optimization, is performed. Also, the change in elevation of neighboring pixels is penalized, and the Census transform is used for matching pixels. The difference is that in [112], the location of the base plane is known a-priori. Therefore, the search and refinement of the base plane are not performed. Furthermore, the plane-sweep approach is used to find the location of possibly matching pixels across multiple images. The plane homography is not used to transform entire images, but only for matching single pixels. The matching cost, however, is calculated across rectangular windows across every single image. Thus, the perspective, which changes the shape of a rectangular patch that is transformed into another camera space, is not considered.

The target in [32] is road profile reconstruction from stereoscopic images, but the approach is different. Rectified images are considered, one of which is transformed by a plane homography to the other image by a slanted plane, located at the mean road surface. Therefore, the perspective transformation of rectangular matching windows is partially taken into account. Afterward, only a small range of disparity values has to be searched. Subpixel accuracy is achieved by interpolating the best matching disparities from adjacent pixels. A smoothing term penalizes jumps in disparity space, where the disparity space has been corrected for the slanted plane. Optimization is performed by a seed and grow algorithm. The base plane is found by the triangulation of feature points. The most significant difference to the proposed method is the search through disparity space in contrast to a search through elevations from the base plane. The search through disparity space makes it necessary to perform interpolation to achieve subpixel accuracy.

The proposed method uses the plane-sweep approach and is capable of transforming images by plane homographies at arbitrary positions. Therefore, subpixel accuracy is achieved without further processing steps. Interpolation of disparity values between adjacent pixels is not necessary. In fact, interpolation of pixel intensities is performed during the transformation of images, and interpolated images are compared to the reference image. In combination with the inherent consideration of the perspective while matching patches, this property is the main advantage of using plane-sweep in the proposed way. Furthermore, in [32], a stereo camera with a baseline of only 120 mm is used, and the matching cost is the normalized cross-correlation (NCC). The small baseline prevents radiometric differences between images. Although not shown in this thesis, own experiments with NCC confirm the findings from [55] that better-suited similarity measures than NCC exist.

3.3 Plane-sweep for dense reconstruction by one-step CNN

In Section 2.5.2 measures for comparing similarities between single pixels and pixel patches were discussed. A very efficient one is a convolutional neural network that is reported to outperform traditional methods, like the sum of absolute differences, Census transform, and normalized cross-correlation [109]. That raised the idea of integrating the cost aggregation and the optimization step into a single neural network. Different architectures thereof exist. In [30] a network for estimating optical flow is proposed, where a region around a pixel in consecutive frames of a video sequence needs to be traced. If the left and right rectified images of a stereo camera are used as input, the optical flow is the disparity. Thus, the idea was modified for use with stereoscopic images. In [73] a correlation layer is used to account for the epipolar geometry. In [63] the idea of a cost volume is introduced. First, the network extracts feature maps from the left and the right image. Then, the feature map from one image is copied multiple times. The copies are shifted according to different disparity levels and are stacked on top of the other image's feature map. This approach embeds the epipolar geometry. In [91] a similar network with a semi-supervised training procedure is implemented. It runs almost in real-time.

Besides Flownet from [30], all the methods mentioned above search through disparity space. In this chapter, the plane-sweep approach is integrated into a neural network. Instead of warping the raw images, feature maps are created first and then transformed by plane homographies. That combines the useful properties of plane-sweep, like the constraint search volume and the correct perspective transformation, with the good performance of a neural network.

3.3.1 Neural network for 3D surface profile reconstruction

The plane-sweep approach can be implemented in all networks based on the idea of a cost volume. Here, the work from [23] is extended by plane-sweep, as it can exploit global context information. By creating image features with the help of spatial pyramid pooling, region-level features are introduced [23]. As roads have little texture, it is believed that region-level features will improve the overall performance, especially if cracks or other contexts are visible. Figure 3.3 gives an overview of the network architecture. It consists of five basic blocks:

1. A Siamese network extracts feature maps from the left and right input images.

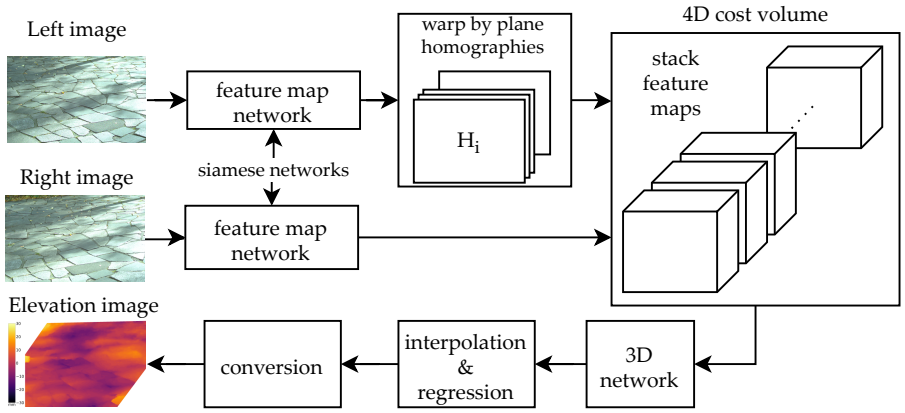


Figure 3.3: Overview of the CNN for 3D surface profile reconstruction. A Siamese network calculates feature maps for both input images. Plane homographies transform the one belonging to the left image, and a 4D cost volume is generated. Matching is performed by the 3D network that outputs the elevation image.

2. The feature maps of the left image are transformed according to plane hypotheses.
3. Transformed feature maps of the left image are stacked on top of the right image's feature map, which results in a 4D cost volume.
4. A 3D network transforms the 4D cost volume into a 3D cost volume.
5. An interpolation and regression block finds the correct plane index from the plane hypotheses. The output of the network is a label image.The label image needs to be converted to the final elevation image.

The network thus implements a procedure similar to that described in Section 3.2. The relationship is explained in the following. To find the correct plane from the set of plane hypotheses, in Section 3.2, image patches are compared by one of the hand-crafted similarity measures. These correspond to the features calculated by the neural network, but in contrast to the previous method, the neural network can learn multiple features and use them concurrently. In Section 3.2, images are transformed. The network transforms feature maps instead. Due to the multiple features, the previously 3D cost volume is turned into a 4D cost volume. The 3D network is built as a classifier that uses the features and gives a class probability for each pixel. The classes are the plane indices. The interpolation and regression block finds the most probable plane

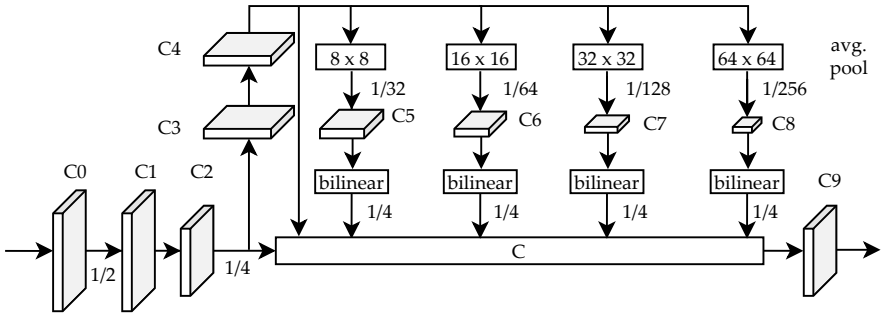


Figure 3.4: Detailed view of the feature extraction network. Refer to Table 3.1 for the convolutional layers. The current size of the intermediate feature maps is specified as a fraction of the input resolution.

index for every input pixel coordinate, which corresponds to the calculation and optimization of an energy function that considers a smoothing term. The single blocks are explained in more detail in the next sections.

Feature extraction

The feature extraction network [23] is based on a spatial pyramid pooling module [49]. It is supposed to create features that describe the fine structure and features that describe the coarse structure. In Figure 3.4 the network structure is shown. Convolutional layers are shown as boxes. Their sizes denote the spatial dimension of the filter outputs. The implemented operations of the convolutional layers are given in Table 3.1. The second column lists the properties of the convolutional filters, i. e. the filter kernel size, the number of output features, the stride if it is present, and the dilation (dila). The third column shows all the operations that are performed in the layer: "conv" is the convolutional layer, "bn" is the batch-norm operation, "ReLU" is the rectified linear unit. Two rows that are enclosed by square brackets are executed consecutively. This is repeated $[] \times y$ times, while the stride is applied only the first time. $+\,in$ denotes the addition of the input of the layer to its output, i. e. it denotes a residual block.

After operation C2, the feature map resolution is reduced to $W/4 \times H/4$ by convolutional layers. After operation C4, it is further scaled down to different resolutions by average pooling. At the different scalings, convolutional layers are applied. The outputs, therefore, contain both the fine and the coarse structure. Afterward, all feature maps are upscaled to the same resolution by bilinear interpolation to make a concatenation possible. In the last step,

Table 3.1: Operations that are used in the feature extraction network.

Name	Convolution	Operation
C0	$[3 \times 3, 32, \text{stride} = 2] \times 3$	[conv, bn, ReLU] $\times 3$
C1	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 3$	$\begin{bmatrix} \begin{bmatrix} \text{conv, bn, ReLU} \\ \text{conv, bn} \end{bmatrix} + in \end{bmatrix} \times 3$
C2	$\begin{bmatrix} 3 \times 3, 64, \text{stride} = 2 \\ 3 \times 3, 64 \end{bmatrix} \times 16$	$\begin{bmatrix} \begin{bmatrix} \text{conv, bn, ReLU} \\ \text{conv, bn} \end{bmatrix} + in \end{bmatrix} \times 16$
C3	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 3$	$\begin{bmatrix} \begin{bmatrix} \text{conv, bn, ReLU} \\ \text{conv, bn} \end{bmatrix} + in \end{bmatrix} \times 3$
C4	$\begin{bmatrix} 3 \times 3, 128, \text{dila} = 2 \\ 3 \times 3, 128, \text{dila} = 2 \end{bmatrix} \times 3$	$\begin{bmatrix} \begin{bmatrix} \text{conv, bn, ReLU} \\ \text{conv, bn} \end{bmatrix} + in \end{bmatrix} \times 3$
C5-C8	$[1 \times 1, 32]$	[conv, bn, ReLU]
C9	$\begin{bmatrix} 3 \times 3, 128 \\ 1 \times 1, 32 \end{bmatrix}$	$\begin{bmatrix} \text{conv, bn, ReLU} \\ \text{conv} \end{bmatrix}$

the final features are created by operation C9. The output has the resolution $W/4 \times H/4$.

Plane-sweep and cost volume assembly

The plane-sweep approach is implemented in the neural network by transforming the feature maps of the left input image according to the plane hypotheses, as was done with the input images in Section 3.2. In order to identify the correct plane for each input pixel, the transformed left and unchanged right feature maps are compared. For this purpose, a cost volume is assembled. The feature map network extracts feature maps of dimension $W/4 \times H/4 \times Z$ per input image, with the number of features Z. For each plane hypothesis, the feature map of the left input image is transformed and stacked on top of the feature map of the right input image, building 3D volumes (see Figure 3.3). Then, the 3D volumes are concatenated to a 4D volume of dimension $W/4 \times H/4 \times P \times 2Z$, with the number of plane hypotheses P. Since the feature maps have a reduced resolution, the camera matrices used to calculate the homographies, have to be scaled accordingly

$$\underline{\mathbf{K}} = \begin{pmatrix} f_x/4 & s/4 & u_0/4 \\ & f_y/4 & v_0/4 \\ & & 1 \end{pmatrix}. \tag{3.26}$$

The transformation of the feature map is realized by inverse warping (Section 2.4). Therefore, a lookup table is calculated that stores a source location for each target pixel. The source location is rounded to the nearest integer

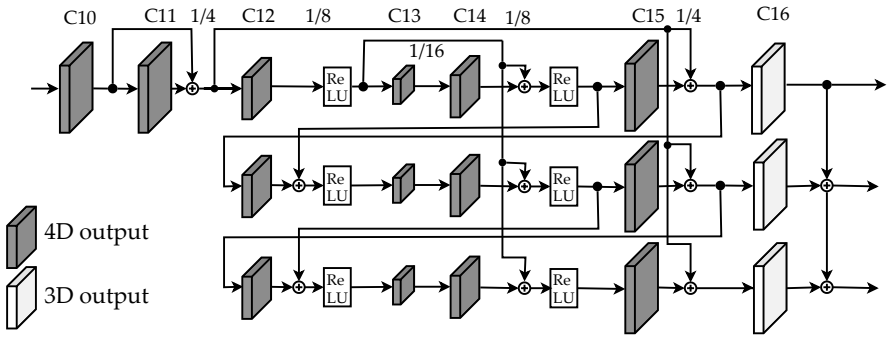


Figure 3.5: Detailed view of the 3D network. Refer to Table 3.2 for the convolutional layers. The current size of the intermediate feature maps is specified as a fraction of the input resolution. All but the last shown arrays are 4D volumes.

coordinate. Instead of rounding, bilinear interpolation could be integrated into neural networks, as in spatial transformer networks [60]. Unfortunately, experiments showed that training of the neural network fails if interpolation is used. The neural network has a large receptive field, which, in combination with interpolation, presumably breaks the relation between an image pixel and its corresponding value in the feature map. However, bilinear interpolation can be used for the evaluation of the network.

3D network

After feature extraction has been performed, and the cost volume has been assembled (dimension $W/4 \times H/4 \times P \times 2Z$), the best fitting plane for each input pixel needs to be found. Therefore, the feature dimension is reduced, so that the overall dimension is $W/4 \times H/4 \times P \times 1$. The plane dimension thereby maintains the compatibility of the plane hypotheses with the data. It is implemented in a 3D network [23], and it is shown in Figure 3.5. The convolutional layers, which are denoted by boxes, are described in Table 3.2. "deconv" is short for deconvolution. The size of the boxes denote the spatial resolution of the layer's output. Dark boxes output 4D volumes and light boxes output 3D volumes. "ReLU" blocks denote additional rectified liniear units. The network uses a stacked hourglass architecture [78] in order to encode context information. Each of the three hourglasses consist of encoder-decoder structures and generate a $W/4 \times H/4 \times P \times 1$ sized output $\underline{\mathbf{O}}$.

Table 3.2: Operations that are used in the 3D network.

Name	Convolution	Operation
C10	$[3 \times 3 \times 3, 32] \times 2$	[conv, bn, ReLU] $\times 2$
C11	$\begin{bmatrix} 3 \times 3 \times 3, 32 \\ 3 \times 3 \times 3, 32 \end{bmatrix}$	$\begin{bmatrix} \text{conv, bn, ReLU} \\ \text{conv, bn} \end{bmatrix}$
C12	$\begin{bmatrix} 3 \times 3 \times 3, 64, \text{stride} = 2 \\ 3 \times 3 \times 3, 64 \end{bmatrix}$	$\begin{bmatrix} \text{conv, bn, ReLU} \\ \text{conv, bn} \end{bmatrix}$
C13	$\begin{bmatrix} 3 \times 3 \times 3, 64, \text{stride} = 2 \\ 3 \times 3 \times 3, 64 \end{bmatrix}$	$\begin{bmatrix} \text{conv, bn, ReLU} \\ \text{conv, bn, ReLU} \end{bmatrix}$
C14	$[3 \times 3 \times 3, 64, \text{stride} = 2]$	[deconv, bn]
C15	$[3 \times 3 \times 3, 32, \text{stride} = 2]$	[deconv, bn]
C16	$\begin{bmatrix} 3 \times 3 \times 3, 32 \\ 3 \times 3 \times 3, 1 \end{bmatrix}$	$\begin{bmatrix} \text{conv, bn, ReLU} \\ \text{conv} \end{bmatrix}$

Interpolation and regression

All three outputs of the 3D network are upsampled by trilinear interpolation to dimension $W \times H \times P$. Regression is performed by

$$o_{u,v} = \sum_{i=1}^{P} i \cdot \sigma_{\text{SM},i}\left(-\underline{\mathbf{O}}\left[u, v\right]\right) \tag{3.27}$$

on all of them. $\sigma_{\text{SM},i}\left(-\underline{\mathbf{O}}\left[u, v\right]\right)$ is the softmax operation, applied on a negative vector of the output volume at position (u, v) and evaluated at plane index i. This way the mean plane index, weighted by its probability, is chosen as the best matching plane. The estimated plane index, i. e. the label, can have non-integer values. Finally, the label image is converted to an elevation image.

3.3.2 Training

In [23] the neural network described above is used to estimate disparity maps from rectified stereoscopic images. Instead of assembling the cost volume according to the plane-sweep approach, feature maps of one image are shifted and stacked on top of the other image's feature map. That way, the horizontal epipolar geometry is implemented. The smooth L1 loss function is utilized to compare each of the three outputs to the target. The three loss values are weighted by 0.5, 0.7, and 1, 0 and are summed.

In order to train the proposed network (referred to as CNN in the following), pre-trained parameters from the disparity network are used. Then, cost volume assembly is switched to the plane-sweep approach. Training is performed on

an NVidia GTX 1080 Ti GPU with 11 GB of memory. Since there is not enough memory to train the network when using full resolution images (1920×1200 pixels), patches of 256×256 pixels are randomly selected from the right image. The corresponding patch in the left image is found by transforming the patch's corners with the plane homographies of the lower and upper plane hypotheses. The patch from the left image is cut out and is padded to a uniform size of 576×300 pixels. This is done because the patch's size depends on its position on the plane and on the orientation of the plane. The uniform patch size enables the use of training batches. The principal point in the camera matrix has to be adjusted according to the patch location. By using two graphics cards in parallel, a batch size of 8 can be used. In evaluation mode, if no gradients are required and if intermediate results are deleted, the network fits into a single GPU's memory while using full resolution images.

3.3.3 Evaluation

During the network training, all three outputs are compared to the target in a supervised way. During the evaluation, only the final output is considered. The yet unknown precise location of the mean road surface is found in the same way as in Section 3.2.4, i. e. an initial location is guessed, the CNN is evaluated, and the mean surface location is refined. These steps are repeated until convergence.

4 Deep Learning Self-Calibration from Planes

In Section 2.6 three different categories of camera calibration techniques were described: calibration with 3D targets and control points, calibration with planar targets and control points, and self-calibration without control points. In [56] it is stated that the precision of the extracted parameters of methods from these categories decreases in the given order. However, the manufacture and high precision measurement of a 3D calibration target are often unfeasible. A 2D target is easier to create and is the standard calibration method using the popular OpenCV and MATLAB libraries. Nevertheless, it is a tedious task, because standard methods require many images from different perspectives of the target. Obviously, it would be more practical not to have to use a custom made pattern.

Self-calibration is the extraction of calibration parameters from unstructured scenes. It is widely used in SfM methods, where a metric reconstruction is performed from an unordered collection of images from uncalibrated cameras. In [83] this is used on a series of images of a single camera. The result is not only the 3D reconstruction but also the in- and extrinsic camera parameters, as well as parameters of a simple distortion model. Self-calibration can be used for a single camera with fixed intrinsic parameters, but also for a collection of different cameras that picture the same scene. A critical part in self-calibration is the automatic matching of corresponding feature points between images. It is addressed in [40], where an iterative search algorithm for feature matches is shown. It incorporates only a single radial distortion parameter, although the method itself is not limited to one parameter. A more recent work in [22] follows a similar approach and incorporates more distortion parameters.

The focus of self-calibration techniques from 3D scenes is on 3D reconstruction and not on the precise estimation of calibration parameters [22, 39, 83, 92]. Planar scenes, on the other hand, have been used for the purpose of estimating parameters, and, in [52], promising results are shown, but none of the previously published methods are capable of reconstructing all parameters of the widely used camera model that is described in Section 2.3.

In this chapter, a novel self-calibration method is introduced, capable of reconstructing all parameters of the complex camera model from Section 2.3. It

is fundamentally different from previous methods, because it is not based on feature points matched across multiple images, but on feature maps of entire images. In Section 4.1 the shortcomings of the previous self-calibration methods are discussed. The idea of the proposed method is laid out for mono cameras in Section 4.2. It requires some initializations, which are found in Section 4.3. In Section 4.4 the proposed method is extended for use with stereo camera systems. The method has previously been published in [15].

4.1 Shortcomings of previous self-calibration techniques

In Section 2.6 current self-calibration techniques were shown. All of them are based on the simultaneous localization of feature points and the extraction of camera parameters. Therefore they require point matches between multiple views of the scene. This approach causes the following problems:

1. Feature point matches between different images must be exact and actually belong to the same object point. Wrong matches impair the calibration result.

2. With severe lens distortion, matching feature points is difficult because the epipolar geometry constraint cannot help guide the search for matches or discard wrong matches.

3. Common distortion models are based on polynomials. Since polynomials extrapolate poorly, feature points must be found uniformly across the entire image space. Not covering the entire image space has been shown to greatly decrease the performance of self-calibration methods [39]. That is also true if most feature points lie in the center and less are covering the edges because the number of points has a weighting effect on the geometric error. While images are being captured, it is difficult to predict where feature detection algorithms will find feature points.

4. Only image coordinates containing a feature point are used for calibration. Information that exists at other coordinates is discarded.

Due to these difficulties, works on self-calibration from 3D scenes focus on 3D reconstruction and not on the precise estimation of calibration parameters [22, 39, 83, 92]. They use only simple distortion models and do not compare the extracted calibration parameters to other calibration techniques [22, 40, 83]. The problem of finding point correspondences is particularly evident in [40], where correspondences are inserted manually. The difficulty of finding point

correspondences also exists in the methods that perform camera calibration from planar scenes. This can be seen in the results of [52] and [67], where again point correspondences are found manually.

4.2 Method

The new self-calibration method overcomes the issues that arise in the calibration from (possibly wrong) feature point matches. It is also based on self-calibration from planar scenes but takes a fundamentally different path. Instead of relying on feature point matches, feature maps of entire images are extracted. Instead of optimizing the distance between reprojected features points, the overall similarity between transformed views is optimized.

Under the assumption of undistorted images and the projective camera model, every two images of a plane are related by a plane homography, and with its help, an image can be transformed to match the other image. The homography depends on the relative orientation between the cameras, the plane's location in space, and the calibration matrices of the cameras, hence on the in- and extrinsic camera parameters. The idea to find the unknown in- and extrinsic as well as distortion parameters is first to undistort images and then to transform them by plane homographies to match an undistorted reference view. Using a roughly estimated set of parameters, this will not be the case initially, but by altering the parameters until the views match precisely, the parameters are determined. The optimization is based on a training approach for neural networks.

In order to infer the unknown parameters, their search must be guided in the right direction. That requires a measure for image similarity that is differentiable to the unknown parameters. Therefore, feature maps are extracted by the feature map network from Section 3.3.1, which was shown in Figure 3.3. The images are undistorted and transformed by inverse warping. Then, feature maps are extracted. Finally, their similarity is compared with the cosine similarity measure. All these steps are implemented in a neural network and are differentiable to the unknown parameters. Hence, they can be found by backpropagation and gradient descent in a neural network training loop. The use of feature maps instead of feature points solves all issues that were mentioned in the previous section:

1. & 2. The issue of mismatched feature points does not arise.

3. Since the feature maps are extracted from the entire images, the polynomials are also fitted over the entire images.

4. No information from parts of the images is discarded.

The new method will be referred to as deep learning self-calibration (DLSC) in the following.

4.2.1 Number of unknowns

Before continuing with the method, the question should be answered how many parameters can theoretically be found and how many images are required. Every additional perspective to the reference perspective provides a homography that possesses eight independent variables. With respect to the reference camera, three translation and three rotation parameters are required to describe another camera's location. The plane is located in front of the reference camera, and the distance and two rotations are necessary to describe its location. By fixing the distance at one, the overall scale of the scene is fixed. That means

$$8(m-1) - 6(m-1) - 2 = k \tag{4.1}$$

unknown parameters k can be found from m views.

The warped views only match the reference view under the assumption of the pinhole camera model. The distortion model indirectly estimates the principal point because it matches the center of distortion. The distortion model itself can be learned from only two images, as is shown in [97], since every pixel functions as one condition on a system of equations. Since zero skew is assumed, this leaves two parameters in the camera matrix: f_x and f_y. Hence, as long as there is lens distortion, three views of a plane are sufficient to estimate the full camera model. This will be demonstrated in Section 7.5. If the principal point cannot be estimated from the distortion model, four views of the plane are required, and if the skew is not assumed zero, five views are required.

4.2.2 Transformation functions

The space where the images or their feature maps are compared is the undistorted reference camera view. That is in contrast to the methods described in Section 2.6.4, where the distance between reprojected points is measured in the distorted reference camera space. The reason for this is that to apply the undistortion function to an image, inverse warping is used. For inverse warping, the inverse of the transformation function is needed, which is the distortion function. However, the distortion function cannot be analytically inverted.

The coordinate system is set such that the XY-plane coincides with the calibration plane, just as was done in Section 3.1.1. With the abbreviations from

Section 2.3, a point from the plane is projected into the i'th camera by

$$\underline{\tilde{m}}^i = \kappa \left(\Theta \left(\delta \left(\begin{bmatrix} \underline{\mathbf{r}}_i^1 & \underline{\mathbf{r}}_i^2 & \underline{T}_i \end{bmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \right) \right) \right) . \tag{4.2}$$

To transform a point from the i'th view into the undistorted reference view, it is first projected onto the plane by inversing Equation 4.2

$$\mathbf{x} = \begin{bmatrix} \underline{\mathbf{r}}_i^1 & \underline{\mathbf{r}}_i^2 & \underline{T}_i \end{bmatrix}^{-1} \delta^{-1} \left(\Theta^{-1} \left(\kappa^{-1} \left(\underline{\tilde{m}}^i \right) \right) \right) , \tag{4.3}$$

and then transformed into the undistorted reference view by applying Equation 4.2, where $\Theta()$ is the identity function

$$\underline{m}^{ref} = \kappa \left(\delta \left(\begin{bmatrix} \underline{\mathbf{r}}_{ref}^1 & \underline{\mathbf{r}}_{ref}^2 & \underline{T}_{ref} \end{bmatrix} \mathbf{x} \right) \right) . \tag{4.4}$$

Equations 4.3 and 4.4 are combined and inversed since for applying inverse warping on images, the inverse transformation function from distorted coordinates in the i'th image to undistorted coordinates in the reference image is needed

$$\underline{\tilde{m}}^i = \kappa \left(\Theta \left(\delta \left(\begin{bmatrix} \underline{\mathbf{r}}_i^1 & \underline{\mathbf{r}}_i^2 & \underline{T}_i \end{bmatrix} \begin{bmatrix} \underline{\mathbf{r}}_{ref}^1 & \underline{\mathbf{r}}_{ref}^2 & \underline{T}_{ref} \end{bmatrix}^{-1} \delta^{-1} \left(\kappa^{-1} \left(\underline{m}^{ref} \right) \right) \right) \right) . \tag{4.5}$$

The reference image only needs to be undistorted. This simplifies Equation 4.5 to

$$\underline{\tilde{m}}^{ref} = \kappa \left(\Theta \left(\kappa^{-1} \left(\underline{m}^{ref} \right) \right) \right) . \tag{4.6}$$

Inverse warping is performed on the reference image with Equation 4.6, and on all other images with Equation 4.5.

4.2.3 Comparing the images

In the DLSC approach, the sought parameters are estimated by varying them until the transformed images from all views match each other. In order to guide the search for the correct parameters, a measure is required that describes the similarity of those transformed images. The easiest method to compare images is a sum of pixel-wise differences, but this would not provide any information on how to change parameters to improve the similarity. Therefore, the feature map network from Section 3.3.1 in Figure 3.3 is used. It calculates features at a region level, and it is assumed that the similarity of features in an image will slowly decrease with distance, thus guiding the search for good

parameters. The feature map network outputs feature maps $\underline{\mathbf{Z}}$ of dimension $W/4 \times H/4 \times Z$, i. e. a feature vector for every location.

For the comparison of feature vectors, the cosine similarity is a suitable measure [72]. In order to calculate a single value describing the overall similarity, the mean cosine similarity measure is used

$$sc = \frac{1}{N} \sum_{i=1}^{N} \sum_{u,v} \frac{\underline{\mathbf{Z}}^i[u,v] \cdot \underline{\mathbf{Z}}^{ref}[u,v]}{|\underline{\mathbf{Z}}^i[u,v]||\underline{\mathbf{Z}}^{ref}[u,v]|}, \tag{4.7}$$

where $\underline{\mathbf{Z}}^i[u,v]$ is the feature vector at the coordinates (u,v) of the feature map that belongs to the i'th image in the undistorted reference view, and N is the number of additional views to the reference view. $\underline{\mathbf{Z}}^{ref}$ is the feature map of the undistorted reference image.

4.2.4 Optimization

The images are undistorted and transformed to the reference view by inverse warping. In order to find values at non-integer pixel locations, bilinear interpolation is used.[1] Then, feature maps are extracted from the images. Finally, the mean cosine similarity cost function is applied to the feature maps. It is greatest if all images have been perfectly transformed to the undistorted reference view, i. e. if all parameters have been estimated correctly. By optimizing the similarity, the parameters are found.

All calculations are implemented in a neural network framework and are differentiable to the parameters. Using a rough estimate of the parameters, they can thus be learned in a training loop through backpropagation and gradient descent, while the weights of the feature extraction network remain unchanged.

Encoding of orientation parameters

The goal of the optimization is to find the intrinsic camera and the distortion parameters. The extrinsic parameters are not of special interest, but also need to be estimated for the method to work. In the transformation functions 4.5 and 4.6, the rotation matrices are used to describe the orientation of the cameras. A rotation matrix possesses nine entries, but only three parameters are required to describe a 3D rotation. For the optimization, an encoding is needed in order to estimate only as many parameters as necessary. One possibility of encoding a rotation matrix are Euler angles. These have the advantage of consisting of exactly three values but suffer under discontinuities, making them unsuitable

[1] In [60] it is shown that bilinear interpolation is differentiable and how it can be used in a neural network

for optimization [66]. Instead, pairs of axis and angles are used to describe the rotations of the cameras. The axes are described by unit length vectors. Since the condition of unit length is not enforced by the gradient descent optimization, they are renormalized after every iteration. Experiments showed that the optimization converges faster when using three parameters for the rotation axis than when using only two and deriving the third.

In order to encode the orientation of the reference camera, two Euler angles are used. They are physically restricted to a suitable range, and the plane can be rotated around its normal axis without changing the image transformations. Axis-angle representations and Euler angles are then converted to rotation matrices. The translation vector of the reference camera is given by $\underline{T}_{ref} = -\underline{\mathbf{R}}_{ref} \cdot (0, 0, 1)^T$.

Training

In order to decrease the training time, all images are downscaled by a factor of four in both dimensions. As soon as the parameters converge, downscaled images with a factor of two are used. Finally, images at full resolution are processed. If the scaling of the images is changed, the camera matrix's scaling also has to be adjusted. The distortion parameters are independent of the camera matrix.

For finding the best parameters, the Adam optimizer [65] is used in all experiments.

4.3 Initialization

To start the gradient descent training loop, the initialization of the parameters is necessary. For the cosine similarity function to calculate a meaningful output, the transformed images have to overlap at least in parts. That can also be checked visually by transforming the images to the reference view. If no prior knowledge about camera parameters is available, they can be roughly estimated. That is done in two steps. First, the radial distortion is estimated and second, the focal length.

4.3.1 Distortion

Views of a plane are only related to each other by plane homographies under the projective camera model assumption without lens distortion. Thus, the distortion function can be found by varying the distortion parameters until the homography assumption holds. That is utilized in the following way:

- Points from all views are extracted with the SIFT feature extractor [69]. They are matched pairwise between the reference view and all other views by a brute-force method and checked for validity by a ratio test.
- The distortion function is inversed iteratively [11], and the undistorted feature point coordinates are found. For undistorting the images, a camera matrix with a principal point located at the center is used. It is assumed that f_x and f_y are similar, and they are set to the same arbitrary value.
- The homography that is compatible with the most points is found by RANSAC. The number of inliers of the RANSAC method indicates how well the point coordinates are undistorted.
- The distortion parameters are varied by a nonlinear optimization method in order to increase the number of inliers. In the experiments, only K_1 and K_2 are varied, and the optimization is carried out by simulated annealing with the Nelder-Mead method as a local method.

This approach is similar to [67] and [97]. It has proven to be robust, and no manual point matching is necessary.

4.3.2 Calibration matrix

With the estimated best fitting homographies from undistorted feature point coordinates from the previous section, and with $\underline{\omega} = \left(\underline{\mathbf{K}\mathbf{K}}^T\right)^{-1}$, Equation 2.105 is set up

$$\left(\underline{\mathbf{H}}_{0\to j}\mathbf{c}_{1,2}{}^0\right)^T \left(\underline{\mathbf{K}\mathbf{K}}^T\right)^{-1} \left(\underline{\mathbf{H}}_{0\to j}\mathbf{c}_{1,2}{}^0\right) = 0 \tag{4.8}$$

and is numerically solved for the unknowns. For initialization, the principal point is assumed to be at the center of the image sensor. Therefore, only f_x and f_y have to be found. For regular photo cameras, f_x can be set equal to f_y, although this is not required. Furthermore, the circular points $\mathbf{c}_{1,2}{}^0$ with two unknown parameters each need to be found. Three images are sufficient to find the unknowns (see Section 4.2.1), i. e. three images give six equations to find six unknowns.

Since the image distortion is calculated in normalized coordinates, the distortion depends on the camera matrix. Therefore, K_1 and K_2 have to be adjusted after finding f_x and f_y. Putting Equations 2.28, 2.29 and 2.26, 2.27 into 2.24 and 2.25, and by only using the distortion coefficients K_1 and K_2, one yiels

$$\tilde{u} - u_0 = (u - u_0)\left(1 + K_1|\underline{\bar{m}}|^2 + K_2|\underline{\bar{m}}|^4\right) \tag{4.9}$$

$$\tilde{v} - v_0 = (v - v_0)\left(1 + K_1|\underline{\bar{m}}|^2 + K_2|\underline{\bar{m}}|^4\right). \tag{4.10}$$

The term $\left(1 + K_1|\underline{\bar{m}}|^2 + K_2|\underline{\bar{m}}|^4\right)$ needs to stay constant if f_x and f_y are replaced by $f_{x,new}$ and $f_{y,new}$. Hence, with $|\underline{\bar{m}}|^2 = \left(\frac{u-u_0}{f_x}\right)^2 + \left(\frac{v-v_0}{f_y}\right)^2$,

$$K_1\left(\left(\frac{u-u_0}{f_x}\right)^2 + \left(\frac{v-v_0}{f_y}\right)^2\right) = K_{1,new}\left(\left(\frac{u-u_0}{f_{x,new}}\right)^2 + \left(\frac{v-v_0}{f_{y,new}}\right)^2\right) \tag{4.11}$$

and

$$K_2\left(\left(\frac{u-u_0}{f_x}\right)^2 + \left(\frac{v-v_0}{f_y}\right)^2\right)^2 = K_{2,new}\left(\left(\frac{u-u_0}{f_{x,new}}\right)^2 + \left(\frac{v-v_0}{f_{y,new}}\right)^2\right)^2. \tag{4.12}$$

With $f_x = f_y = f$ and $f_{x,new} = f_{y,new} = f_{new}$ this simplifies to

$$K_{1,new} = K_1\frac{f_{new}^2}{f^2} \tag{4.13}$$

$$K_{2,new} = K_2\frac{f_{new}^4}{f^4}. \tag{4.14}$$

4.3.3 Extrinsic parameters

With a known camera matrix, the best fitting homography for every view from Section 4.3.1 is decomposed into a rotation matrix and a translation vector that relate the reference camera to the other camera, and into a plane normal that describes the orientation of the plane [71]. The decomposition is ambiguous, but the correct solution can be chosen by comparing the plane normal to the physically possible solution.

Since in- and extrinsic camera parameters for every image pair are known, the feature points that were inliers in the RANSAC method can now be triangulated. The coordinate system is placed at the reference camera center, and a plane is found in the resulting point cloud (see Section 3.2.4). Its Z-axis intercept is found, which is the distance between the reference camera and plane. In order to fix this distance at 1, the translation vectors of the homography decompositions are scaled accordingly. That is done for all cameras to achieve a projection model that is consistent with all cameras.

4.4 Calibration of a stereo camera

The proposed DLSC method is easily extended for use with a stereo camera. The procedure remains the same, and the images from all cameras are transformed to the reference view, where they are compared with the feature maps' help. The difference to the calibration of a mono camera is that every two cameras of the stereo setup are related by a fixed rotation matrix $\underline{\mathbf{R}}_{st}$ and a translation vector $\underline{T}_{st}$. Therefore, the degrees of freedom are reduced.

4.4.1 Transformation functions

In order to perform inverse warping, the transformation functions are needed. Points from the plane are projected into the left and right camera heads of the i'th stereo camera by

$$\underline{\tilde{m}}^{s,i} = \kappa_s \left(\Theta_s \left(\delta \left(\begin{bmatrix} \underline{\mathbf{r}}^1_{s,i} & \underline{\mathbf{r}}^2_{s,i} & \underline{T}_{s,i} \end{bmatrix} \mathbf{x} \right) \right) \right) \text{ for } s = l, r\,, \tag{4.15}$$

with $\underline{\mathbf{R}}_r = \underline{\mathbf{R}}_{st}\underline{\mathbf{R}}_l$, $\underline{T}_r = \underline{\mathbf{R}}_{st}\underline{T}_l + \underline{T}_{st}$. By choosing one of the left-hand stereo heads as the reference camera, all feature maps are transformed to the undistorted left reference camera space by inverse warping. Equation 4.5 is extended to

$$\underline{\tilde{m}}^{s,i} = \kappa_s \left(\Theta_s \left(\delta \left(\begin{bmatrix} \underline{\mathbf{r}}^1_{s,i} & \underline{\mathbf{r}}^2_{s,i} & \underline{T}_{s,i} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{r}}^1_{L,ref} & \underline{\mathbf{r}}^2_{L,ref} & \underline{T}_{L,ref} \end{bmatrix}^{-1} \delta^{-1} \left(\kappa_s^{-1} \left(\underline{m}^{L,ref} \right) \right) \right) \right) \right) . \tag{4.16}$$

4.4.2 Number of unknowns

Once again, every additional view to the reference view provides a homography that possesses eight independent variables. In order to find the location and orientation of the reference stereo camera in relation to the plane, three parameters are necessary. In order to find every additional stereo camera, six more parameters are needed. Therefore,

$$8(2m_{st} - 1) - 6(m_{st} - 1) - 3 = k_{st} \tag{4.17}$$

parameters can be extracted from m_{st} stereo camera views of a plane. The stereo camera itself contains eight intrinsic camera parameters (assuming zero skew) and five parameters for the stereo camera geometry. The five parameters consist of three rotation parameters and two for translation. There are only two translation parameters since the overall scale cannot be extracted from images

alone due to the projective geometry. Hence, two stereoscopic images of the plane in a general orientation are sufficient to extract all parameters.

The proposed method's target application in this thesis is the self-calibration of cameras behind the windshield of a vehicle. In this case, one is faced with the problem that only one plane can be captured without lifting the vehicle, namely the ground level. Following Equation 4.17, this leaves 5 parameters that can be extracted, but these are needed for the camera geometry. To still be able to extract the intrinsic camera parameters, physical measurements are included in the process: The baseline and the distance between stereo camera rig and the ground plane are measured. By fixing the baseline, the overall scale is fixed, and the distance reduces one degree of freedom from the equation. By assuming zero skew and rectangular pixels, only one intrinsic parameter per stereo camera head is left since the principal point is part of the distortion model, and its coordinates are found independently from the projective camera model. By further assuming the same focal length for both camera heads, only one intrinsic camera parameter is left and can be extracted.

There also exist use cases where the intrinsic parameters have been estimated beforehand, and only the extrinsic parameters are sought. One example is cameras whose intrinsic parameters have been estimated in a laboratory that are used as a stereo pair. In [111] structured light is used for this purpose. In contrast, the DLSC method can calibrate the extrinsic parameters from a single stereoscopic image without the need for external light sources.

5 Stereo Camera System Design

The target of the stereo camera system is to capture the 3D profile of the road surface that lies in front of a vehicle. Two cameras shall be used. The following requirements can be derived from this and have to be considered:

- In order to make a 3D reconstruction possible, the surface has to be covered by both cameras.
- The spatial resolution has to be large enough to capture the texture of the road. The size of the road segment that shall be covered in a single stereoscopic image then determines the minimum resolution of the image sensor. The resolution and physical pixel dimensions determine the focal length of the lens.
- The image sensor resolution is proportional to the achievable depth resolution for a given road segment size.
- Processing high-resolution images is computationally demanding and limits the resolution.
- Since the depth resolution of a stereoscopic reconstruction increases with increasing baseline, the baseline should be as large as possible.
- The aperture influences the depth of field and the exposure time. A large depth of field requires a small aperture, but a long exposure time. A long exposure time increases the motion blur.
- Since the vehicle is moving, the cameras must be triggered synchronously.
- The cameras must be calibrated.

Image sensors are available with two different capturing methods: global shutter and rolling shutter. Global shutter image sensors read every pixel at once, while rolling shutter image sensors read pixels sequentially [3]. The sequential acquisition causes distortion effects if the target is moving. Therefore, a global shutter image sensor has to be used. Due to the computing power required for processing high-resolution images, an image sensor with a resolution of $1920 \times 1200\,\mathrm{px}$ was chosen for this research. It has a pixel size of $4.8 \times 4.8\,\mu\mathrm{m}^2$,

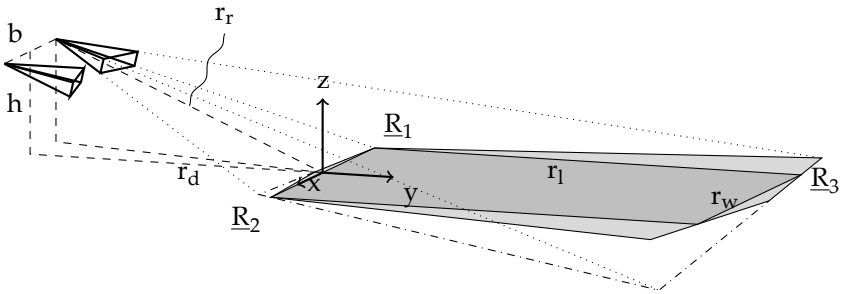


Figure 5.1: Scene layout – The two cameras capture the road surface. The left camera captures the area that is marked by a dash-dotted line. The light gray area is captured by the left and the right camera. The darker area is the road segment of interest.

and it will be used for all experiments throughout this work. The achievable spatial resolution will be derived from the image sensor.

The camera height h above the road and the largest possible baseline b are determined by the vehicle. In Section 5.1 all other geometric parameters are derived from the image sensor, from the lens' focal length and from the width and length of the road segment to be captured. In Section 5.2 the effect of the focal length on the theoretical spatial resolution and on the depth resolution is investigated. Due to the blur from motion and blur from defocus, the effective resolution is reduced. This is discussed in Section 5.3. Finally, in the Section 5.4, a lens is selected and the aperture and focusing length are also specified.

5.1 Geometrical orientation

The geometrical orientation of the cameras is determined by the road segment width r_w and length r_l to be covered in a single stereoscopic image, by the baseline b, by the height h of the cameras, by the focal length of the lens f_L, and by the resolution and size of the image sensor (Figures 5.1, 5.2, 5.3). r_w and r_l are specified by the user, b and h by the vehicle. The focal length is a still free to choose parameter. An optimal focal length is discussed in Section 5.4, but for its determination the geometrical camera orientation must first be known. The parameters that describe the orientation will be derived in this section.

In Chapter 6 the cameras are mounted on an aluminum bar. On it, the cameras can be rotated around their Y^0-axis. The bar, in turn, can be rotated around the X^0-axis (see Figure 3.1). Thus, their orientation is restricted and can be described by extrinsic Tait-Bryan angles around the coordinate system axis in the order Z-Y-X, with rotation φ around X-axis, θ around Y-axis and

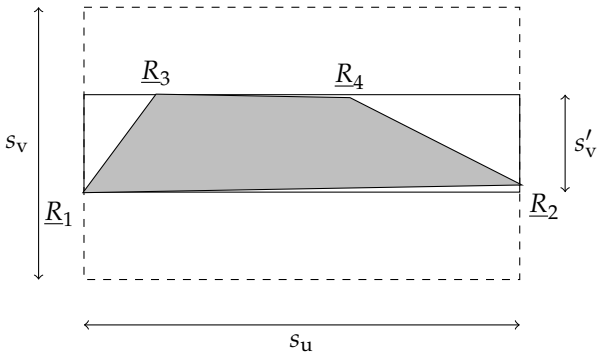


Figure 5.2: The dashed rectangle shows the left camera view of the road surface. The area highlighted in gray is the road segment of interest.

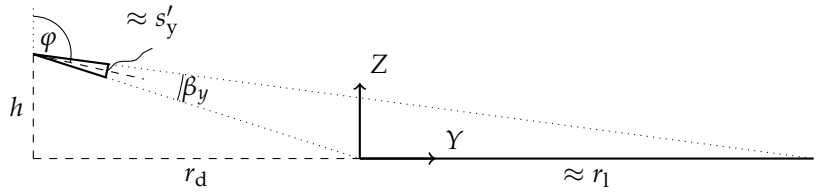


Figure 5.3: Side view of the scene layout.

ψ around the Z-axis. Figure 5.2 shows a typical view of a road segment from the left camera's perspective. It can be seen that the usable vertical sensor resolution s'_v is generally limited due to the aspect ratio of common image sensors. Therefore, the frame is selected so that it is horizontally limited by the lower left and by the lower right corner of the road segment. Vertically the road segment is placed in the center. From this the target is to find s'_v, and φ, θ in the road coordinate system that is shown in Figure 5.1.

Width the edge length of a pixel m_x and the resolution of the image sensor in horizontal direction s_u, the distances to the front end of the segment are approximated by

$$r_\mathrm{r} \approx \frac{f_\mathrm{L} r_\mathrm{w}}{s_\mathrm{u} m_x} \tag{5.1}$$

and

$$r_\mathrm{d} = \sqrt{r^2 - h^2 - \left(\frac{b}{2}\right)^2}, \tag{5.2}$$

which can be seen from Figure 5.1.

$$\theta \approx \arcsin\left(\frac{b}{2r_\mathrm{r}}\right) \tag{5.3}$$

describes the inclination of the cameras towards each other. From Figure 5.3

$$\beta_y = \arctan\left(\frac{r_\mathrm{d}+r_\mathrm{l}}{h}\right) - \arctan\left(\frac{r_\mathrm{d}}{h}\right) \tag{5.4}$$

is found. It is used in Equation 2.3, which gives

$$s'_\mathrm{v} = \frac{2\tan\left(\frac{\beta_y}{2}\right) f_\mathrm{L}}{m_y}\,. \tag{5.5}$$

From Figure 5.3

$$\varphi = \pi - \arctan\left(\frac{r_\mathrm{d}}{h}\right) - \frac{\beta_y}{2} \tag{5.6}$$

is found.

Figures 5.1 and 5.2 show the corners $\underline{R}_1$,$\underline{R}_2$,$\underline{R}_3$ of the road segment. Their image coordinates are given by

$$\underline{r}_i = \delta\left(\underline{\mathbf{K}}\,[\underline{\mathbf{R}}|\underline{T}]\,\delta^{-1}\left(\underline{R}_i\right)\right)\,, \tag{5.7}$$

with $\underline{T} = -\underline{\mathbf{R}}\underline{C}_l$, where $\underline{\mathbf{R}}$ is found from the consecutive rotations θ and φ. $\underline{\mathbf{K}}$ is assembled according to Equation 2.14, where the principal point is placed in the center. As can be seen in Figure 5.2, the corners of the road segment appear in the lower-left corner, the right-hand edge, and the upper edge of the left camera image. With these four conditions and with the previously found values as initialization, a nonlinear system of equations is numerically solved for the unknowns φ, θ, s'_v and r_d.

5.2 Estimating the theoretical resolution

For a given road segment size and a given image sensor, the theoretical spatial and depth resolutions depend on the lens' focal length. The effect is investigated in this section. For simplicity, the theoretical resolution is calculated for the central line between cameras, i. e. along the Y-axis (Figure 5.1). It is evenly discretized, and three samples are created per level: one on the Y-axis, one with a slightly negative X- and one with a slightly positive X-coordinate. The

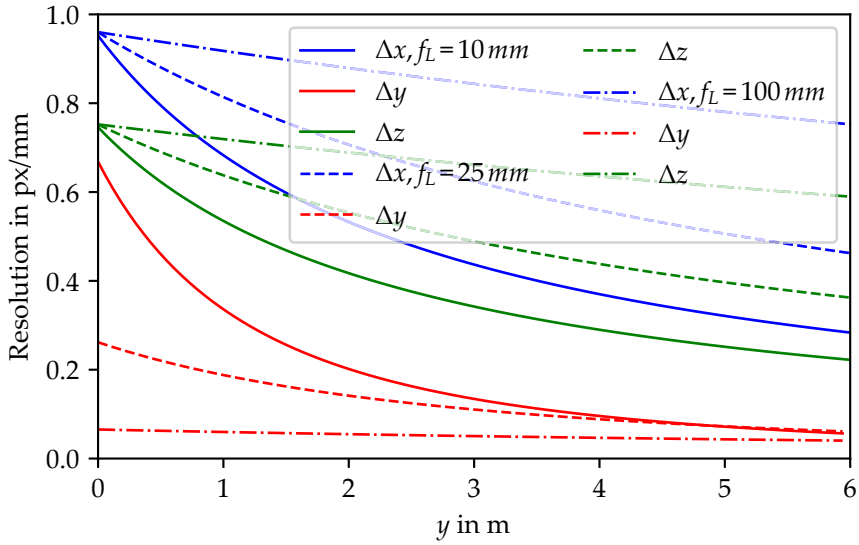


Figure 5.4: Spatial resolution in X- and Y-direction and the elevation resolution in Z-direction over the position on the road segment for different focal lengths are shown. The calculations for this figure are based on a lane width $r_w = 2\,m$ and length $r_l = 6\,m$.

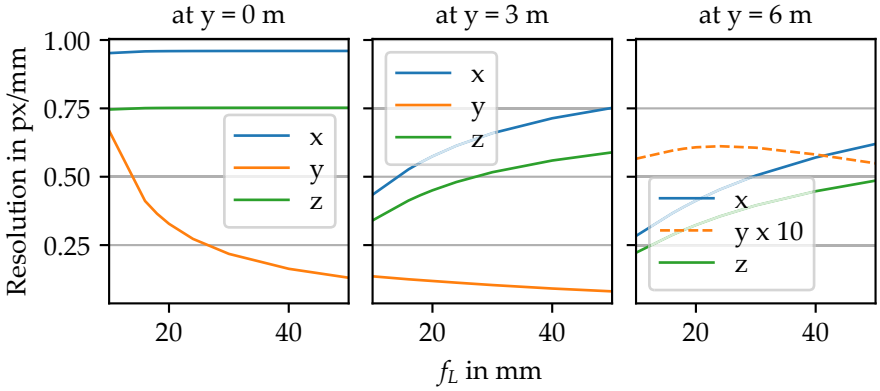


Figure 5.5: The spatial resolution in X- and Y-direction and the elevation resolution in Z-direction depend on the focal length and on the distance between an object and the stereo camera system. The calculations for this figure are based on a lane width $r_w = 2\,m$ and length $r_l = 6\,m$.

points are transformed by Equation 5.7 into pixel coordinates of one of the two cameras. In order to apply the equation, the geometric orientation as a function of the focal length from the last section must first be solved. For the resolution in Y-direction, the spacings of consecutive central point image coordinates are divided by the spacing of discretization levels. For the resolution in X-direction, the distance between points with negative and positive X-coordinate is devided by the distance of their image coordinates.

To calculated the resolution in Z-direction, the central points are transformed to the left camera space by the plane homography (Equation 3.4) that belongs to the XY-plane. Then they are transformed into the right camera space by plane homographies twice – through a plane with a positive and through a plane with a negative Z-coordinate. The distances of the image coordinates in the right camera space are divided by the spacing of the two planes. That resembles the plane sweep approach, which is used for the elevation estimation. The result is shown in Figure 5.4 and 5.5 for a road segment that is $r_\mathrm{w} = 2\,\mathrm{m}$ wide and $r_\mathrm{l} = 6\,\mathrm{m}$ long.

As the focal length increases, the distance between the cameras and the road segment r_d increases because the segment is vertically fitted on the image. Therefore, the inclination of the cameras downwards to the road φ decreases. As the front edge of the lane is fitted to match the image width, the resolution in X-direction stays approximately constant at the front edge and decreases towards the rear edge, since the rear edge covers a smaller area in the image, see Figure 5.2. Due to the perspective, it decreases slower if φ is decreased (and f_L is increased). The resolution in Y-direction generally decreases if the angle between camera and surface decreases, because the usable sensor size s'_v decreases. At the same time, the tilt has the effect of increasing the resolution in Y-direction at the front edge and decreasing the resolution at the rear edge. Thus, at a certain distance Y, there is a focal length that optimizes the resolution in Y-direction, which can be seen in Figure 5.5 at $Y = 6000\,\mathrm{mm}$. The resolution in Z-direction increases with an increasing focal length because φ decreases. It is further approximately proportional to the resolution in X-direction because the search direction of the plane sweep approach is mainly in X-direction. Overall, the lowest resolution always is in Y-direction.

5.3 Estimating the effective resolution

Overall, the theoretical resolutions in X- and Z-direction increase with increasing focal length, while the resolution in Y-direction decreases. However, the theoretical resolution in Z-direction is indirectly limited by the spatial resolution in X- and Y-direction because the surface texture must be captured to make

a stereoscopic reconstruction possible. During the estimation of the elevation, a path in one image is searched for the corresponding pixel in the other image. The length of this path is determined by the considered volume in 3D space. For a 3D reconstruction to be successful for an isolated pixel, the pixels in the search path must be unique. Since the implemented stereo algorithms search for a globally optimal solution, this requirement can be relaxed, but it is clear that ambiguity should be small. The blur from motion and defocus blend the intensities of neighboring pixels, and due to the finite numerical resolution, the pixels become indistinguishable.

The motion blur depends on the exposure time, which in turn depends on the aperture and the focal length. A short exposure time results in little motion blur but requires a large aperture. A large aperture, in turn, leads to high blur due to defocusing. The relationship between exposure time, aperture, and focal length is discussed in Section 5.3.1. The total amount of blur is calculated in Section 5.3.2.

Another factor influencing the total amount of blur is diffraction. The Airy disk, which is the central and most visible part of the diffraction pattern that a circular aperture produces, has radius $\approx 1.22\lambda f_k$ [50], where λ is the wavelength of light. Therefore, diffraction does not affect the image sharpness until $f_k > 5.6$ with the imaging sensor used. Simulations later in this section show that larger apertures are needed, and for that reason, diffraction is neglected.

Blur affects the probability of correctly matching isolated pixels in the stereoscopic reconstruction, which is discussed in Section 5.3.3. In Section 5.4, all these effects are considered in order to select the optimal lens and aperture or exposure time.

5.3.1 Calculation of the exposure time

The image sensor registers irradiation

$$B_e = \frac{\Phi_e t_E}{A_s}, \tag{5.8}$$

with the radiant flux Φ_e, the sensor size A_s, and the exposure time t_E. The output signal of an image sensor is nearly proportional to its irradiation [4, 61].The radiant flux, emitted by the road surface and caught by the lens aperture, is calculated by the basic law of radiometry and photometry [46]

$$d\Phi_e = \frac{L_R \, dA_R \cos\beta_R \, dA_L \cos\beta_L}{{d_L}^2}. \tag{5.9}$$

dA_R is a light emitting road element and dA_L is an element of the lens aperture. β_R and β_L are the light ray angles measured against the plane normals of both surface elements. L_R is the radiance of the road, and d_L is the distance between the elements. As only one exposure time can be used for one image, d_L and β_R are calculated for the road segment's central point and are considered to be constant across the segment. The road surface is idealized as a Lambertian surface, i. e. the radiance L_R is independent from the angle β_R [46]. $\beta_L = 0$ because the camera is pointed at that central point and therefore the light is coming in on the optical axis. $\beta_R = \pi - \varphi$ is found from Figure 5.3. With these simplifications, the radiant flux entering the lens is calculated

$$\Phi_e = \frac{L_R A_R \cos\beta_R A_L}{{d_L}^2}\,. \tag{5.10}$$

The lens passes it onto the image sensor. For constant irradiation, and A_s approximately proportional to s'_{y}, a new exposure time can be calculated from parameters that are known to produce a well-illuminated picture. Using Equation 5.10 in Equation 5.8 gives

$$t_{E,new} = t_{E,old}\frac{A_{L,old}\cos\beta_{R,old} d_{L,new}^2 s'_{\mathrm{y},new}}{A_{L,new}\cos\beta_{R,new} d_{L,old}^2 s'_{\mathrm{y},old}}\,. \tag{5.11}$$

The size of the lens aperture is commonly defined by the f-number

$$f_k = \frac{f_{\mathrm{L}}}{D_L}\,, \tag{5.12}$$

where D_L is the diameter of the aperture. With this definition the new exposure time is

$$t_{E,new} = t_{E,old}\frac{f_{\mathrm{L},old}^2 f_{k,new}^2 \cos\beta_{R,old} d_{L,new}^2 s'_{\mathrm{y},new}}{f_{\mathrm{L},new}^2 f_{k,old}^2 \cos\beta_{R,new} d_{L,old}^2 s'_{\mathrm{y},old}}\,. \tag{5.13}$$

5.3.2 Blur from motion and defocus

The defocusing effect is illustrated in Figure 5.6. The lens is focused, such that an object at object distance (or focusing distance) s_{o} creates a sharp image on the sensor at image distance s_{i}. s_{o} and s_{i} are related by Equation 2.1. In Figure 5.6 the edges of the sensor's pixels are shown and are also drawn where their image would appear at object distance. Since the road is slanted to the lens, only the intersect between the road and the sensor's image appears perfectly sharp.

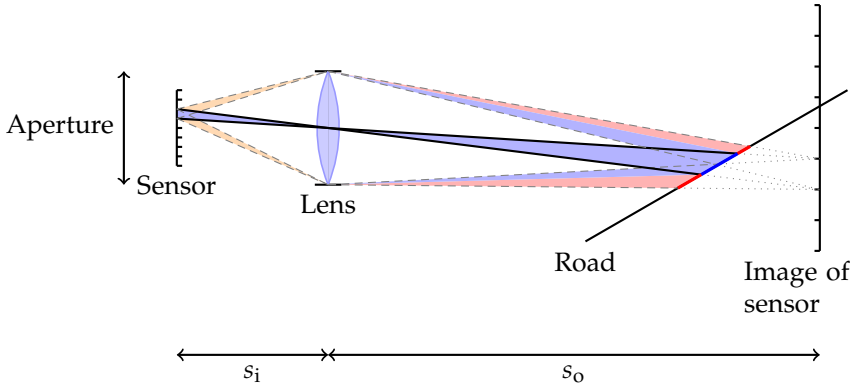


Figure 5.6: Because the road is not parallel to the image sensor, the entire road cannot be completely in focus. The blue segment would ideally be seen by a single pixel, but since the aperture cannot be infinitely small, the red segments also emit light that is captured by that pixel. As a result, the parts of the segment that are not at focusing distance appear blurred.

For illustration, three rays are drawn for both edges of a pixel. They pass through the center of the lens, the uppermost and the lowermost possible point, these points being determined by the aperture. Looking at the cones of light formed by the rays from the two edges passing through the same point on the lens, one can see that the pixel is hit by light from different parts of the road, depending on which point of the lens was passed. Ideally, with an infinitely small aperture, only light emitted by the part drawn in blue would hit the pixel. The parts that are actually captured by the same pixel are those marked in blue and red.

The effect is also shown in Figure 5.7. It shows the light distribution coming from different parts of the road to a single pixel, and for comparison, the part that would ideally be seen by the same pixel. It was calculated by constructing light cones, as shown in Figure 5.6, with the cones' tips uniformly distributed across the aperture. Then, a normalized histogram was created by simultaneously sampling across the road. The overall result of defocus is a blurred image, as the light from each element of the road is scattered across several pixels.

In order to calculate the amount of blur from defocus across the entire image, a single camera is considered, and the rotation θ is neglected. Calculations are performed separately for the X- and Y-direction. Figures 5.6 and 5.7 illustrated the relations for the Y-direction, and this is done equivalently for the X-direction. Since the calculation of the light distribution, as shown in Figure 5.7, is compu-

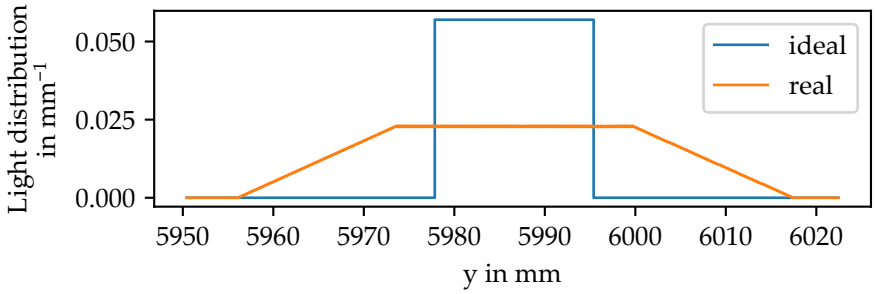


Figure 5.7: Example of the ideal and real light distribution, which is captured by one particular pixel due to defocusing. For the creation of this figure, the following parameters were used: $r_w = 2\,\text{m}$, $r_l = 6\,\text{m}$, $f_L = 25\,\text{mm}$, $f_k = F/3.4$, $s_o = 6.5\,\text{m}$.

tationally expensive, and the calculation is part of an optimization procedure in Section 5.3.3, the maximum width of the trapezoid in both directions b_x and b_y is considered as the size of the road that is captured by a pixel due to defocus. The blur from motion is accounted for by adding the distance that is traveled during the exposure time to b_y. The blur is defined by

$$\text{blur} := \frac{b_x}{b_{x,ideal}} \cdot \frac{b_y}{b_{y,ideal}}, \tag{5.14}$$

where $b_{x,ideal}$ and $b_{y,ideal}$ are the dimensions of the patch that would ideally be captured by a pixel.

The blur depends on the exposure time, the aperture, the focal length, the focusing distance, the vehicle speed, and the road segment's position. By adjusting the focusing distance, either the front end or the segment's rear end becomes blurrier. The point with the minimal blur, therefore, always is in-between. An example is shown in Figure 5.8 for different focusing distances and different focal lengths. In the creation of this figure, the lighting condition and the vehicle speed were fixed, and the focusing distance, the aperture, and the exposure time were optimized, as will be shown in Section 5.3.3.

5.3.3 The impact of blur on the reconstruction quality

In a stereoscopic reconstruction, a path of pixels in one image is searched for a corresponding pixel of another image. If isolated pixels are considered, the reconstruction can only be successful if all pixels in the path are distinguishable.

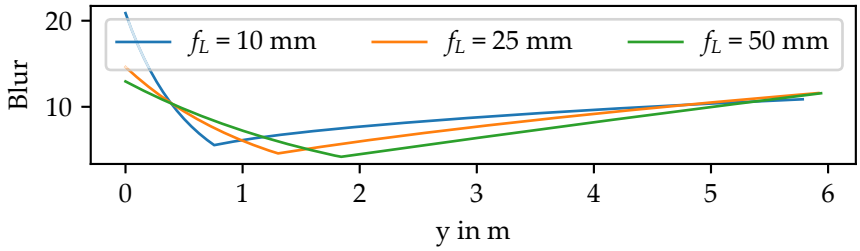


Figure 5.8: The combined blur from motion and from defocus is shown over distance for different focal lengths. Exposure time, aperture, and focusing distance are determined as shown in Section 5.3.3. The lighting condition is given by the following settings: $t_E = 350\,\mu s$, $f_k = F/1.4$, $f_L = 25\,mm$. Furthermore, $r_w = 2\,m$, $r_l = 6\,m$, $v = 50\,km\,h^{-1}$.

Generally, that is not the case. Due to the blur from motion and defocus, pixel intensities of neighboring pixels are blended, and due to the finite numerical resolution, the pixels become indistinguishable. In the plane-sweep approach, as it is implemented in this work, for all pixels in an image, the same elevation range in 3D space is searched for pixel correspondences. Near the stereo camera system, the 3D search space corresponds to a longer search path through image coordinates than at a distance from it. The amount of blur, therefore, is not equally significant across the image. Therefore, it is not the blur that is of interest, but the probability that the pixels are correctly matched.

Probability of correctly matching isolated pixels

In information theory the entropy of a discrete random variable X is estimated by [70]

$$H(X) = -\sum_i P_X(x_i) \log_2 P_X(x_i)\,. \tag{5.15}$$

It measures the average information content of an event. Shannon's source coding theorem states that N independent and identically distributed random variables, each having an entropy of $H(X) = H$ can be represented by NH bits (assuming a negligible risk of information loss and $N \to \infty$) [70]. Regarding the pixel values of a search path as random variables, that means, on average, every pixel within a search path can be characterized by H bits. That effectively makes it possible to distinguish between $n_Q = 2^H$ quantization levels.

All pixels in a search path of length n_{sp} are distinguishable, if there are n_{sp} quantization levels. If there are fewer quantization levels than pixels, the corresponding pixel cannot be matched uniquely. The probability of correctly

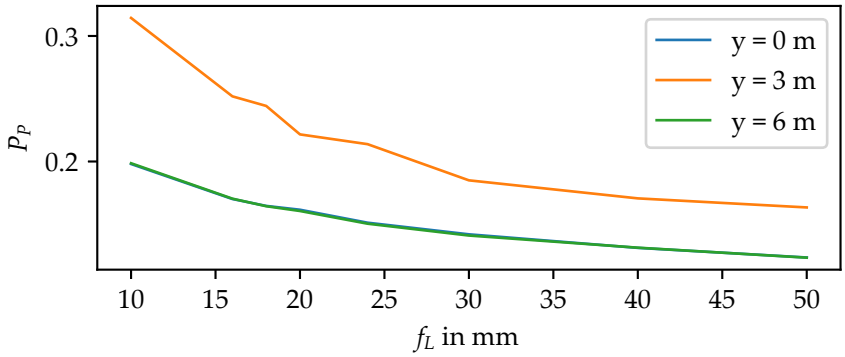


Figure 5.9: The expected probability of correctly matching isolated pixels is shown for different focal lengths at three positions on the lane. Due to the optimization, the probability at $Y = 0\,\mathrm{m}$ and $Y = 6\,\mathrm{m}$ is the same, and the graphs are on top of each other. Calculations are based on the reference lighting condition given by $t_E = 350\,\mu\mathrm{s}$, $f_k = F/1.4$, and $f_\mathrm{L} = 25\,\mathrm{mm}$.

matching isolated pixels therefore is

$$P_P = \frac{n_Q}{n_\mathrm{sp}}\,. \tag{5.16}$$

In order to calculate an average P_P for a real image, P_X is approximated at every image coordinate by a histogram of pixel values. It is calculated from the neighboring pixels that lie within the limits of the currently treated search path. With this, a mean entropy is calculated for the entire image and is used in Equation 5.16.

Choosing the best parameters by simulation

The probability of correctly matching isolated pixels depends on the number of effective quantization levels of the surface texture. During the image capturing process, the road's natural texture is blurred by motion and defocus and is sampled spatially and quantified into discrete values by the image sensor. As seen in the previous sections, the blur from motion and from defocus depends on the focal length f_L, the aperture f_k, the exposure time t_E, the focusing distance s_o, the position on the road segment Y, and on the vehicle speed v. Under fixed lighting conditions, the exposure time can be determined from the focal length and the aperture. Thus, there are three free to choose parameters:

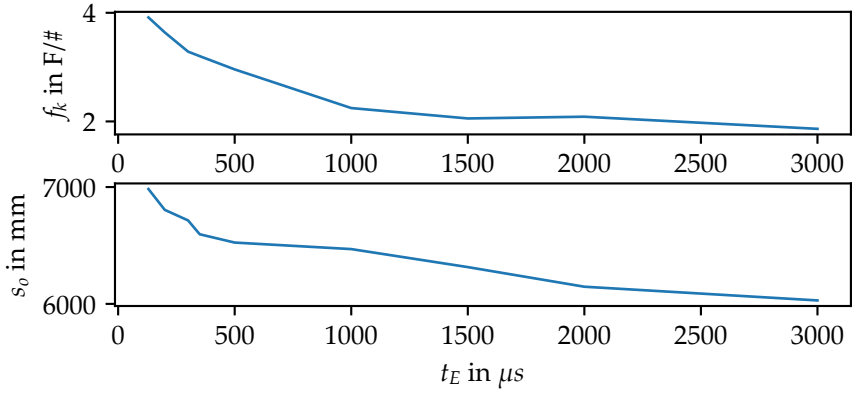


Figure 5.10: Optimal aperture and focusing distance for different lighting conditions, which are described by $f_\mathrm{L} = 25\,\mathrm{mm}$, $f_k = F/1.4$ and the exposure time shown in the graphs. The vehicle speed is $v = 50\,\mathrm{km\,h^{-1}}$.

f_L, f_k and s_o. f_k and s_o are determined by optimizing P_P for a given v. P_P then only depends on f_L.

To determine the effective number of quantization levels and, therefore, the probability P_P, the effects are simulated by artificially blurring and downscaling high-resolution close-up grayscale images of a typical asphalt concrete pavement and by then calculating the entropy on those images. The necessary exposure time has been found by experiments to be $t_E = 350\,\mu\mathrm{s}$ at an aperture of $f_k = F/1.4$[1] with a $f_\mathrm{L} = 25\,\mathrm{mm}$ lens on a sunny winter morning. This set of parameters describes a lighting condition. The exposure time for different focal lengths and apertures is calculated from this reference, as shown in Section 5.3.1. The blurring is implemented as a moving mean filter with the dimensions $b_x \times b_y$, which are determined as shown in Section 5.3.2. The resulting image is down-sampled, such that the resolution according to Section 5.2 is obtained. Then P_P is calculated. Since blur, resolution, and the search path's length depend on the position on the road segment, these steps are performed for three positions: at the front edge, at the rear edge, and in-between.

For a constant quality of the reconstruction result across the road segment, the minimum of all three values of P_P is optimized by adjusting the focusing distance and the aperture. Differential evolution is used for optimization. The result is shown for different values of f_L and a constant velocity $v = 50\,\mathrm{km\,h^{-1}}$ in Figure 5.9. By optimizing the minimum value of P_P, which is located either

[1] It is customary to write f-numbers preceded by F/.

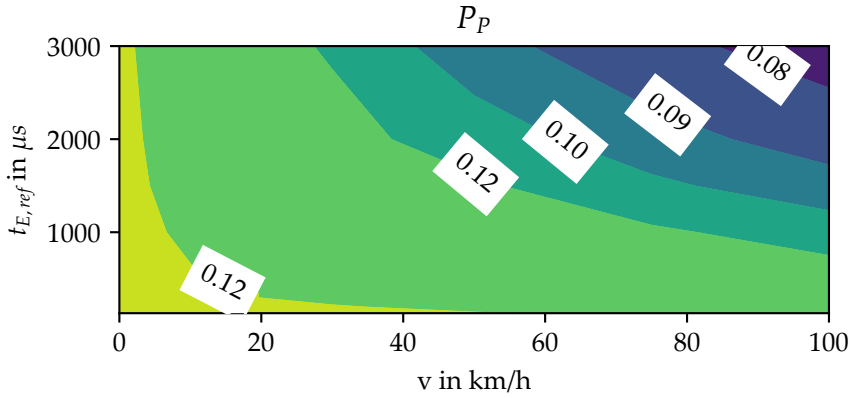


Figure 5.11: The expected probability of correctly matching isolated pixels depends on the vehicle speed and lighting condition. The lighting condition is specified by the necessary exposure time at $f_k = F/2.0$ and $f_L = 25\,\text{mm}$. The actual parameters t_E, f_k and s_o are determined by the optimization of P_P. The lane segment is $r_w = 2\,\text{m}$ wide and $r_l = 2\,\text{m}$ long.

at the front or at the rear edge, it results that the values at $Y = 0$ and at $Y = 6\,\text{m}$ are always the same. Their diagrams, therefore, lie on top of each other. It can be seen that the probability of correctly matching pixels decreases with a growing focal length, while the resolution in X- and Z-direction increases (Figures 5.4 and 5.5). Figure 5.8 shows the resulting blur over distance for different focal lengths. Looking at the result for $f_L = 10\,\text{mm}$, it shows that the result is not the same as an optimization of the maximum amount of blur would produce, as it is much higher at the front end of the road segment than at the rear end.

5.4 Optimal lens

An increasing focal length overall has the following effects:

- The resolution in X-direction stays constant at the front end and increases at the rear end of the road segment (Figure 5.5).
- The resolution in Y-direction decreases overall (Figure 5.5).
- The resolution in Z-direction stays constant at the front end and increases at the rear end (Figure 5.5).

- The reconstruction quality decreases (Figure 5.9).

The focal length that can be used is limited by the vehicle and the stereo camera system's position in it. Since the system is mounted behind the windshield, the field of view is limited by the hood. In the test vehicle used, 19 mm is the smallest possible focal length if a 2 m wide road segment is to be recorded. Choosing reconstruction quality over resolution in Z-direction, the best fitting commercially available lens has a focal length of $f_\mathrm{L} = 25\,\mathrm{mm}$. At the same time, this lens yields the optimal resolution in Y-direction at the rear end, as it has its maximum at 24.2 mm (Figure 5.5).

Because the aperture and the focusing distance are manually set on most lenses, they cannot be changed during operation. Therefore, values have to be found that are suitable for different lighting conditions. They are found by optimization at different lighting conditions, which are described by different exposure times t_E that would produce a well illuminated picture with $f_\mathrm{L} = 25\,\mathrm{mm}$ and $f_k = F/1.4$, as was shown in Section 5.3.3. The vehicle speed is set to $v = 50\,\mathrm{km\,h^{-1}}$. The result is shown in Figure 5.10. Figure 5.11 shows the expected performance under different lighting conditions and for different vehicle speeds if one chooses $f_k = F/2.0$ for an overall acceptable performance and a focusing distance $s_\mathrm{o} = 6300\,\mathrm{mm}$. The lighting conditions are measured by a reference exposure time $t_{E,ref}$ that would produce well illuminated pictures at $f_\mathrm{L} = 25\,\mathrm{mm}$ and $f_k = F/1.4$. The actual exposure time is calculated by Equation 5.13.

6 Experiments – Stereoscopic 3D Road Profile Reconstruction

The developed stereo vision methods are evaluated in the application of 3D road profile reconstruction with cameras mounted behind the windshield in Section 6.5, but first, the design of the employed image acquisition system is described in Section 6.1. In Section 6.2 the test scenes are described, and in Section 6.3, the determination of the achieved accuracy is discussed by comparing the measurements obtained with the stereo vision system to those obtained with industrial laser scanners.

6.1 Stereo camera system

The stereo camera system consists of two cameras, a circuit for generating a trigger signal, and a laptop computer for reading and saving the camera images. In Chapter 5 an image sensor with a resolution of $1920 \times 1200\,\mathrm{px}$ and a pixel size of $4.8 \times 4.8\,\mu\mathrm{m}^2$ was chosen. The dimensions of the road segment to be captured are defined by $r_\mathrm{w} = 2\,\mathrm{m}$ width and $r_\mathrm{l} = 7\,\mathrm{m}$ length, and it was found that in this case a lens with a focal length $f_\mathrm{L} = 25\,\mathrm{mm}$ is suitable. The cameras are mounted on an aluminum bar with a baseline $b = 1080\,\mathrm{mm}$, which is the widest possible baseline in the test vehicle. The cameras are both inclined

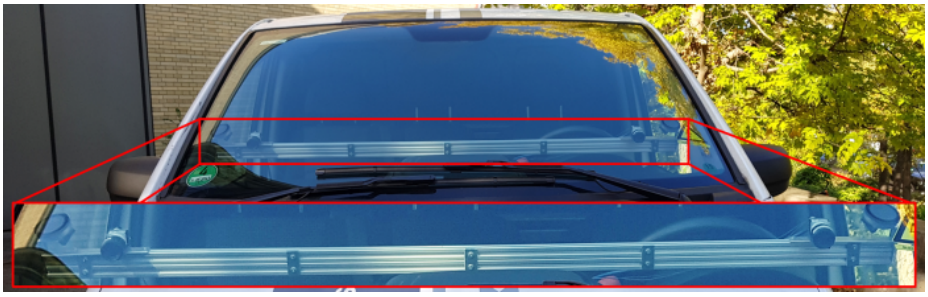


Figure 6.1: The cameras are mounted on an aluminium bar. The bar is attached to the windshield with suction cups.

Table 6.1: Camera specifications

Manufacturer	Basler AG
Model	acA1920-150uc
Color	Bayer filter
Sensor Type	Global shutter CMOS
Resolution (H x V)	$1920 \times 1200\,\mathrm{px}$
Optical size	2/3 in
Pixel Size	$4.8\,\mu\mathrm{m} \times 4.8\,\mu\mathrm{m}$
Synchronization	hardware trigger
Interface	USB 3.0

Table 6.2: Lens specifications

Manufacturer	Kowa
Model	LM25JC1MS
Focal length	25 mm
Aperture F/#	1.4–16
Resolution	120 lp/mm–100 lp/mm

around the horizontal axis by $\theta = \pm 6°$ to each other. In the case of static scenes, the bar is mounted on a tripod. For dynamic scenes, the bar is attached to the windshield of the vehicle with suction cups from the inside, as shown in Figure 6.1. In both cases, the bar with the cameras is rotated around the vertical axis φ towards the road.

6.1.1 Camera description

The specifications of the employed camera are shown in Table 6.1. It has a global shutter sensor (see Chapter 5) to handle objects under motion. The sensor is equipped with a Bayer filter to produce color images, as these are thought to help in a future classification of defects. The cameras are equipped with directly coupled and optocoupler isolated I/O lines. They can be triggered by a rising or falling edge of a signal applied to an input.

The lens specifications can be found in Table 6.2. It has the required focal length and an aperture that can be manually set in the required range. The optical resolution between $120\,\mathrm{lp\,mm^{-1}}$ and $100\,\mathrm{lp\,mm^{-1}}$ (line pairs per millimeter) does not limit the spatial sensor resolution of $4.8\,\mathrm{px\,mm^{-1}}$.

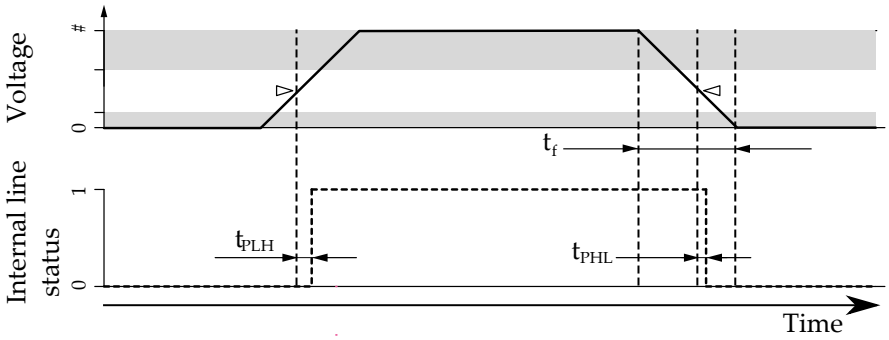


Figure 6.2: Analog external signal and associated internal line status with propagation delays for the employed cameras [2].

6.1.2 External trigger and asynchronism

The trigger signal is externally generated by a microcontroller board and is applied to both cameras in parallel. The delay between a trigger signal and the start of exposure is deterministic, except for the propagation delay t_{PHL} or t_{PLH}, which is the time between reaching a transition threshold voltage and when the camera reacts by changing its internal line status [2]. The timing diagram is shown in Figure 6.2. The exact values of the threshold are unknown, but the fall time of the setup was measured and is $t_f = 50\,\text{ns}$. The propagation delay from high to low is typically $t_{PHL} < 0.5\,\mu\text{s}$ [2]. Therefore, the total maximum propagation delay is $50\,\text{ns} + 0.5\,\mu\text{s}$ and the maximal asynchronism is $\approx 0.5\,\mu\text{s}$. It is negligible in comparison to exposure times of at least $100\,\mu\text{s}$. The directly coupled I/O lines have a shorter propagation delay than the optocoupler coupled ones, which is why the directly coupled lines are used to process the trigger signal.

6.1.3 Camera communication

The camera manufacturer provides an application programming interface (API) called "pylon API" for the C++ programming language, which is used to set parameters on the camera and to read uncompressed images from the camera via the USB 3.0 interface.

Exposure time calculation

The camera supports an automatic determination of the exposure time based on previous images' average gray values. If both cameras are set to determine

the exposure time independently, the exposure time may differ between the two cameras. That is because the road does not have a perfectly Lambertian surface, so the light is reflected differently depending on the angle between light source, surface, and camera. However, different exposure times lead to different degrees of motion blur between the left and right images and thus to a different appearance of the same object, which is problematic in the stereo reconstruction process. In order to solve this issue, the right camera is set to determine the exposure time automatically. The value is read and used on the left camera.

Data transmission and compression

The pylon API provides the images encoded in the RGB8[1] data format. Thus, a stereoscopic image pair generates 13.2 MB of data. If a video stream is to be saved for later analysis, the amount of data prohibits the uncompressed saving of the data. Intel central processing units (CPUs) starting with the Haswell architecture support video hardware encoding in the H.264 video format of high-resolution content [57]. This feature is marketed as "Intel Quick-Sync-Video" and enables the real-time encoding of video streams. It is supported by the libavcodec C-library, which is part of the FFmpeg project [98]. The library is used in the software for encoding the video and for the later decoding. It was found by experiments that the H.264 compression does not noticeably reduce the 3D reconstruction performance.

6.1.4 Camera calibration

The air-glass-air transition at the windshield influences the calibration parameters of the camera. As a result, the baseline of a stereo camera seems to be smaller than it actually is [47]. The curved surface of the windshield also affects the distortion parameters, and external forces can bend the bar on which the cameras are mounted. For these reasons, the calibration parameters change if the stereo camera system is mounted behind the windshield, and they also change if it is reinstalled in approximately the same position. Therefore, the stereo camera system must be calibrated when it is already installed in the vehicle.

For the calibration of the vehicle-mounted stereo camera system, images of a calibration target are taken through the windshield. The target is a printed circular point pattern glued to a wooden board. It is shown in Figure 7.4b. It contains 481 points and has dimension 1080 mm $\times$ 740 mm. The calibration parameters are determined by the MPT method described in Section 2.6.4.

[1] In the RGB8 data format, the color channels red, green, and blue are encoded by 8 Bits each.

6.2 Test scenes

The experiments are performed on static and dynamic scenes. Static scenes are considered because the results are compared to laser scanners and the terrestrial laser scanner used for static scenes (Section 6.3.3) has much higher accuracy than the laser line scanner mounted on the mobile mapping vehicle (Section 6.3.4) that is used for dynamic scenes. Furthermore, the terrestrial laser scanner's accuracy is given in a datasheet, while the accuracy of the scanner mounted on the vehicle was estimated from measurements. The dynamic scenes illustrate the performance of the stereo methods under difficult conditions.

Static scenes were captured with the stereo camera system mounted on a tripod. The viewing angle and the height of the cameras are similar to those in the test vehicle. For a large depth of field and since motion blur is not an issue, the aperture is set to $f_k = F/8$ and the exposure time to $t_E = 40\,\mu s$. Dynamic scenes are captured during the drive with $v \approx 70\,km\,h^{-1}$. The first set of images was captured in the morning under low lighting conditions. Therefore the aperture was set to $f_k = F/1.4$. The exposure times were determined automatically. The second and third sets were captured later that day under better lighting conditions.

Figure 6.7b shows the right camera image of the first static test scene. The image shows an asphalt concrete roadway in the foreground and a cobblestone pavement in the background. Some parts of the road are hit by direct sunlight, others by indirect light. Some leaves are lying on the road, and others are squeezed flat onto the ground. A large crack in the asphalt concrete pavement can also be seen. Although it cannot be seen in the image, the crack is accompanied by a bump, and the cobblestone pavement has a surface depression. Figure 6.8b shows the second static example. It is a paved walkway with raised and lowered tiles. The surface has both shiny parts and shadows.

Figures 6.9b, 6.10b, 6.11b show the same segment of a highway that was recorded three times under different lighting conditions during the drive. Two repair patches can be seen. The major surface defects are not visible in the image. They are a step in the foreground, rutting on the right-hand side and an overall unevenness. Figures 6.12b, 6.13b, 6.14b show a similar segment under different lighting conditions. The dark parts in the upper right in Figures 6.13b and 6.14b are shadows from the guardrail. A change in the road surface is visible. The large surface depression is not visible on the camera image.

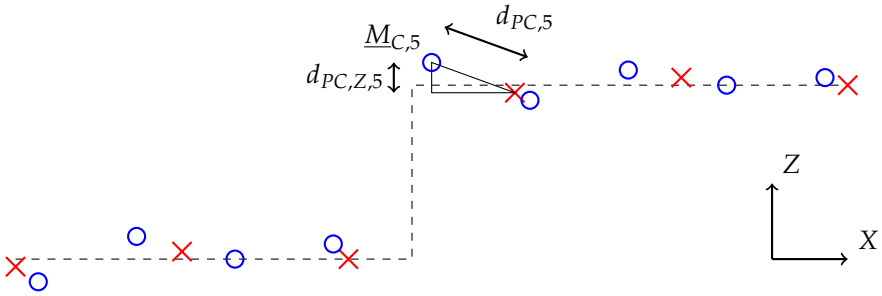


Figure 6.3: Calculation of the distances between two point clouds. Because the point clouds mostly resemble surfaces parallel to the XY-plane, and two nearest neighbors are usually not on top of each other, only the Z-component of the distance is considered the point cloud distance.

6.3 Accuracy

A measurement's accuracy is usually determined by comparing it to a reference measurement with much higher accuracy; however, in the present case, that is not available. The proposed stereo methods are applied to stereoscopic images of road surfaces and compared with measurements from industrial laser scanners. Static scenes are captured with cameras mounted on a tripod and scanned by a terrestrial laser scanner. Dynamic scenes are captured with cameras mounted behind the windshield of a moving vehicle, and the corresponding laser scans are performed by a company that uses laser line scanners mounted on a mobile mapping vehicle [43, 45]. The laser scan point clouds and the stereo vision point clouds are subject to measurement noise in the same order of magnitude.

Section 6.3.1 discusses how to compare the two noisy point clouds. The terrestrial laser scanner and the uncertainty of its measurements are discussed in Section 6.3.3. The laser line scanner and the uncertainty of its measurements are discussed in Section 6.3.4. Section 6.3.2 deals with the question of how to calculate the uncertainty of the stereo vision point cloud from the noisy observations.

6.3.1 Point cloud comparison

Both point clouds are subject to measurement noise, and it is assumed that the noise is Gaussian-shaped. Furthermore, the point clouds are not necessarily sampled evenly. Therefore, to compare two point clouds, the nearest neighbor of every point $\underline{M}_{C,i}$ of a compared point cloud is found in a reference point

cloud, and their distances $d_{PC,i}$ are computed, as shown in Figure 6.3. Since the point clouds mostly resemble surfaces that are parallel to the XY-plane, and two nearest neighbors are usually not on top of each other, only the Z-component of the distance $d_{PC,Z,i}$ is considered the point cloud distance. The total distance would lead to an overestimate. Since the point cloud coordinates are random variables, the difference also is a random variable

$$D_{PC,Z} \sim Z_{LS} - Z_{SV}\,, \tag{6.1}$$

where Z_{LS} is the Z-component of point coordinates of the laser scan point cloud as a random variable, and Z_{SV} is the Z-component of point coordinates of the stereo vision point cloud as a random variable. Following the guide to the expression of uncertainty in measurement (GUM) [75], their standard deviations $u_{Z_{LS}}$ and $u_{Z_{SV}}$ are regarded as their measurement uncertainties. The standard deviation of the difference is given by

$$u_{D_{PC,Z}} = \sqrt{{u_{Z_{LS}}}^2 + {u_{Z_{SV}}}^2}\,. \tag{6.2}$$

That shows that the distance measurement noise is a combination of the deviation of the laser scan point cloud and the stereo vision point cloud.

6.3.2 Distance between laser scan reference and stereo methods

Point clouds are extracted from the stereo vision methods and are compared to those generated by the laser scanners, but for both, static and dynamic scenes, the geometric relationship between the stereo camera system coordinate system and the laser scanner coordinate system is unknown. Therefore, the point clouds are first aligned using the Iterative closest point (ICP) algorithm [6], which is implemented in the software "Cloud Compare" [44]. It assigns to every point of a compared point cloud the nearest neighbor from the reference point cloud. Then, the distance between every two neighbors is calculated. The compared point cloud is rotated and shifted to minimize the root mean square distance. New nearest neighbors are assigned, and the algorithm is repeated until convergence. As a result, the mean difference in Z-direction is approximately equal to zero. The point clouds are compared as described in Section 6.3.1. Therefore, the standard deviation of the stereo vision method can be isolated by

$$u_{Z_{SV}} = \sqrt{{u_{D_{PC,Z}}}^2 - {u_{Z_{LS}}}^2}\,. \tag{6.3}$$

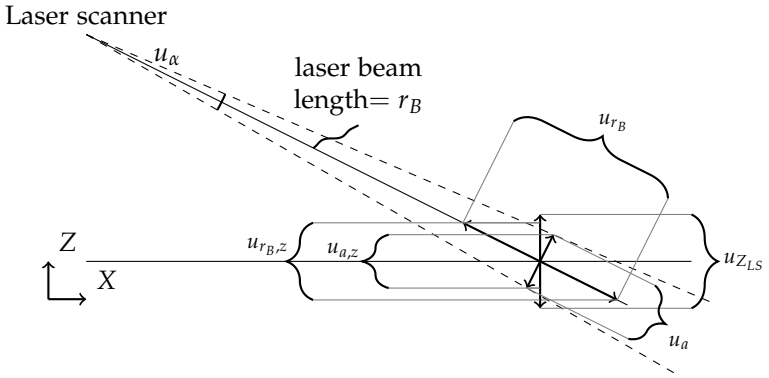


Figure 6.4: The uncertainty of a terrestrial laser scanner measurement is a combination of the uncertain beam direction and the laser distance measurement. In the present application, the combined uncertainty in Z-direction is of interest.

It is assumed that the bias of the laser scan measurements is zero. By aligning the stereo vision point cloud with the laser scan point cloud, the stereo vision point cloud's bias is also assumed to be zero. Hence, the empirical standard deviations are equal to the RMS values. RMS values are a commonly used error measure and are therefore used in the evaluation.

6.3.3 Terrestrial laser scanner

For static scenes, the terrestrial laser scanner "Z+F IMAGER 5006h" from Zoller + Fröhlich is used. Non-moving terrestrial laser scanners generate a 3D point cloud from 1D depth measurements by deflecting a laser beam around two axes, while the whole apparatus is mounted on a tripod [105]. The laser scanner used carries out the depth measurement by evaluating the phase position of the outgoing and incoming laser beams [105]. The measurement uncertainty in Z-direction $u_{Z_{LS}}$ therefore is a combination of the uncertainty u_{r_B} of the laser range measurement r_B and the uncertainty u_α of the azimuthal laser beam angle α. This is shown in Figure 6.4. Following the GUM, the azimuthal angular uncertainty u_α is propagated to the azimuthal distance uncertainty u_a by

$$u_a \approx r_B u_\alpha \,. \tag{6.4}$$

Table 6.3: Measurement uncertainty of terristrial laser scanner.

reflectivity	> 10%	> 20%	> 100%
u_{r_B} in mm	1.2	0.7	0.4
u_α in °		0.007	

The measurement uncertainty in Z-direction of both components add up. Hence, the total uncertainty is

$$u_{Z_{LS}} = \sqrt{u_{r_B,Z}^2 + u_{a,Z}^2}\,. \tag{6.5}$$

The uncertainty in radial direction further depends on the reflectivity of the measured surface (Table 6.3). The total uncertainty therefore depends on the location and reflectivity of the measured point.

6.3.4 Laser line scanner on mobile mapping vehicle

For dynamic scenes, a laser line scanner is mounted to the roof of a mobile mapping vehicle[2] [43, 45]. As the vehicle is moving forward, stripes of point measurements are concatenated to form a 3D point cloud. Therefore, the uncertainty of the depth measurement depends on the distance measurement of the Laser Light Detection and Ranging (LIDAR) measurement and the accuracy of the position estimation of the scanner.

The measurement uncertainty of dynamic scenes is not known, but the mobile mapping vehicle is equipped with two separate laser scanners, and two point clouds are provided. The point clouds have an offset and, therefore, are aligned using the ICP algorithm. The measurement noise is estimated by comparing the point clouds from both scanners. For the comparison, the distances between nearest neighbors are calculated as described in Section 6.3.1. If both laser scanners encounter the same noise, the noise of a single scanner can be recovered from the measurements by rearranging Equation 6.2

$$u_{Z_{LS}} = \frac{u_{D_{PC,Z}}}{\sqrt{2}}\,. \tag{6.6}$$

Experiments show an empirical standard deviation $u_{D_{PC,Z}} = 2.0\,\text{mm}$, thus $u_{Z_{LS}} = 1.4\,\text{mm}$. The laser scanner's standard deviation is a combination of the

[2]The data that is used in this work was acquired within the BMBF project (Bundesministerium für Bildung und Forschung / Federal Ministry of Education and Research) "Kooperative cloudbasierte Straßenzustandserfassung / Cooperative cloudbased road condition monitoring – StreetProbe". Funding reference number 01MD16006D.

measurement itself and the granular roughness of the road surface. However, this does not lead to an underestimation of the accuracy of stereovision methods in Section 6.3.2, as the cameras are not able to resolve the granular surface roughness due to their spatial resolution. Therefore, it is correct to subtract the combined uncertainty value. It is assumed that the terrestrial laser scanner does not measure granule roughness because the laser beam hits the surface at an acute angle so that the beam skips over the granules.

This approach of deriving the laser scanner's measurement uncertainty also neglects the noise of the position estimate of the laser scanners. Nevertheless, it is used because no better estimate is available. The RMS values for the dynamic laser scans therefore are donoted by RMS_{LS+N} in the experiments.

6.3.5 Reported accuracy value

The oblique viewing angle of the cameras on the road, as well as that of the terrestrial laser scanner, leads to an uneven distribution of the point cloud densities. The densities decrease with the distance to the cameras or laser scanner. Additionally, the accuracy of the near points is higher than the accuracy of the far points. Accordingly, the empirical standard deviation that is calculated on all points seems to be quite low. Therefore, the point cloud is divided into bins that are evenly distributed along the surface's Y-axis. The empirical standard deviation is then calculated per bin. Standard deviation over distance is plotted in Section 6.5. In order to have one single value per stereoscopic image to encode the accuracy, the mean value of the standard deviation per bin is used for the later comparisons. The mean value is divided by the range in which points from the laser scan measurements are encountered to relate the accuracy to the underlying geometry. This is done because the proposed stereo methods favor flat surfaces. The range is found independently for every bin.

6.3.6 Visualization of the results

In the experiments in Section 6.5, elevation images and difference images are shown from a camera's perspective (e. g. in Figure 6.7). The elevation images generated by stereo vision are a direct result of the stereo algorithms. The laser scanners' elevation images are generated by placing a virtual camera with the same parameters as the real right camera in a 3D scene with the measured point cloud. An image is created by drawing a voxel with a finite size for every measured point. They are colorized by a color map that corresponds to the points' Z-component. The difference images are generated in the same way,

but the voxels are colorized according to the difference. The calculation of differences was explained in Section 6.3.1.

6.4 Practical experiments

Elevation images are generated by the proposed methods described in Sections 3.2 and 3.3 from the stereoscopic images captured with the developed stereo camera system. With both methods, 128 plane hypotheses are used in the last iteration, which are evenly distributed around the mean road surface. The images are downscaled by the factors $5 \ldots 1$ in both dimensions. Thereby the search volume is altered from $\pm 150\,\mathrm{mm} \ldots \pm 50\,\mathrm{mm}$ around the mean road surface. The initial road surface is found from feature points in case of static scenes. In the case of dynamic scenes, the initial road surface is found from feature points only once. Its refined location is then used for the initialization of all other dynamic scenes.

For comparison, static scenes are measured with the terrestrial laser scanner, and dynamic scenes are scanned with the laser scanner mounted on a mobile mapping vehicle.

6.4.1 Traditional method

The method iproposed in Section 3.2, in combination with the similarity measure being used are referred to as: HMI, BilSub and Census in the following. The Census transform itself is calculated on $9 \times 9\,\mathrm{px}$ patches. The patch size over which the similarity measures are accumulated has dimension $\mathcal{N}_{\underline{m}} = 5 \times 5\,\mathrm{px}$. K_s is chosen depending on the similarity measure:

$$K_s = \begin{cases} 10^{-6} & \text{for HMI} \\ 10^{-2} & \text{for BilSub} \\ 5 & \text{for Census} \end{cases} \tag{6.7}$$

Both, the patch size and the parameter K_s, were chosen by increasing them until obvious mismatches no longer occurred in the elevation images.

The SGM implementation uses 16 search paths: horizontally, vertically, diagonally, and the directions in-between. These are implemented by, e. g., traversing the path moving two steps to the left and one step down.

6.4.2 CNN method

The CNN method is implemented in the neural network framework PyTorch [80]. It is pre-trained on the KITTI 2015 stereo dataset [74] in disparity mode.

Then cost volume assembly is switched to the plane-sweep approach, and training is continued with data generated by the HMI method. The training set is created with the plane-sweep method with 64 equally spaced planes between ± 30 mm around the mean road surface. The training data consists of left and right color images as input and an elevation image as output. An example of a right camera image and the corresponding elevation image is shown in Figure 6.10c (although this particular example was used for validation only). The HMI method also gives the mean road surface location, which is necessary to calculate the plane hypotheses for training the network. The training dataset consists of 510 examples of road surfaces captured while driving from the inside of the vehicle. Another 20 static examples were taken outside the vehicle.

For training, the same 64 equally spaced planes between ± 30 mm used to create the dataset were used. For evaluation, the iterative method with 128 plane hypotheses was employed.

6.5 Results

The results for static scenes are discussed in Section 6.5.1 and those for dynamic scenes in Section 6.5.2. An overview of the results for both is given in Table 6.4.

6.5.1 Results on static scenes

Static scene #1

Figure 6.7a shows an elevation image that was generated by the laser scan measurement. It is shown from the right hand cameras' perspective (see Section 6.3.6). A bump accompanying the crack becomes visible, and a surface depression in the cobblestone pavement can be seen. The bump has a height of 25 mm and the depression a depth of 28 mm in relation to the mean surface.

Figures 6.7c to 6.7j show the elevation images generated by the stereo methods. They also show the corresponding difference between the laser scan and the stereo methods from the right camera's perspective. The differences are calculated in 3D space, as explained in Section 6.3.1. All four methods show high accordance with the laser scan. The Bump and the depression can be seen clearly. The difference image shows a correspondence within ≈ 2 mm over large parts of the surface. However, the cobblestone joints are not resolved on the stereo vision elevation images. The HMI and BilSub methods are able to reconstruct some of the leaves, whereas the Census and CNN methods smooth them out.

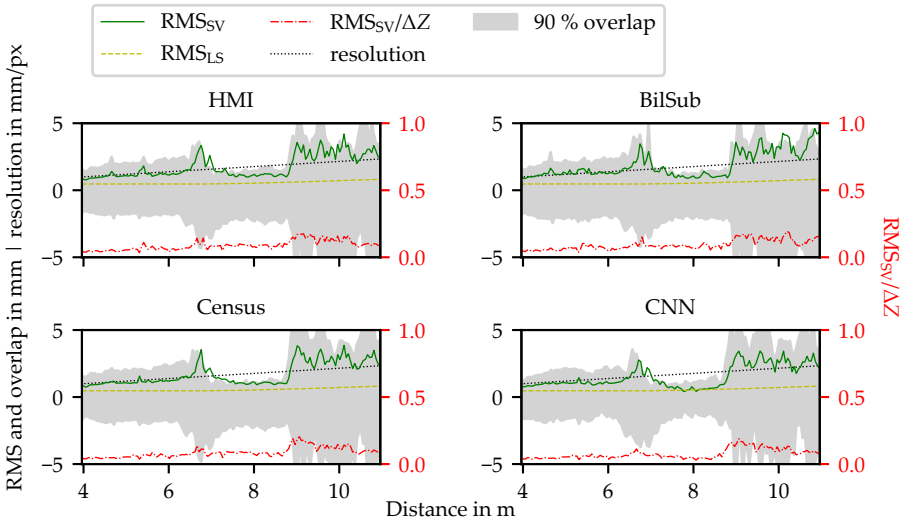


Figure 6.5: Static scene #1 – comparison of stereo method results to laser scan data.

The graph in Figure 6.5 shows the interval enclosing 90% of distances between laser scan point cloud and stereo vision point clouds for evenly spaced bins across the Y-axis (see Section 6.3.5). It also shows the RMS value of the laser scan RMS_{LS}, which is calculated based on the laser scanner's datasheet, and the RMS value of the stereo vision method RMS_{SV}, which is calculated as described in Section 6.3.5. The shown resolution is the theoretical resolution, which was calculated in Section 5.2. The enclosing interval is within $\pm 2\,\text{mm}$ close to the camera and jumps to larger values at a distance of $\approx 9\,\text{m}$. That is where the cobblestone pavement begins. The mean RMS_{LS} values are reported in Table 6.4. It shows that the CNN method produces the smallest error.

Static scene #2

Figures 6.8c to 6.8j show the elevation images and the corresponding difference images created by the different stereo methods for the second static example. The graph in Figure 6.6 shows the same values as in the previous example. All methods can reconstruct the elevation of partly smooth surfaces. The jumps between tiles are visually best reconstructed by the HMI and BilSub methods. The HMI method produces the largest errors in the upper left-hand corner. The Census and CNN methods both create smooth surfaces. The HMI and BilSub methods are able to reconstruct some of the leaves, whereas the Census and

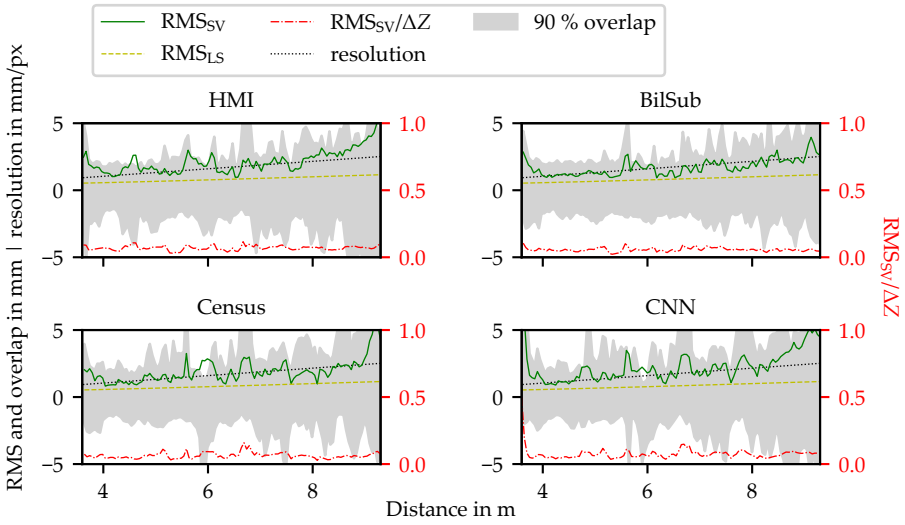


Figure 6.6: Static scene #2 – comparison of stereo method results to laser scan data.

CNN methods smooth them out. These visual results are confirmed by the RMS values in Table 6.4. The best result is achieved by the BilSub method.

6.5.2 Results on dynamic scenes

Next, the reconstruction quality of dynamic scenes is examined. Figures 6.9, 6.10, 6.11 show camera images and visual results for dynamic scene #1 under different lighting conditions. The corresponding graphs in Figures 6.15 to 6.17 show the the same values as in the previous exmaples. The resulting motion blur and mean RMS values are found in Table 6.5. With values up to 19 pixels at long exposure times, motion blur is not negligible. The effect on the reconstruction quality depends on the stereo method. The HMI method is very sensitive for motion blur, which can be seen on the right-hand side, where the elevation of a large patch was not reconstructed correctly. The other methods are not as sensitive, but still, more errors appear under bad lighting conditions. That can be seen in the upper left-hand corner. However, the major surface defects (a step in the foreground, rutting on the right-hand side, and overall an unevenness) are reconstructed by all methods under all lighting conditions. Observing the graphs in Figures 6.15 to 6.17 it is noticeable that the differences between the laser scan and the stereo vision point clouds show the same parabola-like bias in all twelve cases.

Table 6.4: Results for different scenes with different settings. The best results of a scene are marked in bold, the worst in italic. The motion blur is given for the near and far end of the road segment.

				motion blur		HMI		BilSub		Census		CNN	
Scene	v in km/h	f_k F/#	t_E in ms	near in px	far in px	mRMS_{SV} in mm	$\frac{\text{mRMS}_{SV}}{\Delta Z}$	mRMS_{SV} in mm	$\frac{\text{mRMS}_{SV}}{\Delta Z}$	mRMS_{SV} in mm	$\frac{\text{mRMS}_{SV}}{\Delta Z}$	mRMS_{SV} in mm	$\frac{\text{mRMS}_{SV}}{\Delta Z}$
Static 1	0	8	40.0	0	0	*2.1*	*0.10*	2.1	0.10	1.9	0.09	**1.7**	**0.08**
Static 2	0	8	40.0	0	0	*2.2*	*0.07*	**1.7**	**0.06**	1.9	0.06	2.3	0.08
Dynamic 1	65	1.4	2.89	18.7	1.4	*3.2*	*0.18*	2.0	0.11	2.0	0.11	**1.3**	**0.07**
	70		0.37	2.8	0.3	*1.9*	*0.11*	**1.3**	**0.07**	**1.3**	**0.07**	**1.3**	**0.07**
	68		0.13	0.9	0.1	*2.1*	*0.12*	1.5	0.08	1.4	0.08	**1.3**	**0.07**
Dynamic 2	67	1.4	2.34	16.4	1.3	*3.6*	*0.15*	**1.7**	**0.08**	**1.7**	**0.07**	**1.7**	**0.07**
	70		0.38	2.8	0.2	*2.0*	*0.09*	1.2	0.06	1.3	0.06	**1.2**	**0.05**
	71		0.13	1.0	0.1	*2.5*	*0.12*	1.7	0.08	1.6	0.08	**1.3**	**0.07**

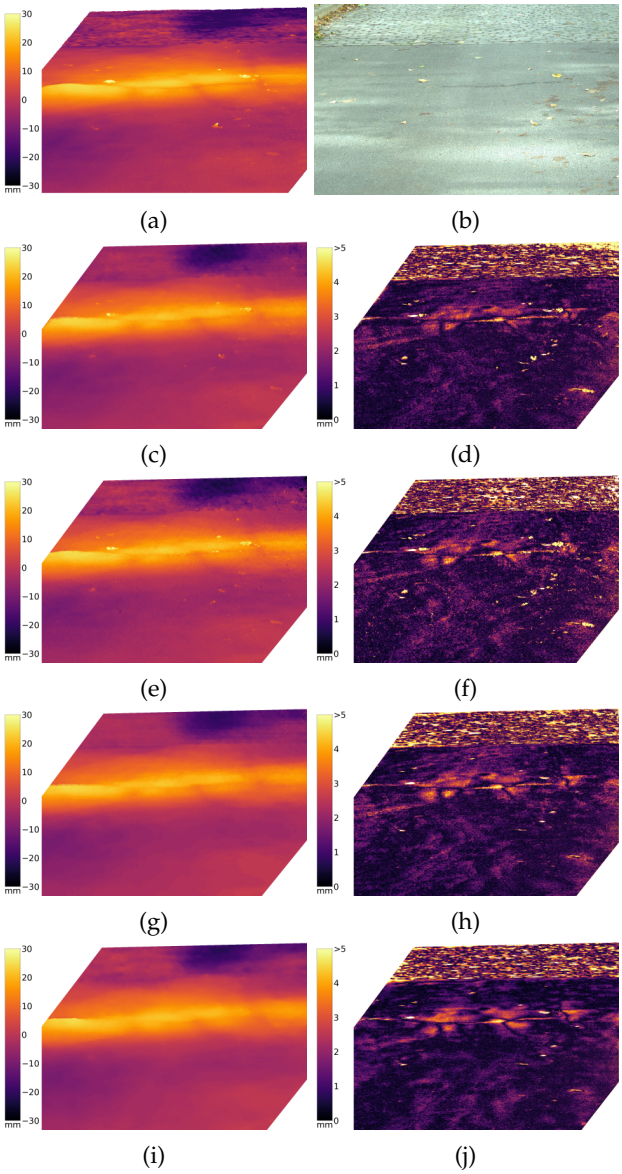


Figure 6.7: Static scene #1 – **a** laser scan elevation image, **b** right hand camera image, **c**,**d** HMI elevation/difference image **e**,**f** BilSub elevation/difference image, **g**,**h** Census elevation/difference image, **i**,**j** CNN elevation/difference image

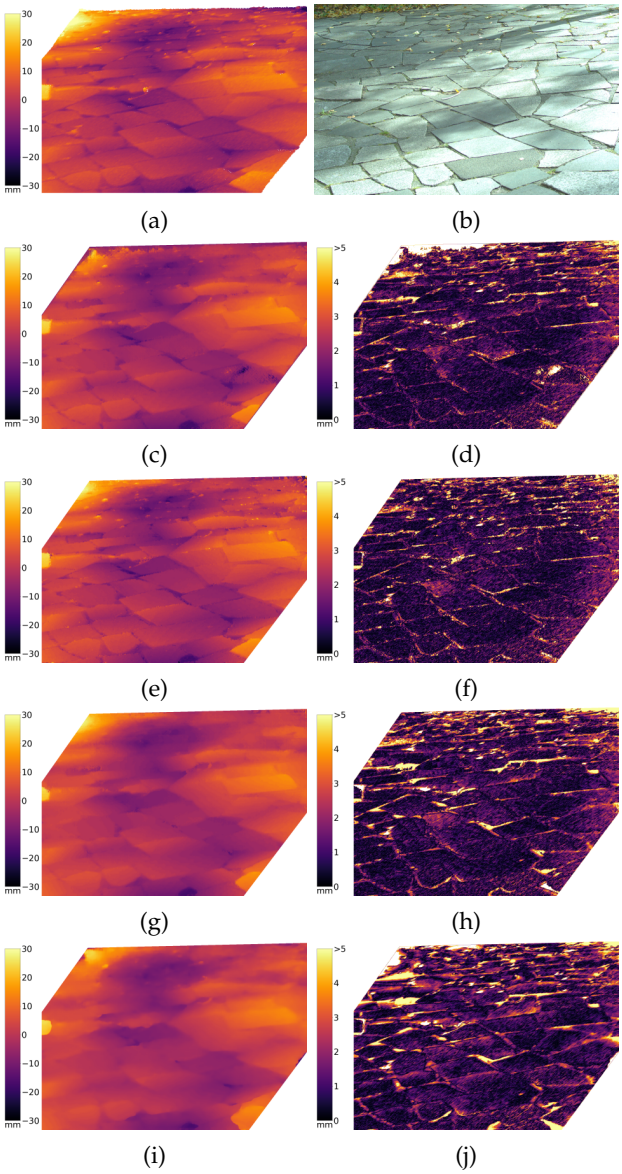


Figure 6.8: Static scene #2 – **a** laser scan elevation image, **b** right hand camera image, **c**,**d** HMI elevation/difference image **e**,**f** BilSub elevation/difference image, **g**,**h** Census elevation/difference image, **i**,**j** CNN elevation/difference image

Figures 6.12, 6.13, 6.14 show a similar road segment under different lighting conditions. The results are similar to the previous one. All methods can reconstruct the large surface depression under all lighting conditions, and the HMI method is most sensitive to bad lighting conditions with high motion blur. That can be seen in the upper right corner and at the bottom edge. The parabola-like bias observed on the previous segment is not observed on this segment.

Table 6.4 shows that the CNN method performs best in almost all cases for the dynamic scenes.

6.6 Discussion

The static scenes show a characteristic of the different methods: the similarity measures that work on pixels (HMI, BilSub) reconstruct fine structures in more detail, such as the edges of the tiles or the leaves. Census, as a window-based method, loses the fine structure and produces smooth surfaces. That is because it is implicitly assumed that all pixels in a window belong to the same plane and thus have the same elevation in 3D space. For asphalt concrete surfaces, this is not a problem, because fine structures are rare. Usually, they have an overall smooth surface. This can be seen in the first static scene, where the Census method outperforms HMI and BilSub. The joints between the cobblestones are not important in this comparison, because none of the methods could reconstruct them. The second static scene contains many fine structures, and, therefore, BilSub performs best.

On the one hand, these results depend on the choice of the window size used by the Census transform. They also depend on the choice of the patch size on which the similarity measures are accumulated. On the other hand, the Census transform needs a minimal window size to work with to match windows robustly. The size over which similarity measures are accumulated was the same throughout all experiments, and, therefore, it does not explain the smoother surfaces if Census is used. The parameter K_s does affect the results. It puts more weight on the smoothing or data term, but the smoothness assumption itself is encoded in the smoothing term, as discussed in Section 2.5.4. Therefore, a smooth surface, as generated by the Census method, cannot be achieved by e. g. the BilSub method by simply adapting the parameter K_s.

Shadows and shiny surfaces do not have a strong influence on the reconstruction. In the second static scene, this is due to the direction of the sunlight coming from the right side, which produces similar shiny surfaces in both camera images, but also to the similarity measures used, which are robust against radiometric differences.

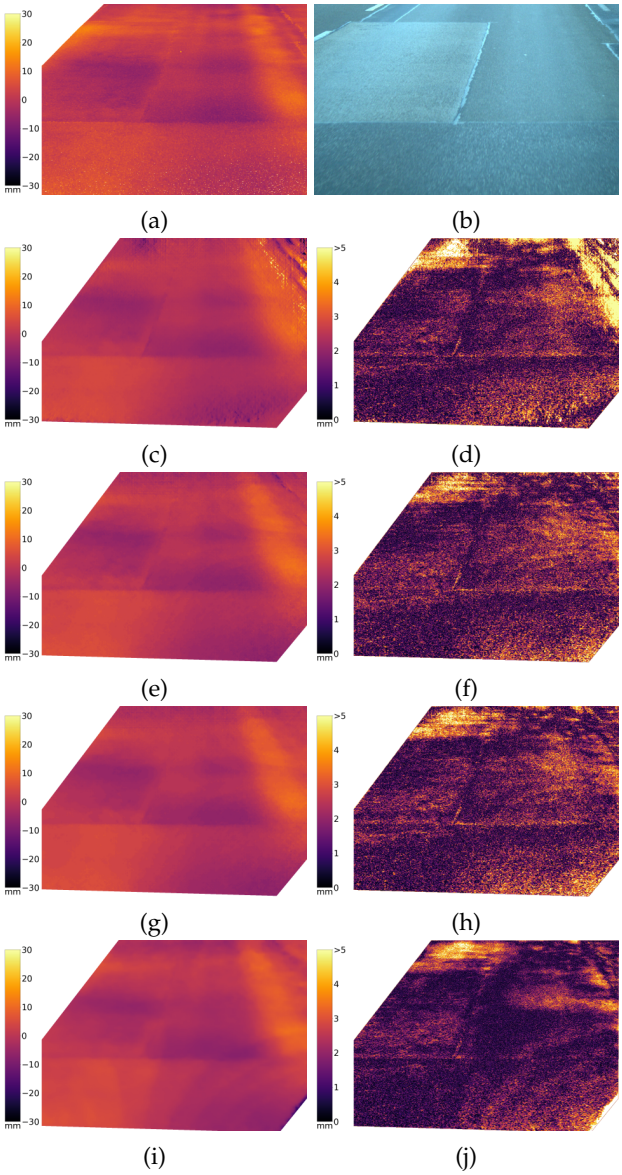


Figure 6.9: Dynamic scene #1, long exposure time – **a** laser scan elevation image, **b** right hand camera image, **c**,**d** HMI elevation/difference image **e**,**f** BilSub elevation/difference image, **g**,**h** Census elevation/difference image, **i**,**j** CNN elevation/difference image

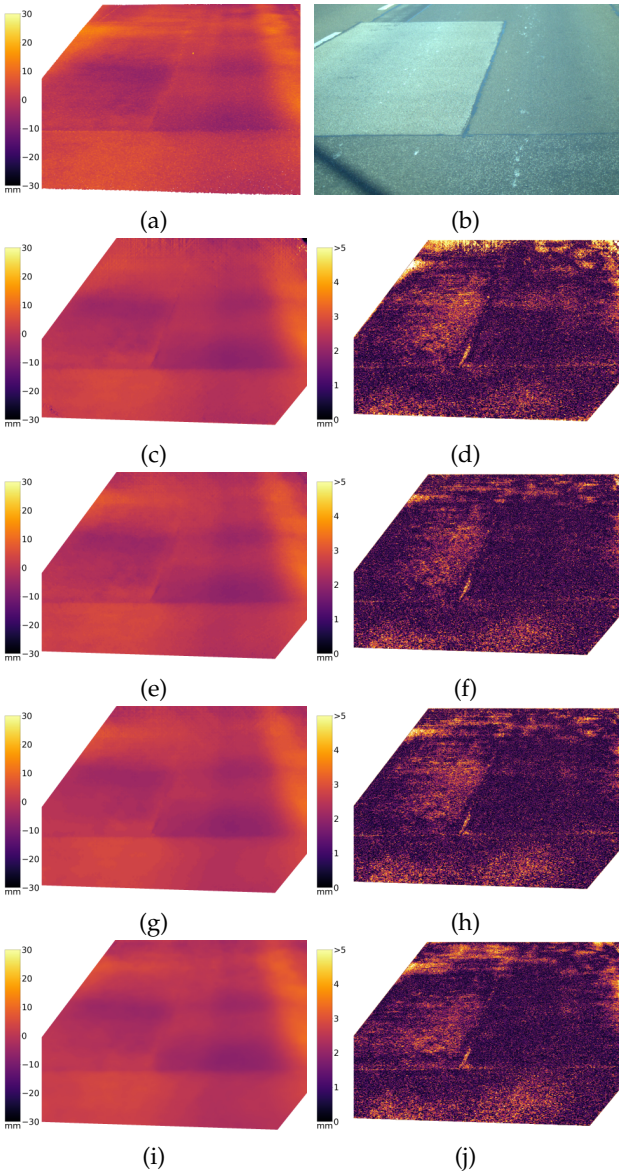


Figure 6.10: Dynamic scene #1, medium exposure time – **a** laser scan elevation image, **b** right hand camera image, **c**,**d** HMI elevation/difference image **e**,**f** BilSub elevation/difference image, **g**,**h** Census elevation/difference image, **i**,**j** CNN elevation/difference image

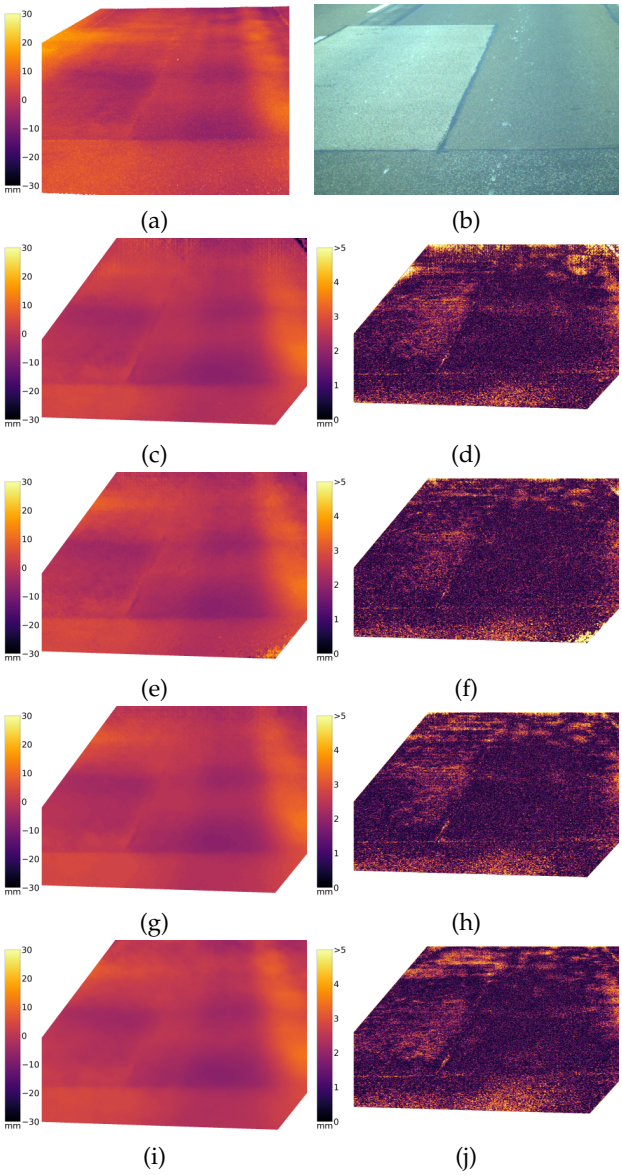


Figure 6.11: Dynamic scene #1, short exposure time – **a** laser scan elevation image, **b** right hand camera image, **c**,**d** HMI elevation/difference image **e**,**f** BilSub elevation/difference image, **g**,**h** Census elevation/difference image, **i**,**j** CNN elevation/difference image

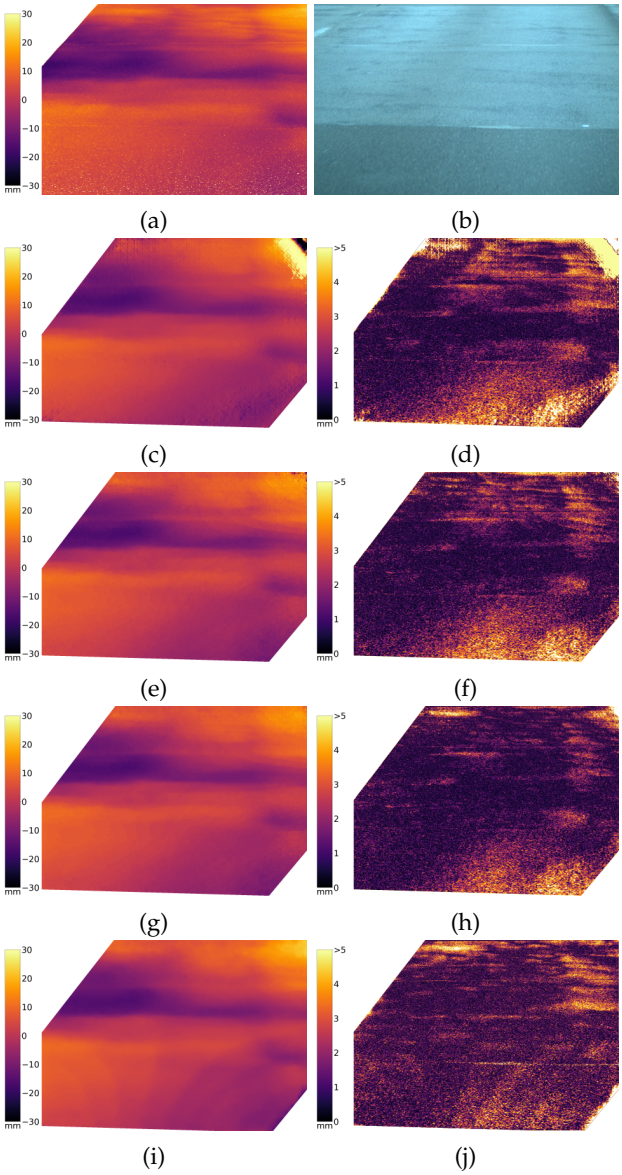


Figure 6.12: Dynamic scene #2, long exposure time – **a** laser scan elevation image, **b** right hand camera image, **c**,**d** HMI elevation/difference image **e**,**f** BilSub elevation/difference image, **g**,**h** Census elevation/difference image, **i**,**j** CNN elevation/difference image

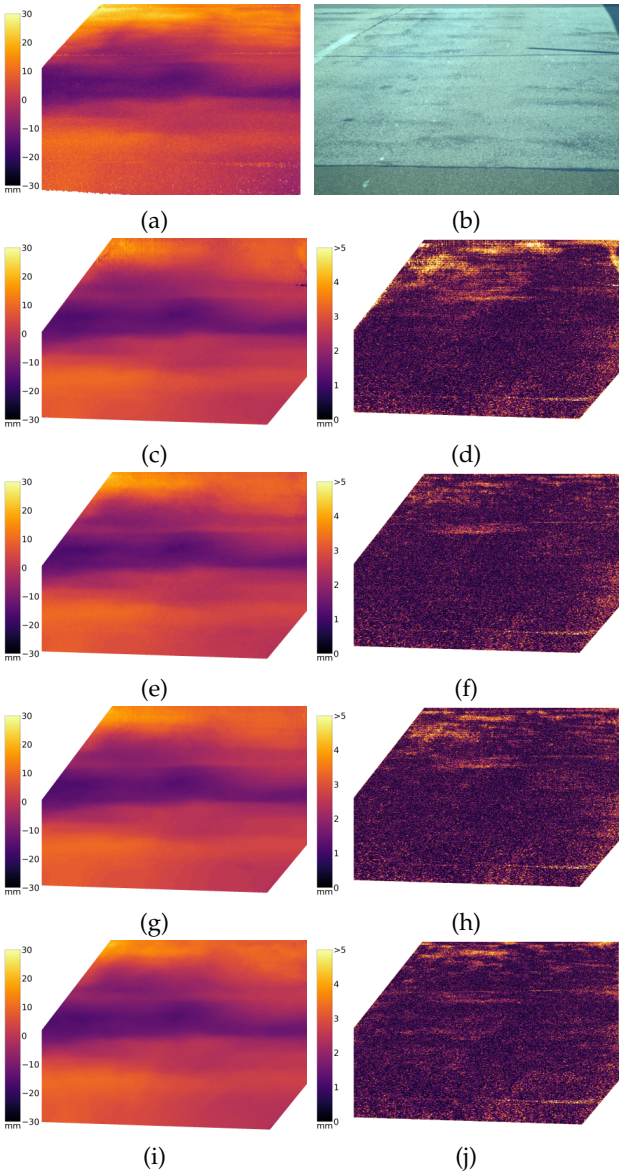


Figure 6.13: Dynamic scene #2, medium exposure time – **a** laser scan elevation image, **b** right hand camera image, **c**,**d** HMI elevation/difference image **e**,**f** BilSub elevation/difference image, **g**,**h** Census elevation/difference image, **i**,**j** CNN elevation/difference image

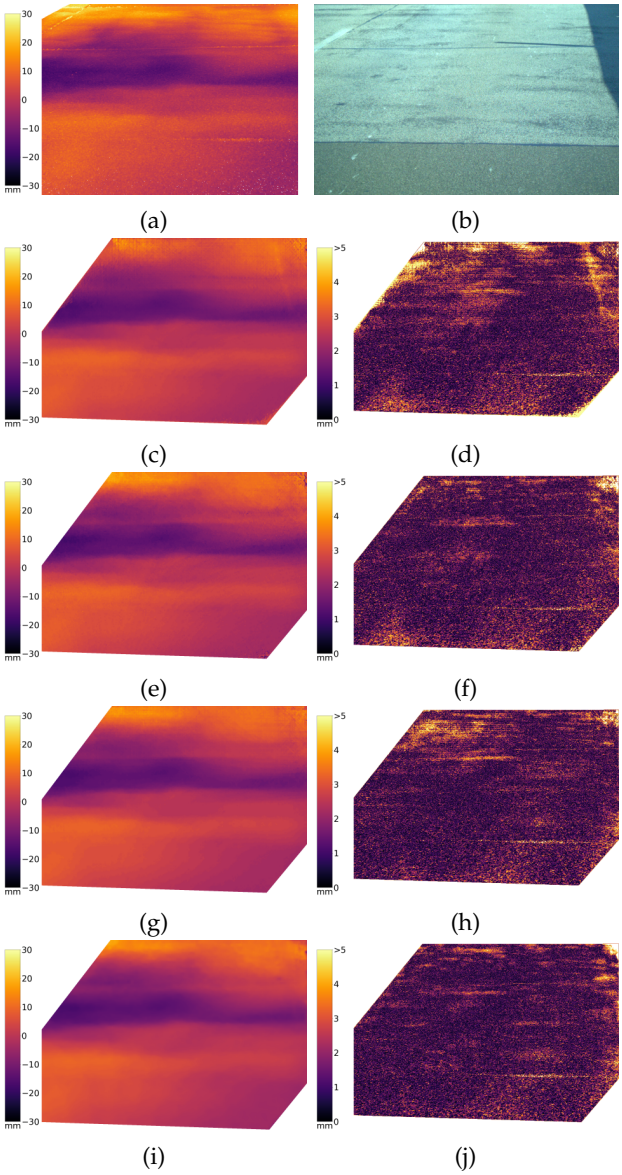


Figure 6.14: Dynamic scene #2, short exposure time – **a** laser scan elevation image, **b** right hand camera image, **c**,**d** HMI elevation/difference image **e**,**f** BilSub elevation/difference image, **g**,**h** Census elevation/difference image, **i**,**j** CNN elevation/difference image

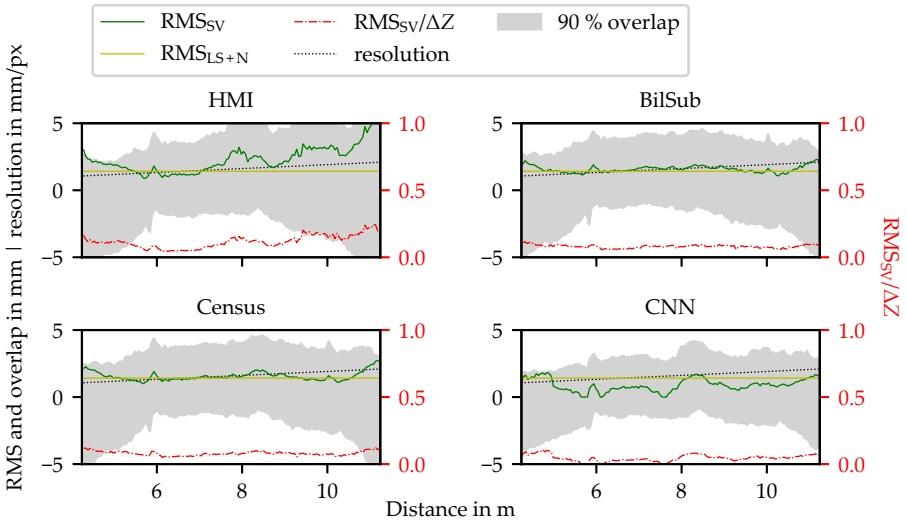


Figure 6.15: Dynamic scene #1, long exposure time – comparison of stereo method results to laser scan data.

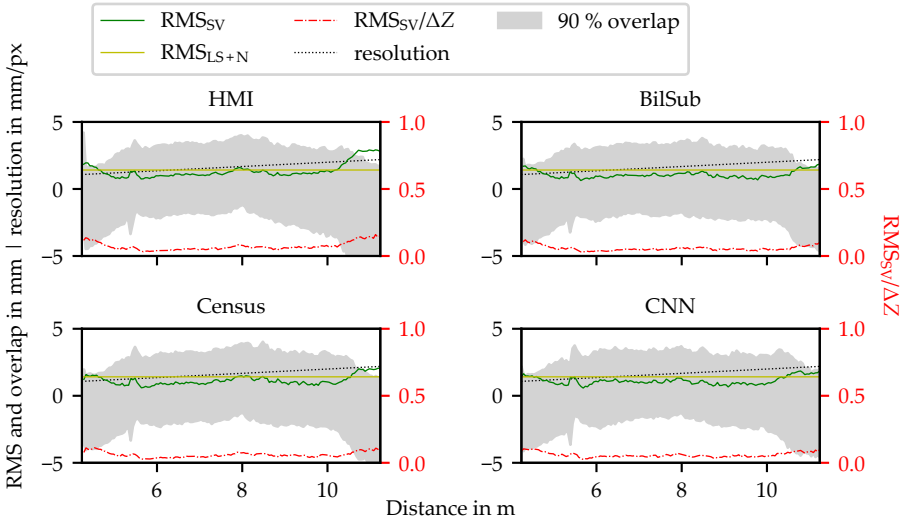


Figure 6.16: Dynamic scene #1, medium exposure time – comparison of stereo method results to laser scan data.

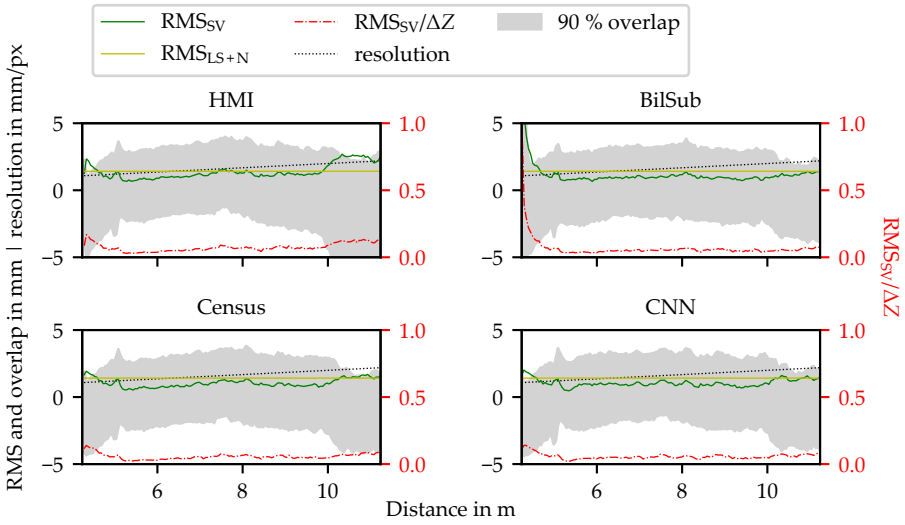


Figure 6.17: Dynamic scene #1, short exposure time – comparison of stereo method results to laser scan data.

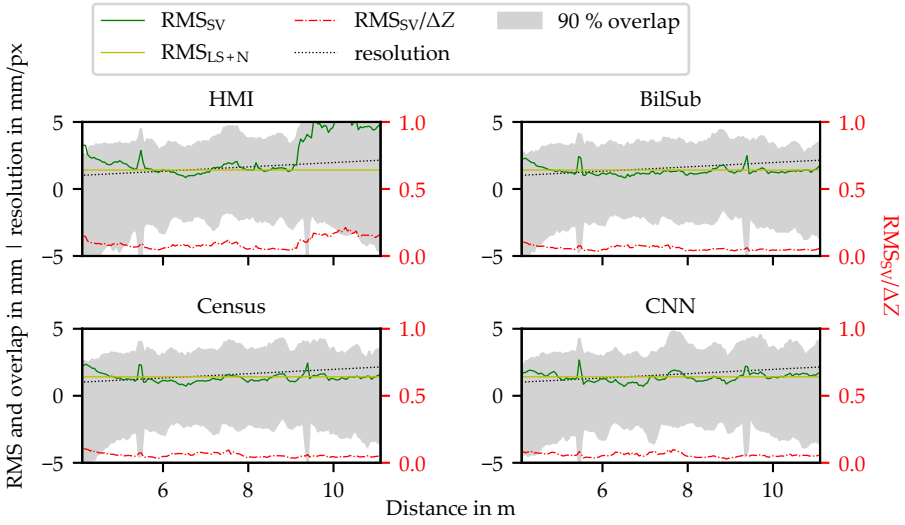


Figure 6.18: Dynamic scene #2, long exposure time – comparison of stereo method results to laser scan data.

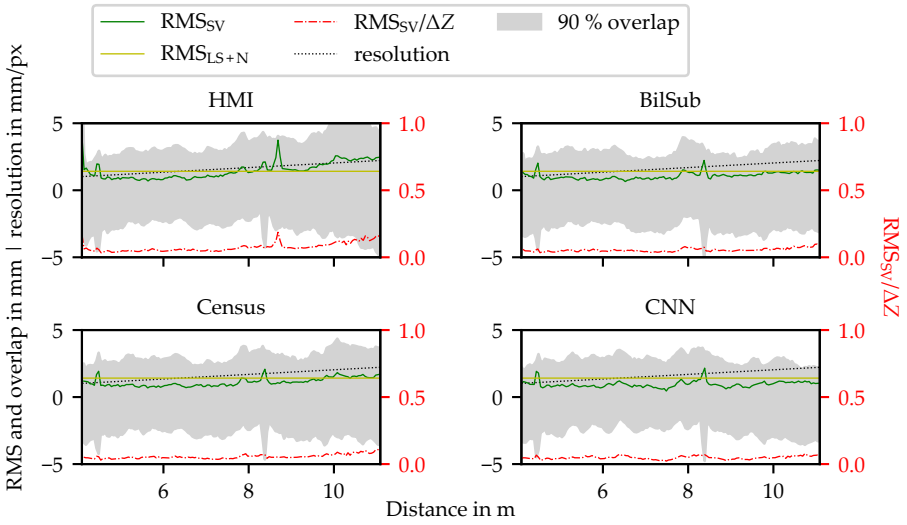


Figure 6.19: Dynamic scene #2, medium exposure time – comparison of stereo method results to laser scan data.

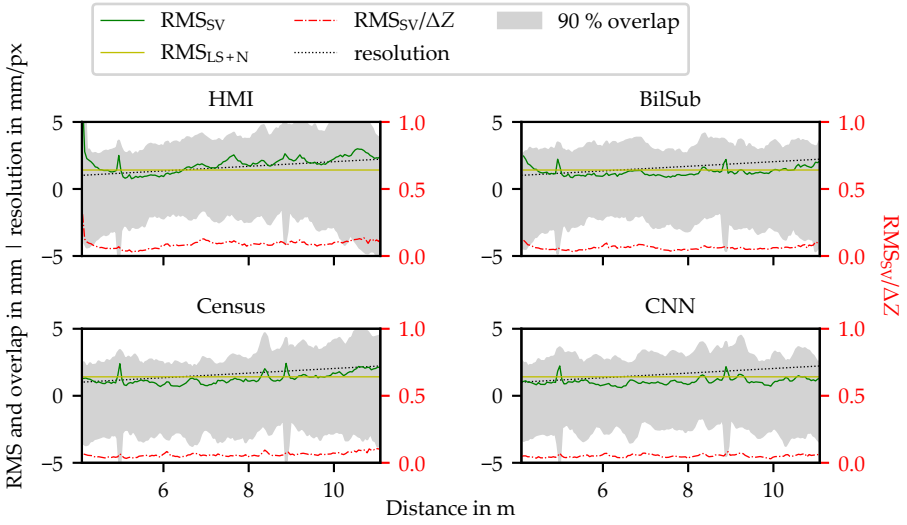
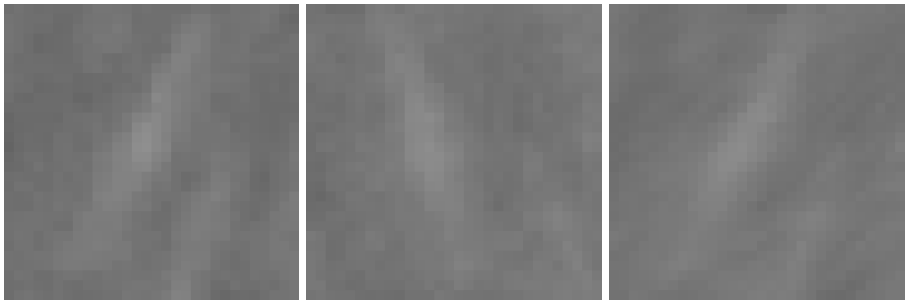


Figure 6.20: Dynamic scene #2, short exposure time – comparison of stereo method results to laser scan data.



(a) Detail view from the right camera image.

(b) Detail view from the left camera image if the perspective transformation is ignored.

(c) Detail view from the left camera image after the perspective transformation.

Figure 6.21: Detail view from dynamic scene #1 with long exposure time and strong motion blur. Only if the perspective transformation is taken into account, corresponding pixels are compared.

Due to a lack of training data, the CNN was trained on data generated by the HMI method. Laser scans were not used for training because a sufficiently large data set was not available and is difficult to obtain. Laser scanning and the acquisition of stereoscopic images would have to take place simultaneously, and the relationship between scanner and camera would have to be known precisely. That makes the measurement complex and expensive, especially for measurements recorded during the drive. Nevertheless, the CNN method produces interesting results. It produces rather smooth surfaces, and it loses the fine structure. The region-level features might cause that, but more importantly, the training set mostly consisted of asphalt concrete surfaces and did not contain many examples of tiles.

The more interesting result is that the CNN method performs better than the HMI method, i. e. it outperforms the training data. That becomes especially evident in the dynamic examples, where the CNN method performs best in almost all cases. An explanation for this result is the following: in the training data, the elevation is estimated correctly for most pixels, while for some pixels, the elevation is wrong. The correct pixels fit the model, which is learned by the CNN. The wrong pixels do not fit the model and appear as noise. Since the correct pixels outweigh the wrong pixels, the CNN can generalize the correct and reject the erroneous ones.

Comparing the results on dynamic scenes shows that the motion blur does not have as strong an effect as one might expect. Since the recorded surfaces are piecewise flat, the movement of each point on this surface follows a linear path

in both images. Therefore, the blurred texture is transformed perspectively correct by plane homographies. After that, blurred objects look exactly the same in both images. The motion blur results in a reduced spatial resolution in the direction of motion, which is in the vertical direction, and fine details in this direction can no longer be distinguished. In the horizontal direction, it has little effect. However, the horizontal direction is the direction that is primarily being searched with horizontally arranged stereo camera systems. The motion blur, therefore, has little effect on the estimated elevations.

If the surface texture is blurred, and fine details are lost, patches should be compared instead of individual pixels. That is done by accumulating similarity measures on patches in the traditional method, through the Census transformation itself, and it is done in the CNN when calculating feature maps. Before patches are compared, they first must be transformed perspectively correctly, as shown in Figure 6.21. If one transforms only a single pixel and compares the patch surrounding the corresponding pixels in both images, the perspective transformation is ignored. With the proposed method, the entire patch is transformed correctly.

Comparing the results of static and dynamic scenes, it is noticeable that the errors in dynamic scenes are smaller than in static scenes. Several reasons explain this. The static scenes that are used in this work are generally more difficult to reconstruct because the first contains cobblestones, and the second contains tiles, while the dynamic scenes all contain asphalt concrete surfaces. The static scene in Figure 6.5 shows an asphalt concrete surface between 4 m and 9 m distance. The errors of that part are smaller than the errors in all dynamic scenes. Furthermore, the results depend on the RMS_{LS} values, which are only roughly approximated for dynamic scenes. Therefore, the results for static and dynamic scenes cannot be compared quantitatively.

The reason for the parabola-like bias of the differences in the dynamic scene #1 is unclear. The images of the scene were taken on three consecutive test drives, but the results are similar on all three of them. The behavior is not observed in the dynamic scene #2 nor in the static scenes. An explanation could be a faulty laser scan measurement. The laser scan point cloud is composed of scan lines measured with laser line scanners mounted on a mobile mapping vehicle. If the position estimation of the moving scanner is inaccurate, this may lead to the shown deviation between the measured data.

6.7 Summary

In this chapter, the stereo methods proposed in Chapter 3 were evaluated on real-life examples of the target application of road surface elevation estimation.

The data was recorded with the developed stereo camera system. In the case of dynamic scenes, it was mounted behind the windshield, and pictures were captured during the drive.

The methods performed differently on the data of the different scenes. For fine details the BilSub method seems to be suited best. On smooth surfaces and blurred images, the CNN method has the best overall performance. Since the CNN was trained on data generated by the HMI method, it is quite possible that the results of the CNN could be further improved if better training data was available. In particular, the results of the CNN on fine structures should be studied in more detail, and it should be found out whether the reason for over-smoothed surfaces is in the training data.

Overall, the results showed that the stereo camera system combined with the stereo algorithms can reconstruct the road surface up to several meters in front of the vehicle. The accuracy of the system is within a few mm for large parts of the surfaces with a RMS value of $<2\,\mathrm{mm}$.

7 Experiments – Deep Learning Self-Calibration from Planes

In order to evaluate the proposed DLSC method, first, images are created by artificial cameras, such that all calibration parameters are known. The parameters are reconstructed by the DLSC method and for comparison by the MPT method that was described in Section2.6.4. Afterward, the proposed method is tested on real monocular cameras and on stereo camera systems.

7.1 Test cameras

Artificial images are created in two steps. First, a projective camera is simulated in the 3D computer graphics software blender [26]. Objects are modeled as flat, textured surfaces, and undistorted images are rendered. In a second step, the images are distorted by inverse warping. This requires the inverse of the distortion model (Equations 2.21–2.27), which is calculated by an approximate iterative algorithm [11].

In addition, three different real cameras are calibrated in the experiments:

- A GoPro Hero 4 Black action camera with a wide-angle lens and strong distortion (see Figure 7.2) is calibrated. The image sensor has 3840×2160 px. In the experiments, the images are first cropped and then downscaled to 1920×1200 px. The camera has a fixed focus.
- The same industrial camera that was used in the depth reconstruction experiments in Section 6 is calibrated. The image sensor and lens specifications were given in Tables 6.1 and 6.2. The resolution is 1920×1200 px. The focus is set manually but remains fixed during the experiments.
- A stereo camera system mounted behind a vehicle's windshield is calibrated. It consists of two of those industrial cameras from Section 6 equipped with a 16 mm lens. The resolution is 1984×560 px. The focus is set manually but remains fixed during the experiments.

7.2 Calibration targets

The DLSC method requires images of a plane from different perspectives. For calculating the similarity between views, it requires a textured surface.

7.2.1 Artificial targets

The artificial target is a plane that is equipped with a photograph as a texture. Various camera parameters are used to create different sets of calibration images. A set consisting of three images is shown in Figure 7.1.

7.2.2 Real targets for monocular camera calibration

For the calibration of real monocular cameras, an example of a flat surface is chosen that is available almost everywhere. It is the floor of a staircase. Figure 7.2 shows a set of three images that were taken by the GoPro camera. The same floor is also captured by the industrial camera.

7.2.3 Real targets for stereo camera system calibration

For a realistic scene that would appear in road profile measurements, the target is an asphalt concrete roadway in a perfect condition. The images were taken during the drive at a low speed. The set of calibration images, consisting of a single left and right image, is shown in Figure 7.3.

7.2.4 Targets for comparison

For the calculation of reprojection errors, images of a planar calibration target with control points are used. Artificial and real images thereof are created. The target used for real images is the same as was used in Chapter 6. It contains 481 points and has dimension 1080 mm $\times$ 740 mm. For the artificial images, a plane with the same dimension was modeled. It was equipped with the same texture that was printed and glued on the real calibration target. Examples of an artificial and a real image are shown in Figure 7.4.

7.3 Accuracy

Correct camera parameters are known only in the case of artificial images. Therefore, to determine the accuracy of the calibration, the geometric error

(a) Distorted first view. (b) Distorted reference view. (c) Distorted second view.

Figure 7.1: Artifical images (test set #2) of a flat surface that were used for self-calibration.

(a) Distorted first view. (b) Distorted reference view. (c) Distorted second view.

Figure 7.2: Images of a flat surface that were used for self-calibration of the GoPro camera. The tile pattern is not used by the algorithm, but it makes the distortion visible.

(Section 2.6.2) of back-projected image points of the planar calibration target is calculated.

In order to make the distortion parameters easier to interpret, the mean and median shift, that each pixel in an image undergoes, are calculated. That is useful because different distortion parameters can lead to a similar amount of distortion. In the case of artificial images, the distortion error is also calculated, which will be shown in Section 7.3.2.

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

Figure 7.3: Images of an asphalt concrete road surface that were used to calibrate the stereo camera behind the windshield.

(a) Artifical image of the calibration target from test set #3 (Section 7.4.1).

(b) GoPro camera image of the calibration target.

Figure 7.4: The MPT method is used for comparison. It requires images of a calibration target.

7.3.1 Reprojection error

Since the calibration target's location and orientation in relation to the camera are unknown, they must be determined first. Therefore, the reprojection error is iteratively minimized by Levenberg-Marquardt optimization [11] for every view j of the target

$$\min_{\underline{\mathbf{R}}_j, \underline{T}_j} \sum_i \left| \underline{\tilde{m}}_i - \delta \left(\kappa \left(\Theta \left(\left[\underline{\mathbf{R}}_j | \underline{T}_j \right] \mathbf{X}_i \right) \right) \right) \right|^2 . \tag{7.1}$$

Several images of the calibration target from different perspectives are captured and the RMS of all euclidian distances

$$\mathrm{RMS}_{\mathrm{re}} = \sqrt{\frac{1}{MN} \sum_{j=1}^{M} \sum_{i=1}^{N} \left| \underline{\tilde{m}}_i - \delta \left(\kappa \left(\Theta \left(\left[\underline{\mathbf{R}}_j | \underline{T}_j \right] \mathbf{X}_i \right) \right) \right) \right|^2} \tag{7.2}$$

is reported as the reprojection error in the results.

7.3.2 Distortion

The distortion is the distance between a pixel in distorted and undistorted image coordinates

$$\underline{d}_i = \underline{\tilde{m}}_i - \underline{m}_i \,, \tag{7.3}$$

where the distorted image coordinates are found by $\underline{\tilde{m}}_i = \kappa \left(\Theta \left(\kappa^{-1} \left(\underline{m}_i \right) \right) \right)$ for all pixel coordinates $\underline{m}_i$ of the image sensor. The mean and median values of $|\underline{d}_i|$ are reported in the results.

Table 7.1: Learning rates used in the DLSC optimization.

Parameter	Learning Rate
encoding of $\underline{\mathbf{R}}_i$	1e-3
$\underline{T}_i$	1e-3
f_x, f_y	1e-1
u_0, v_0	1e-1
K_1, K_2	1e-2
P_1, P_1	1e-2
K_3	1e-3
K_4, K_5, K_6	1e-4

For artificial images, the real distortion parameters are known, and the distortion error is calculated

$$\underline{e}_i = \hat{\underline{d}}_i - \underline{d}_i \,, \tag{7.4}$$

where $\hat{\underline{d}}$ is the estimated distortion and $\underline{d}$ is the true distortion. The relative error is calculated by

$$\bar{\underline{e}}_i = \left(\frac{e_{u,i}}{d_{u,i}}, \frac{e_{v,i}}{d_{v,i}} \right)^T . \tag{7.5}$$

The mean and median values of $|\bar{\underline{e}}_i|$ are reported as the mean relative error (MRE) and the median relative error (MdRE) in the results.

7.4 Practical experiments

The DLSC method is implemented in the neural network framework PyTorch [80], which requires GPUs for computation. In the experiments, NVidia GTX 1080 Ti GPUs with 11 GB of memory are employed. With an input image resolution of 1920×1200 px, the feature extraction network requires 8 GB of GPU memory per input image. Therefore, the feature extraction network for every individual input image is calculated on a dedicated GPU. For the feature extraction network, the learned weights from Chapter 3 are used. The Adam optimizer [65] with learning rates shown in Table 7.1 is used.

For comparison, calibration parameters are estimated by the MPT method that was described in Section 2.6.4. The implementation from the OpenCV library is used for this purpose.

7.4.1 Experiments with monocular cameras

The DLSC method uses a reference camera in whose space the images of all cameras are compared. To calibrate the distortion model in all parts of the image, it is advantageous if all images overlap the reference image in all areas. Therefore, the reference camera is pointed approximately vertically to the plane that is used for calibration. That makes it easier to ensure that all other images cover this part of the surface. Furthermore, it helps in finding the correct homography decomposition (Section 4.3.3).

In every experiment, three images of the textured plane are created for use with the DLSC method. Furthermore, one set of images of the calibration target with the point pattern for use with the MPT method and another set for the calculation of the reprojection error is generated.

Experiments with artificial cameras

Three artifical cameras with various camera parameters are created. They all have the resolution 1920×1200 px. The calibration parameters are shown in Table 7.2. In experiments #1 and #2, 14 images are used for the MPT method, and three images are used to calculate the reprojection error. In experiment #3, 10 and 3 images are used. That is sufficient because there is no noise in artificial images.

1. In the first experiment (artificial #1) the principal point of the artifical camera is shifted 100 pixels in horizontal direction away from the center. The pictures are radially distorted by the parameters K_1, K_2. All other distortion parameters are set to zero. For initialization, one paramater for $f_x = f_y$ and the distortion parameters K_1, K_2 are estimated. The parameters $f_x = f_y, u_0, v_0, K_1, K_2$ are calibrated by the MPT and by the DLSC method.

2. In the second experiment (artificial #2), the camera's aspect ratio is not equal to 1, and the principal point is displaced vertically and horizontally from the center. The same distortion parameters as in the first camera are used. Due to the different intrinsic parameters, this leads to a different amount of distortion. Since the aspect ratio is not equal to one, an initial value for both f_x and f_y, and initial distortion parameters K_1, K_2 are estimated. The parameters $f_x, f_y, u_0, v_0, K_1, K_2$ are calibrated by the MPT and by the DLSC method.

3. The artificial camera in the third experiment (artificial #3) resembles the real GoPro camera of the next experiment, i. e. the intrinsic and

distortion parameters are set to similar values as are found by experiment for the GoPro camera. The principal point is displaced from the center, and a higher degree of radial distortion than in the previous cameras is used. Tangential distortion is also present. The distortion parameters K_1, K_2, K_3, P_1, P_2 are defined and all other paramters are set to zero. For initialization, one paramater for $f_x = f_y$, and the distortion parameters K_1, K_2 are estimated. The parameters $f_x = f_y, u_0, v_0, K_1, K_2, K_3, P_1, P_2$ are calibrated by the MPT and by the DLSC method.

Experiments with real cameras

1. With the GoPro camera, one paramater for $f_x = f_y$, and the distortion parameters K_1, K_2 are estimated for initialization. Then, the complete distortion model with the parameters $f_x = f_y, u_0, v_0, K_1, K_2, P_1, P_2, K_3, K_4, K_5, K_6$, is calibrated by the MPT and by the DLSC method. 52 images are used for the MPT method and 51 to calculate the reprojection errors.

2. With the industrial camera, the distortion parameters are initialized to zero due to the low lens distortion. Although, from the datasheet, a pixel aspect ratio of one is known to be correct, two experiments are carried out: with $f_x = f_y$, and with f_x, f_y both as free parameters. In both cases, the principal point and the complete distortion model are calibrated by the MPT and by the DLSC method. 106 images are used for the MPT method and 36 to calculate the reprojection errors.

7.4.2 Experiments with stereo cameras

With the stereo camera system, a single stereoscopic image is used for the DLSC method. Once again, two sets of images of the calibration target with the point pattern are captured, one for use with the MPT method and one to calculate the reprojection errors. Since the windshield has an influence on the calibration parameters, all images are taken through the windshield. For initialization, the focal lengths are set to the same value $f_{x_L} = f_{y_L} = f_{x_R} = f_{y_R}$. Due to the low lens distortion, the distortion parameters are initialized to zero.

Because the number of intrinsic parameters that can be recovered from only two images is limited, the DLSC method is used to calibrate the intrinsic parameters $f_{x_L} = f_{y_L} = f_{x_R} = f_{y_R}, u_{0L}, v_{0L}, u_{0R}, v_{0R}$. Furthermore, the distortion parameters K_1, K_2, P_1, P_2, K_3 are calibrated for both cameras. The distance between the camera and the road surface and the baseline length are measured with a ruler and are fixed during the optimization.

(a) Distorted reference view.

(b) All three views are blended with the initial parameters in the reference camera view.

(c) All three views are blended with the optimized parameters in the reference camera view.

Figure 7.5: Images of a flat surface that were used to calibrate the GoPro camera. In (b), initial parameters are used to undistort and transform the images, and the blended images appear blurred. In (c), the parameters after the DLSC optimization are used, and the blended images appear perfectly sharp. The algorithm does not use the tile pattern, but it makes the distortion visible. [15]

With the MPT method, the parameters $f_x, f_y, u_0, v_0, K_1, K_2, P_1, P_2, K_3$ are calibrated for both cameras, but different conditions are enforced. These are shown in Table 7.3.

7.5 Results

7.5.1 Monocular cameras

The DLSC method is initialized by imprecise parameters at first. Images are undistorted, transformed, and compared. Due to the imprecise parameters, the images do not match at first in the reference camera space. This is visualized in Figure 7.5. Figure 7.5a shows the distorted reference image. Figure 7.5b shows the three blended, undistorted, and transformed images in the reference camera space, where the initial parameters have been used. After the optimization by the DLSC method, all three images match perfectly. That is shown in Figure 7.5c. The three images cannot be distinguished anymore, and they look like one single image. Only in the upper right-hand corner, it can be seen that the three images did not overlap in this part. Figure 7.5 shows the result for the GoPro camera, but the result is representative for all other artificial and real cameras.

Results for all experiments with monocular cameras are reported in Table 7.2. In each case the MPT, initialization, and DLSC methods are evaluated. For artificial cameras, results obtained with the real parameters are shown as well.

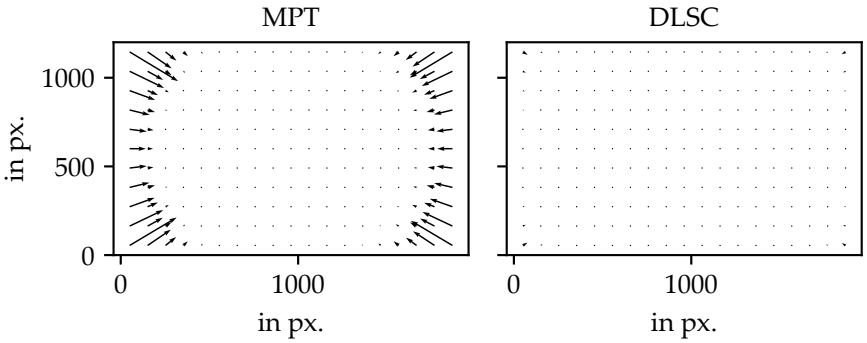


Figure 7.6: The displacement error generated by the MPT and the proposed DLSC method in the artificial test case #3. Arrows represent the displacement error with a factor of 10.

Results for artificial cameras

The results for the artificial experiment #1 show that both, the MPT and the DLSC method, can estimate the true intrinsic parameters to a precision of $\approx$1 px. The distortion parameters are recovered precisely by both methods. The reprojection errors are very small with 0.04 px and 0.05 px, respectively. The reprojection error of the MPT method is even smaller than the error that was achieved by the real parameters.

The results for experiment #2 show a larger error of the DLSC method compared to the MPT method, although the reprojection error still is within 0.09 px and the MRE of the distortion is within 0.1 %. In this case, the DLSC method searches two intrinsic parameters from three calibration images. This number is the theoretical limit under the assumption that the principal point is found through the distortion model, which makes it more difficult to solve the problem.

In experiment #3 the reprojection of the DLSC method is larger, but the MRE and MdRE are significantly smaller compared to the MPT method. Therefore, in Figure 7.6, the displacement error produced by both methods is compared for individual pixels. It shows that the DLSC method predicts the displacement much more accurately. This is particularly visible at the edges.

Table 7.2: *Note: The table extends horizontally over two pages.*Results for artificial found by the initialization method are referred to as Ini. Values highlight the best result.

		parameters							
camera	method	f_x in px	f_y in px	p_0 in px	p_1 in px	K_1	K_2	P_1 (×e-3)	P_2 (×e-3)
artifical # 1	-	1386.5		1060.0	600.0	-0.240	0.050		
	Ini.	1323.7		960.0	600.0	-0.176	0.019		
	MPT	**1387.2**		**1060.6**	599.3	**-0.240**	**0.050**		
	DLSC	1385.6		1058.7	**600.2**	**-0.240**	**0.050**		
artifical # 2	-	1386.5	1663.9	975.0	590.0	-0.240	0.050		
	Ini.	1289,8	1558,5	960.0	600.0	-0.140	-0.018		
	MPT	**1387.3**	**1664.7**	**975.6**	589.4	**-0.240**	**0.050**		
	DLSC	1384.8	1661.0	972.2	**590.0**	-0.239	0.049		
artifical # 3	-	840.9		975.0	590.0	-0.250	0.100	-1.000	-0.010
	Ini.	824.3		960.0	600.0	-0.261	0.115		
	MPT	841.9		974.5	588.8	**-0.250**	**0.104**	-0.755	-0.297
	DLSC	**841.4**		**975.0**	**589.9**	-0.253	0.106	**-0.980**	**-0.032**
GoPro	Ini.	695.2		960.0	600.0	-0.146	0.020		
	MPT	840.2		973.2	586.5	-2.701	1.808	-0.232	0.249
	DLSC	839.2		974.1	590.6	-0.249	0.095	-1.108	-0.078
industrial # 1	Ini.	5247.5		960.0	600.0				
	MPT	5275.4		894.9	690.6	-1.106	4.835	3.121	-2.779
	DLSC	5264.5		960.4	601.4	-0.095	1.811	-5.868	2.094
industrial # 2	Ini.	5247.5	5247.5	960.0	600.0				
	MPT	5284.5	5280.7	882.9	682.9	-13.654	4.757	2.458	-3.427
	DLSC	5163.0	5104.6	944.5	591.8	-0.036	-0.737	-5.320	-1.735

Results for real cameras

For real cameras, the true distortion is unknown, and therefore only the reprojection errors are compared. For the GoPro camera, the results show that the DLSC method slightly outperforms the MPT method.

In the first experiment with the industrial camera, where the aspect ratio is fixed at one, both methods' reprojection errors and the amount of predicted distortion are comparable. Although the real calibration parameters are unknown, it seems as if the principal point estimated by the MPT method is too far away from the center, whereas the DLSC method places it approximately at the center. Due to the low lens distortion, the parameters used for the initialization already achieve an equal reprojection error as the final parameters estimated by the DLSC method.

In the second experiment with the industrial camera, the aspect ratio is not fixed in the calibration. Measured in terms of the reprojection error, the

and real cameras. The rows marked by - show the real parameters. Parameters shown in gray are not optimized but are defined or are fixed. Values in bold

parameters				reproj-	distortion			
K_3	K_4	K_5	K_6	tion err. in px	median in px	mean in px	MdRE in %	MRE in %
				0.05	26.7	39.5	ref.	ref.
				1.69	21.8	31.6	0.55	4.12
				0.04	26.6	39.5	**0.01**	**0.03**
				0.05	26.7	39.5	**0.01**	0.05
				0.05	23.3	35.2	ref.	ref.
				2.16	16.7	26.8	0.90	3.01
				0.04	23.3	35.2	**0.01**	**0.03**
				0.09	23.3	35.2	0.03	0.10
0.010				0.11	61.6	65.1	ref.	ref.
				0.49	64.9	68.8	0.43	1.48
0.001				**0.08**	61.8	68.2	0.07	1.17
0.007				0.12	61.9	65.5	**0.04**	**0.13**
				1.42	49.7	65.1		
0.688	-2.416	0.963	1.430	0.20	70.5	88.3		
0.008	0.018	-0.012	0.043	**0.17**	61.2	82.8		
				0.27	0	0		
-9.919	-10.999	2.716	9.095	**0.23**	0.61	0.86		
1.881	-0.001	0.015	-0.593	0.27	0.79	1.00		
				0.27	0	0		
-10.253	-13.593	2.611	9.769	**0.23**	0.61	0.92		
28.640	-0.001	0.036	-2.537	0.74	0.72	0.89		

MPT method performs equally well as in the previous experiment, although all calibration parameters differ by a rather large amount. Especially K_1 and K_4 differ by an order of magnitude. In this case, the DLSC method performs significantly worse than in the first experiment. Due to the low distortion, the center of distortion, and thus the principal point, cannot be found through the distortion model as precisely. As a result, the other intrinsic calibration parameters are also not estimated accurately. The estimated amount of distortion stays approximately the same, but the reprojection error increases.

7.5.2 Results for a stereo camera system mounted behind the windshield

Calibration results for a stereo camera system mounted behind the windshield, produced by the MPT and the DLSC methods, are shown in Table 7.3. First, four

Figure 7.7: After the DLSC-Stereo calibration, the left and right camera images blend perfectly.

intrinsic parameters were calibrated by the MPT method. Because the principal point seemed to be too far away from the center, it was fixed at the center in the next experiment. With the DLSC method and only a single calibration image (and by including some measurements), one focal length for both cameras and a principal point for both individual cameras can be estimated. Therefore, these conditions were used with the MPT method. As a result, the reprojection error increased. Because the principal point again seemed to be very far away from the center, it again was fixed at the center. That decreased the reprojection error.

For a fair comparison of the MPT and DLSC methods, the baseline used in the DLSC calibration was adjusted to minimize the reprojection error of the calibration target. An inaccurate baseline has the effect of changing the scene's overall scale, and a small error of the baseline leads to a large reprojection error, making the values uncomparable. The DLSC method estimates the principal point to be close to the center in a reasonable position. Compared to the previous results, the reprojection error is larger, probably due to the inaccurately estimated focal length. That is expected because the distance between camera and road surface and the baseline length were measured with a ruler and themselves are subject to a measurement error. The visual calibration result is shown in Figure 7.7, where the left camera image is undistorted and transformed to the undistorted right camera image. It can be seen that both images blend perfectly.

Table 7.3: Results of the DLSC-Stereo method compared to the results of the MPT. l – left, r – right, all values are shown in pixels.

Method	Cam	f_x | f_y	p_x	p_y	error
MPT	l	3508.7	969.4	304.1	0.28
	r	3521.7	1019.6	319.4	
	l	3507.7	992	280	0.28
	r	3521.7			
	l	3452.7	1004.7	390.9	0.38
	r		1015.3	389.2	
	l	3516.6	992	280	0.30
	r				
DLSC	l	3440.4	990.5	281.0	3.13
	r		992.6	280.0	
	l	3440.4	990.5	281.0	0.94*
	r		992.6	280.0	

*For a fair comparison for calculating the reprojection error, the baseline was adjusted to fit the calibration target best.

7.6 Discussion

7.6.1 Monocular cameras

The experiments with artificial images show the potential of the proposed DLSC method. In the first artificial experiment, all parameters were reconstructed almost precisely. In the third artificial experiment, lens distortion was strong, and the principal point not located at the center. Nevertheless, all calibration parameters were also reconstructed accurately. The reprojection error is approximately the same as that generated by the actual parameters. In this experiment, the DLSC method achieves a distortion error that is significantly smaller than the distortion error achieved by the MPT method. The reason for this is the higher accuracy of the distortion model, as was shown in Figure 7.6. The DLSC method uses every pixel to estimate the distortion model. That is in contrast to the MPT method, which relies on feature points. This is an important advantage due to the poor extrapolation capability of the distortion model.

The second artificial camera is more difficult to calibrate. The distortion is not as strong, and at the same time, the aspect ratio is not equal to one. A strong distortion would help because the principal point is calibrated as a part of the distortion model. Since only three images are used with the DLSC method, the theoretical limit of parameters to be calibrated is reached in this experiment.

Therefore, both, the reprojection and the distortion error, are larger with the DLSC method than with the MPT method.

Similar results are achieved with the real industrial camera. It has a low lens distortion, making it difficult to calibrate the principal point as a part of the distortion model. Consequently, in the second experiment, where four intrinsic parameters were searched for, the parameters cannot be estimated precisely. If only three intrinsic parameters are searched, as in the first experiment, the results are much better. However, this behavior is not inherent to the DLSC method, but rather shows that the DLSC method still performs remarkably well with this absolute minimum number of calibration images. It can be assumed that all parameters would be estimated with even greater accuracy if more input images were used. However, since the number of input images is currently limited by the amount of available GPU memory, more input images could not yet be used.

The GoPro camera has an extreme lens distortion. Therefore, the principal point can be estimated precisely as part of the distortion model. That leaves theoretically two intrinsic parameters that could be calibrated. As a result, the only intrinsic parameter left, which is the focal length, is estimated precisely, which can be seen at the small reprojection error. In this case, the DLSC method outperforms the MPT method. If one compares the result with the artificial test case #3, this is probably due to more precise distortion parameters.

In all the experiments with artificial images, the reprojection error of the MPT method is smaller than the error that was achieved by the real parameters. The cause probably is the finite resolution and the two-step method of creating those images, where the projective camera is modeled first, and the distortion is added afterward. Another cause might be that the algorithm that finds the individual points of the calibration target does not find their precise locations, and the MPT method might adapt to these imprecise locations.

With all these results, one has to remember that only three calibration images were used in each case for the DLSC method, whereas for the MPT method many more calibration images were used.

7.6.2 Stereo camera system

The stereo camera system behind the windshield is particularly tricky to calibrate. Due to the influence of the windshield, it has to be calibrated after being installed (Section 6.1.4). The cameras are focused on the road surface, and the focus cannot be changed, as that would change the intrinsic camera parameters (Section 2.2). If the calibration method with a movable 2D calibration target is applied, the target must be captured in different poses and positions. For a reliable estimation of distortion parameters, the entire

image space has to be used, and for a good estimation of the focal lengths, the distance between cameras and target needs to be varied. The overlapping space that is captured by both cameras, however, is small. Most of it is located behind the road surface and is therefore not accessible without lifting the vehicle. If the calibration target is brought closer to the cameras, it quickly goes out of focus. These considerations show that calibration with a movable target is not ideal for the intended setup.

In the experiments with the MPT method, the coordinates of the principal points differ, depending on the fixed conditions, while neither the focal lengths nor the reprojection errors change much. It indicates that the location of the principal point and the distortion model could not be estimated accurately. It demonstrates the difficulty of calibrating the stereo camera behind the windshield.

The DLSC method produces a much larger reprojection error. It is caused by the imprecise length and distance measurements, and by the windshield's effect on the baseline length (Section 6.1.4). If the baseline is adapted to fit the calibration pattern best, the reprojection error is reduced by an order of magnitude.

The reprojection error generated by the DLSC method and corrected by scaling is $\approx$3 times as large as the error generated by the MPT method. Nevertheless, the result is remarkable because only one stereoscopic image pair was used to perform the calibration.

7.6.3 Non-planar calibration surface

One aspect that has not been considered so far is the flatness of the calibration surface. The DLSC method assumes that the surface is perfectly flat, which is not the case in reality, especially if some surface in a working environment is used for calibration that appears flat enough to the user. The transformation of the calibration images into the undistorted reference camera space is only valid for undistorted images of flat surfaces. If the surface had a shape that corresponds to the distortion model if projected onto an image, the shape of the surface would influence the calibration result. If the surface is not perfectly flat, and the shape's image does not fit the distortion model, the effect appears as noise, and the calibration result should still be ok. This assumption seems to be valid, as the calibration results are quite good, but the effect has not been studied in detail yet.

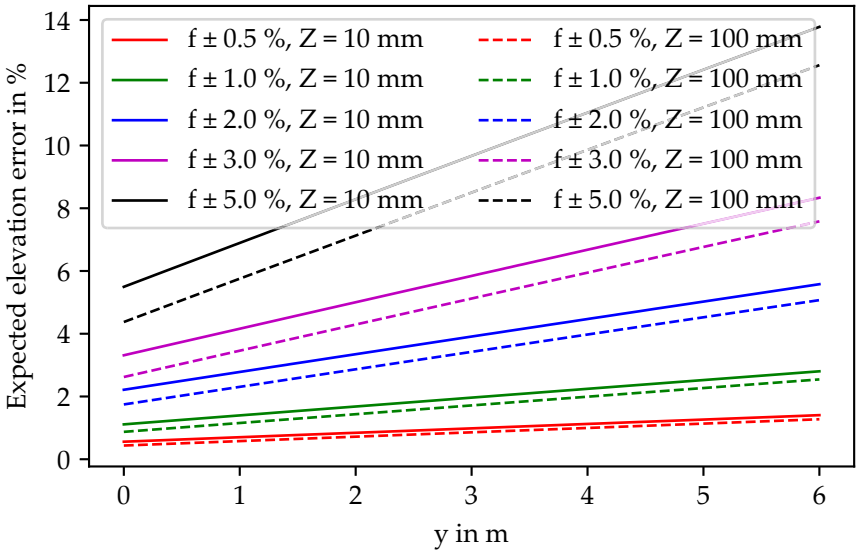


Figure 7.8: The expected elevation error depends on the measured height of the cameras, the distance to the cameras, and the accuracy of the focal length estimation.

7.6.4 The influence of an inaccurately estimated focal length

After the DLSC calibration, the undistorted and transformed images match each other perfectly, which can only be the case if the distortion has been removed and the homography between those images has been estimated correctly. Given the intrinsic camera parameters, a homography can be decomposed into the scene's geometrical layout, consisting of poses and orientations of cameras and plane [71]. The length of the baseline cannot be recovered from the decomposition. With a baseline of more than 1 m, one can assume that it is measured with a small relative error. For the intrinsic camera parameters, the assumptions are zero skew, an aspect ratio of one, and that the principal point is estimated independently from the projective camera model. Therefore, a reasonable assumption is that the focal length is the only parameter that is not estimated accurately. The effect of this inaccuracy on the elevation estimation is investigated in the following.

Following Section 3.1, the stereo camera setup with the base plane defines a homography. It is used to transform pixel coordinates from one image to the other. The base plane's homography is decomposed with an inaccurate focal length, and the result is used to perform the plane-sweep again. It

results in a slight change of pixel coordinates in comparison to the accurate plane-sweep. The difference of pixel coordinates can now be translated to a deviation in elevation that would be estimated by the plane-sweep method. These calculations are performed numerically for the stereo camera system used in the experiments. The result is an erroneous elevation estimate. It depends on distance, the elevation itself, and the inaccurate focal length, and is shown in Figure 7.8.

Assuming the focal length estimated by the MPT method is the true value (≈3500 px), then the focal length estimated by the DLSC method (3440 px) is off by ≈1.7 %. That leads to a predicted elevation error of ≈2 % – ≈5 %.

7.7 Summary

In this Chapter, the new method for camera self-calibration, which was introduced in Chapter 4, was evaluated. The experiments with mono cameras in Section 7.5 show that the method is capable of reconstructing intrinsic camera parameters as well as distortion parameters with high precision. The effort is reduced to a minimum, since only three views of a flat surface are required to estimate all camera calibration parameters. That is in contrast to established methods, which require many pictures of custom made calibration patterns.

In Section 4.4 the method was extended for use with a stereo camera system. In this case, two stereoscopic views are required to do a full calibration. In case the focal lengths have been calibrated beforehand, and the system's baseline is known, the distortion parameters of both individual cameras together with the extrinsic parameters can be estimated from a single stereoscopic picture of a flat surface. For comparison, in [111], a method is described that uses structured light for this purpose. Without the prior knowledge of the focal lengths, it was shown that adding a distance measurement enables the extraction of a shared focal length, together with the principal points, the extrinsic, and the distortion parameters from a single stereoscopic image. Although this technique can only be as accurate as the distance measurement, the accuracy may be sufficient, depending on the application.

8 Road Condition Monitoring

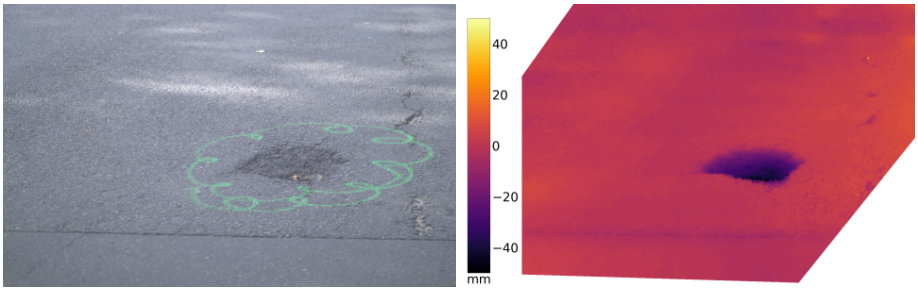
For the planning of road renewal at the network level, road construction authorities use indices that describe the overall condition of road sections [87, 93]. They are calculated from surface defects, such as cracks, patches, and potholes, and also take into account the surface shape, which may include rutting or an overall unevenness. In [35] the concatenation of condition variables describing certain conditions to an overall index value for an asphalt concrete pavement for inner-city roads is given. The index value is frequently used by road construction authorities in Germany. Its composition is shown in Figure 8.2. The reviews in [24] and [85] show that many papers have been published on the detection of surface cracks and potholes by the analysis of camera images, e. g. in [21]. Potholes can also easily be found in depth maps by comparing the depth of a contiguous region to a threshold, as can be seen from Figure 8.1. An approach is described in [102]. Approaches for patch detection are also available [84].

What is missing to calculate the index are the condition variables concerning the deformation of the road surface. Therefore, this section demonstrates the calculation of condition variables from the road profile of an inner-city road, generated from the stereo camera system. It is an example of how the developed methods are useful for road condition monitoring in practice.

8.1 Calculation of condition variables

In order to calculate the condition index, several individual condition variables are needed, as is shown in Figure 8.2. Their calculation is described in the following.

Simulated leveling board – PGR A simulated 4 m leveling board is placed on the measured points along the longitudinal road profile of the right wheel path. The distance vertically to the road profile in the middle of the board is the PGR value, as is shown in Figure 8.3. The board is shifted from support point to support point, and the result is a new derived profile. The maximum and average values of the derived profile of a road section are the PGR-M and PGR-A values [36].



(a) Right camera image

(b) Elevation image from the right camera's perspective

Figure 8.1: A large pothole that has been marked for repair by a road construction authority is clearly visible in the elevation image.

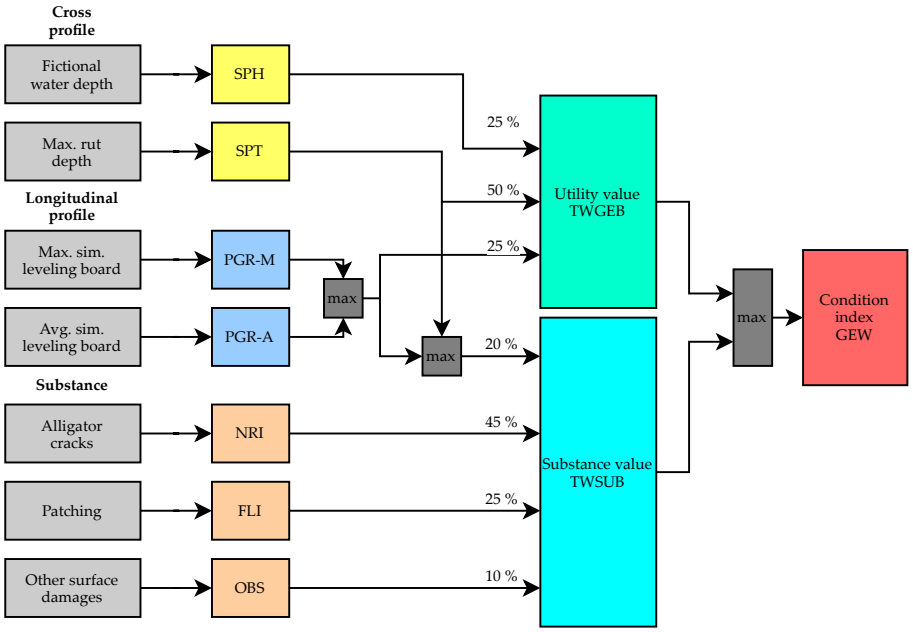


Figure 8.2: Overview of the calculation of the condition index [35].

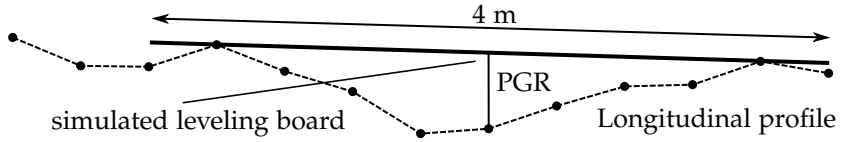


Figure 8.3: A simulated 4 m leveling board is used to measure the longitudinal road profile.

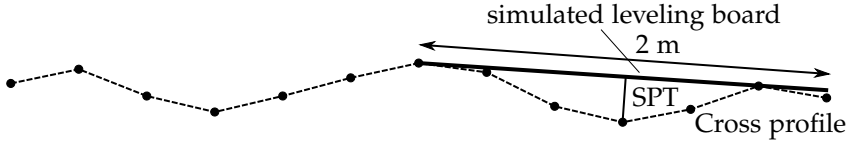


Figure 8.4: A simulated 2 m leveling board is used to measure the maximum rut depth.

Maximum rut depth – SPT The maximum rut depth is determined by moving a simulated 2 m leveling board across the road's cross profile (see Figure 8.4) [29]. The maximum distance between the board and the profile measured on the center's left-hand side is the SPTL value. The one measured on the right-hand side is the SPTR value. MSPTH and MSPTR are mean values of the former, calculated across road sections, and the maximum of the two is the maximum rut depth MSPH [36].

Figure 8.5: The fictional water depth is determined from the cross profile.

Fictional water depth – SPH The fictional water depth is the height up to which water could be filled into the cross profile on both sides from the center. [29, 36]. The values are SPHL for the left side and SPHR for the right side, as shown in Figure 8.5. The mean values of both are calculated on road sections, and the maximum of both values is the maximum fictional water depth MSPH.

Alligator (fatigue) cracking – NRI The ratio of the surface with irregular surface cracks to the total surface area is the NRI value [37].

Patching – FLI The ratio of the surface with patches to the total surface area is the FLI value [37].

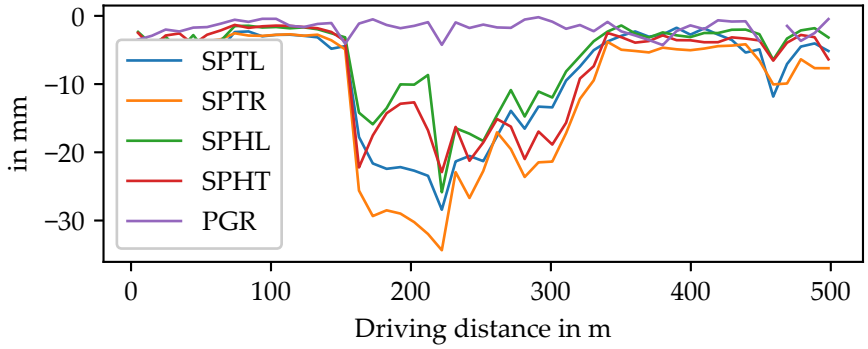


Figure 8.6: Road condition variables over traveling distance – Bismarckstraße from Kaiser-Friedrich-Straße to Wilmersdorfer Straße in Berlin, second lane from the left.

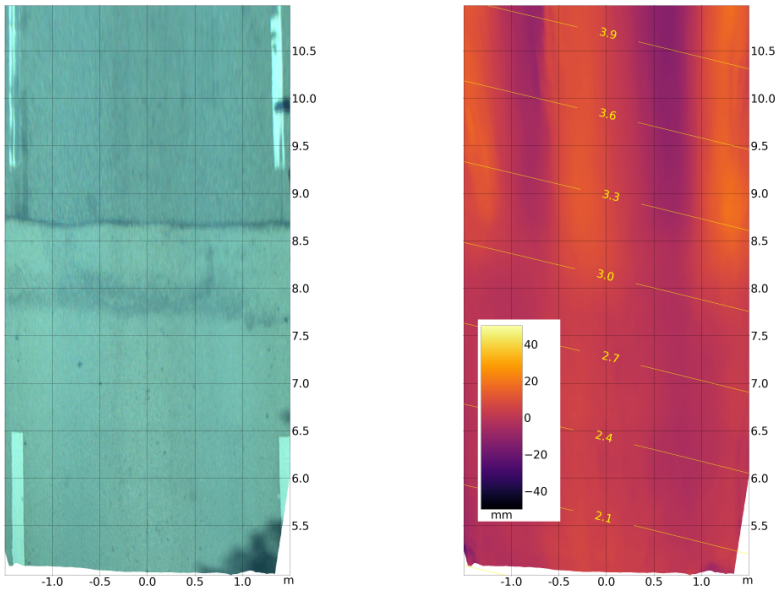
Other surface damages – OBS The ratio of the surface with other surface damages to the total surface area is the OBS value [37].

Each condition value is normalized to a range between 1 (great) and 5 (bad). 1 to 1.5 is considered an excellent condition, 3.5 corresponds to a condition from where measures should be taken and is considered the warning value, and 4.5 is the limit value. Inbetween, linear interpolation is used [37].

8.2 Demonstration of the developed system

For the determination of the condition variables SPH, SPT, and PGR, the requirements in [36] demand to record a lane width of at least 3 m. A suitable lens for the previously chosen camera is determined by following Chapter 5 and has a focal length of 16 mm. The stereo camera system is mounted behind the windshield, as was shown in Figure 6.1. A lane length of 6 m and a frame rate of 14 Hz are chosen. The frame rate corresponds to approximately one frame per meter at a driving speed of 50 km h^{-1}. Although not performed in this work, this would enable a fusion of data points to reduce noise. Camera calibration is performed by the method introduced in Section 4.4 from a picture of a known to be flat surface, while the cameras are already installed behind the windshield. The calibration images are shown in Figure 7.7.

The variables MSPTL, MSPTR, MSPHL, MSPHR and MPGR are calculated for road sections of 10 m length and are shown for a total driving distance of



(a) Transformed color image.

(b) Elevation map with resolution drawn as contour lines in mm px^{-1}.

Figure 8.7: Bird's eye view and elevation map at 160 m driving distance – the start of rutting is clearly visible.

500 m in Figure 8.6. It can be seen that the road condition abruptly worsens at a driving distance of ≈160 m. Figure 8.7b shows the elevation map and its resolution (refer to Section 5.2) of the segment where the condition changes. For comparison, the color image of the right camera is transformed into a Bird's-eye view in Figure 8.7a that shows the same segment as the elevation map. At a distance of approximately 8.5 m in the elevation map, the start of rutting is clearly visible, which interestingly is where the color of the asphalt changes. Presumably, one of both sides has been renewed at some point. It can also be seen that the lane width is 3 m and fits the recorded width.

8.3 Discussion

In this chapter, a real-life application of the stereo vision system in combination with the proposed stereo methods and the proposed stereo camera calibration method was presented. From the obtained point clouds, condition variables concerning the surface shape were calculated. In combination with the condi-

tion variables on surface defects, the overall condition index from Figure 8.2 could be calculated from camera images alone.

In [35], other methods of calculating the overall condition index are shown, which incorporate a roughness and a grip condition variable. The roughness variable is calculated using frequency responses, which in turn are calculated on continuous measurement data of the longitudinal road profile [36]. In [36], the longitudinal road profile of these segments is obtained from only four laser distance measurements mounted on a bar of 2 m length, which in turn is attached to a vehicle, by combining consecutive depth measurements. The same technique can be applied by combining the overlapping depth maps created by the proposed method. The only variable that cannot be measured with the stereo camera system is the surface's grip.

In the presented example application, the depth resolution shown in Figure 8.7 varies between $2\,\mathrm{mm\,px^{-1}}$ and $4\,\mathrm{mm\,px^{-1}}$. If this is not sufficient, a higher resolution can be achieved in two ways:

- The length covered in a single frame can be shortened so that the part with high resolution in the front can be used.
- A camera with a higher sensor resolution can be used, but that, of course, increases the computing workload.

9 Conclusion and Future Work

This thesis's main objective was to explore the suitability of a stereo camera system behind the windshield of a moving vehicle for measuring the 3D profile of road surfaces in the context of road condition monitoring. For this purpose, two intermediate objectives had to be achieved. First, a stereo camera system had to be designed and put into operation (objective 1. a). Second, stereo algorithms had to be developed to extract depth information from images captured with that system (objective 1. b). The results were compared to measurements obtained with industrial laser scanners. Since the extraction of depth from stereoscopic images requires a calibrated stereo camera system, this thesis's second objective was to explore how the stereo camera system could be self-calibrated.

Before answering the questions about the suitability of a stereo camera system and the self-calibration in Section 9.2, a summary is first given in Section 9.1. Finally, possible future work is discussed in Section 9.3.

9.1 Summary

The theory of depth reconstruction from stereo cameras was explained in Chapter 2. Based on the theory, in Chapter 3, two stereo methods were proposed for the special case of elevation estimation of low-textured, slanted planes. The proposed methods use the plane-sweep approach, as it solves the issues that arise in this context. The first stereo method was based on traditional methods that consist of comparing pixels by a similarity measure and an optimization scheme that considers possible mismatches. The second method utilizes a CNN, which solves the stereo correspondence problem in one single step. Thus, the intermediate objective 1. b was achieved.

In Chapter 5 basic thoughts about the stereo camera system were presented. The relationship between the system parameters, such as focal lengths of the lenses or camera orientation, and the achievable resolution in all three spatial directions was established. Furthermore, the influence of exposure time and aperture on the probability of correctly estimating the elevation of an object shown on individual pixels was identified. Based on these relations, it was shown how to determine optimal parameters for the stereo camera system.

Based on the considerations from Chapter 5, a stereo camera system was designed and put into operation in Chapter 6. Thus, the intermediate goal 1. a was achieved. With the stereo camera system, static and dynamic scenes were recorded, and measurements were performed with a terrestrial laser scanner and with a laser scanner mounted to the roof of a mobile mapping vehicle. The stereoscopic images were converted into elevation images with the developed stereo methods and further converted into point clouds. After a method for comparing the stereo vision point clouds with those generated by the laser scanners was developed, the results were evaluated. It was found that the measurements from stereo vision and laser scanners show a high degree of agreement.

What has been summarized so far assumed that the images were corrected for lens distortion because only then the projective camera model can be applied. Therefore, a mathematical model for lens distortion was described in Chapter 2. It contains calibration parameters, and the methods found in the literature for extracting these parameters from calibration images were also described. In Chapter 4, a novel method for camera self-calibration of mono cameras and stereo camera systems was presented. Although only three images were used to calibrate mono cameras, experiments with artificial cameras showed that the model's parameters were identified almost exactly. Experiments with real cameras showed reprojection errors similar to those observed with an established method. Thereby, the established method used a large number of images of planes with control points, whereas the proposed method does not require control points and requires only a few images. With the proposed self-calibration method, the stereo camera system behind the windshield was calibrated. If the ground plane is used for calibration, the difficulty arose that only one perspective of it can be captured. In order to be able to determine all parameters nevertheless, some additional information had to be provided.

In Chapter 8, the stereo camera system, in combination with the proposed stereo methods, was used to capture condition variables used in road construction. For this purpose, a broader lane had to be captured than in Chapter 6. Therefore, a new lens and accompanying parameters of the stereo camera system were found following Chapter 5. An inner-city street was captured with the system, and the captured stereoscopic images were converted into 3D point clouds using the proposed stereo methods. From the point clouds, the condition variables used in road construction were obtained.

9.2 Conclusions

The answer to the central research question from Section 1.2

1. *Is a stereo camera system that is mounted behind the windshield of a moving vehicle suitable for measuring the 3D profile of road surfaces in the context of road condition monitoring?*

depends on the required accuracy of the measurements and the stereo camera system used. Throughout this research, an image sensor with a resolution of $1920 \times 1200\,\mathrm{px}$ was used, limiting the elevation resolution. In Chapter 6 it was used to capture a lane width of 2 m, and resulted in a depth resolution of $\approx 0.75\,\mathrm{px\,mm^{-1}}$ near the cameras. It decreases with increasing distance between the depicted part of the road and the cameras. At a distance of 6 m from the foremost part, it has dropped to $\approx 0.4\,\mathrm{px\,mm^{-1}}$. If a 3 m lane is to be captured with the same sensor, it decreases to $\approx 0.5\,\mathrm{px\,mm^{-1}}$ and $\approx 0.25\,\mathrm{px\,mm^{-1}}$. In Chapter 6, experiments were carried out with a 2 m lane width and 7 m segment length. The approximate mean RMS error across the entire segment achieved by the stereo method was within 2 mm. With a higher sensor resolution, the same result should be achieved on a 3 m lane.

For road construction in Germany, requirements are given in terms of a comparative measurement of state variables, some of which were described in Chapter 8 [36]. For example, for state variables related to the cross-section of a road, averages are calculated on 100 m segments on the reference and the compared measurements. The mean difference between these averages must be within 1 mm. The reference measurements must be carried out by the Bundesanstalt für Straßenwesen (Federal Highway Research Institute). Such measurements were not available, but this requirement might be met with a sufficiently high sensor resolution.

A stereo camera system that is mounted behind the windshield of a moving vehicle is, therefore, in principle suitable for measuring the 3D profile of the surface of roads in the context of road condition monitoring. However, whether the requirements of a road construction authority can be met cannot be answered in general terms.

The answer to the second research question from Section 1.2

2. *Is it possible to automatically calibrate the stereo camera system installed in a vehicle behind the windshield with a sufficiently high accuracy for road condition monitoring?*

depends on the required accuracy of the final depth measurements and whether additional information can be provided. One difficulty in applying the calibration of cameras mounted behind the windshield is that only one perspective of

the ground plane can be captured for the calibration. Therefore, not all calibration parameters can be estimated automatically, and some prior knowledge is required. That can be the focal length and a distance in the images, or it can be two distances of the stereo camera setup. The accuracy of the results depends on the accuracy of the additional information or measurements. However, as shown in Chapter 8, qualitatively good results can be obtained with such a measurement.

Besides answering the research questions, another result and probably the most innovative part of this thesis is the novel self-calibration method that can be used for mono cameras and stereo camera systems. It has proven to provide high-quality results with minimal effort.

9.3 Future work

9.3.1 3D surface profile reconstruction

The current implementations of the proposed stereo methods do not achieve real-time performance, which means that they cannot process the data as fast as it occurs during the drive. The current implementation of the SGM optimization algorithm has complexity $\mathcal{O}(WHP^2)$, whereas the original SGM algorithm has complexity $\mathcal{O}(WHP)$, which is due to a different smoothing term (refer to Section 2.5.6). While the current implementation is highly optimized, as it runs in parallel on all 48 cores of a CPU and makes use of the Intel AVX2 SIMD instruction set, it is also possible to implement SGM on GPUs [51]. In [54] experiments showed that the utilized GPUs were faster than the CPUs. Therefore, a future work could evaluate the simpler smoothing term and utilize a GPU implementation for optimization.

It would also be interesting to try a different optimization algorithm. The variational approach is applicable in the presented case. It is also reported to be fast and is parallelizable on GPUs. Furthermore, it does not suffer from a grid bias, and it finds the global optimum.

Future work on the CNN method for 3D surface profile reconstruction could focus on accelerating its execution time by reducing the feature map network size. It was originally designed for the reconstruction of general scenes. Since the road surfaces have little texture, the extraction of rich features may be unnecessary, and a network with fewer layers may be sufficient.

Until now, the stereoscopic images were processed one by one, but for road condition monitoring, continuous measurement data is needed. This may be obtained by merging 3D point clouds extracted from a series of partially

overlapping images. At the same time, the overall accuracy could be improved by a higher weighting of points estimated from parts close to the cameras.

9.3.2 Camera self-calibration

In this work only up to three images were processed for camera self-calibration, but the proposed method itself is not limited in the number of images. The limiting factor is the amount of GPU memory. Future work could focus on reducing the amount of memory needed, e. g. by reducing the size of the feature extraction network. Another approach is changing the order of computations from undistorting and transforming the images first and then extracting the feature maps to first extracting the feature maps and then undistorting and transforming the feature maps. That is the approach that was used in the depth reconstruction network. That way, more calibration images could be used, but the transformation of the images would be performed at a reduced resolution, and some precision might be lost. In general, the advantage of using more pictures would be a greater precision of the extracted parameters and the estimation of the principal point independently from the estimation of the distortion model. It might even be possible to separate the location of the principal point and the center of distortion.

Another point to be examined in the self-calibration approach is the problem of local minima. Although, with a descent initialization, this was not a problem in the experiments, there is no guarantee that the optimization by gradient descent finds the global minimum or even a "good" minimum. However, by visually examining the optimization results, it is possible to determine whether the optimization has found a minimum so that the undistorted and transformed images blend well.

The assumption for the self-calibration method to work are images of perfectly flat planes. Since the planes used cannot meet this requirement, the influence of non-flat planes and the effect of the surface's shape on the calibration result should also be investigated.

Bibliography

[1] URL: vision.middlebury.edu/stereo/.

[2] Basler AG. *Basler ace - User's manual for USB 3.0 Cameras*. Jan. 12, 2017.

[3] Basler AG. *Global Shutter, Rolling Shutter - Functionality and Characteristics of Two Exposure Methods (Shutter Variants)*. May 14, 2018. URL: https://www.baslerweb.com/de/vertrieb-support/downloads/downloads-dokumente/global-shutter-rolling-shutter/ (visited on 06/17/2020).

[4] Basler AG. *What is sensitivity and why are sensitivity statements often misleading?* Sept. 19, 2019. URL: https://www.baslerweb.com/en/sales-support/knowledge-base/frequently-asked-questions/what-is-sensitivity-and-why-are-sensitivity-statements-often-misleading/14987/ (visited on 09/19/2019).

[5] A. Ansar, A. Castano, and L. Matthies. "Enhanced real-time stereo using bilateral filtering". In: *Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission*. Sept. 2004, pp. 455–462. DOI: 10.1109/TDPVT.2004.1335273.

[6] P. J. Besl and N. D. McKay. "A method for registration of 3-D shapes". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (Feb. 1992), pp. 239–256. ISSN: 0162-8828. DOI: 10.1109/34.121791.

[7] Christopher M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. New York, NY: Springer, 2006. 738 pp. ISBN: 978-0-387-31073-2.

[8] Michael J. Black and Anand Rangarajan. "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision". *International Journal of Computer Vision* 19.1 (July 1996), pp. 57–91. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/BF00131148.

[9] Jean-Yves Bouguet. *Camera Calibration Toolbox for Matlab*. 2015. URL: http://www.vision.caltech.edu/bouguetj/calib_doc/ (visited on 05/14/2019).

[10] Yuri Boykov, Olga Veksler, and Ramin Zabih. "Fast approximate energy minimization via graph cuts". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.11 (2001), pp. 1222–1239.

[11] G. Bradski. "The OpenCV Library". *Dr. Dobb's Journal of Software Tools* (2000).

[12] Gary Bradski and Adrian Kaehler. *Learning OpenCV*. 1st ed. Sebastopol, CA: O'Reilly, 2008. 555 pp. ISBN: 978-0-596-51613-0.

[13] Duane C. Brown. "Close-range camera calibration". *Photogrammetric Engineering* 37.8 (1971), pp. 855–866.

[14] Tobias Brücker. "Stadt lässt Leverkusens Straßen scannen". *rp-online* (Aug. 29, 2017).

[15] Hauke Brunken and Clemens Gühmann. "Deep learning self-calibration from planes". In: *Twelfth International Conference on Machine Vision (ICMV 2019)*. Vol. 11433. 2020, p. 114333L.

[16] Hauke Brunken and Clemens Gühmann. "Incorporating Plane-Sweep in Convolutional Neural Network Stereo Imaging for Road Surface Reconstruction". In: *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP*. 2019, pp. 784–791. ISBN: 978-989-758-354-4. DOI: 10.5220/0007352107840791.

[17] Hauke Brunken and Clemens Gühmann. "Pavement distress detection by stereo vision/Straßenzustandserkennung durch stereoskopische Bildverarbeitung". *tm-Technisches Messen* 86 (s1 2019), pp. 42–46.

[18] Hauke Brunken and Clemens Gühmann. "Road Surface Reconstruction by Stereo Vision". *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 88.6 (Dec. 1, 2020), pp. 433–448. ISSN: 2512-2819. DOI: 10.1007/s41064-020-00130-z.

[19] Dimitri Bulatov. "Temporal Selection of Images for a Fast Algorithm for Depth-map Extraction in Multi-baseline Configurations". In: *Proceedings of the 10th International Conference on Computer Vision Theory and Applications*. Berlin, Germany, 2015, pp. 395–402. ISBN: 978-989-758-091-8. DOI: 10.5220/0005239503950402.

[20] Bundesanstalt für Straßenwesen. *Measuring vehicles and systems*. 2018. URL: https://www.bast.de/durabast/DE/durabast/Messfahrzeuge/fahrzeuge_node.html (visited on 10/11/2018).

[21] Ulrich Canzler and Benjamin Winkler. *Weiterentwicklung der automatisierten Merkmalserkennung im Rahmen des TP3*. 67. Bremerhaven, Germany: Bundesanstalt für Straßenwesen, 2012.

[22] A. Cefalu, N. Haala, and D. Fritsch. "Structureless bundle adjustment with self-calibration using accumulated constraints". In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. III-3. June 2, 2016, pp. 3–9. DOI: 10.5194/isprsannals-III-3-3-2016.

[23] J. Chang and Y. Chen. "Pyramid Stereo Matching Network". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5410–5418.

[24] Tom B. J. Coenen and Amir Golroo. "A review on automated pavement distress detection methods". *Cogent Engineering* 4.1 (2017), p. 1374822. DOI: 10.1080/23311916.2017.1374822.

[25] Robert T. Collins. "A space-sweep approach to true multi-image matching". In: *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1996, pp. 358–363.

[26] Blender Online Community. *Blender - A 3D modelling and rendering package*. Version 2.78. Stichting Blender Foundation, Amsterdam: Blender Foundation, 2017. URL: http://www.blender.org.

[27] Ingemar J. Cox et al. "A Maximum Likelihood Stereo Algorithm". *Computer Vision and Image Understanding* 63.3 (May 1996), pp. 542–567. ISSN: 10773142. DOI: 10.1006/cviu.1996.0040.

[28] F. Devernay and O. D. Faugeras. "Computing differential properties of 3-D shapes from stereoscopic images without 3-D models". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*. Seattle, WA, 1994, pp. 208–213. ISBN: 978-0-8186-5825-9. DOI: 10.1109/CVPR.1994.323831.

[29] *DIN EN 13036-8:2008-06, Oberflächeneigenschaften von Straßen und Flugplätzen - Prüfverfahren - Teil 8: Bestimmung der Parameter zur Ermittlung der Breitenunebenheit; Deutsche Fassung EN 13036-8:2008*. Berlin, Germany: Beuth Verlag GmbH, 2008.

[30] Alexey Dosovitskiy et al. "Flownet: Learning optical flow with convolutional networks". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 2758–2766.

[31] Markus Eisenbach et al. "How to get pavement distress detection ready for deep learning? A systematic approach". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. Anchorage, AK, May 2017, pp. 2039–2047. ISBN: 978-1-5090-6182-2. DOI: 10.1109/IJCNN.2017.7966101.

[32] Rui Fan, Xiao Ai, and Naim Dahnoun. "Road surface 3d reconstruction based on dense subpixel disparity map estimation". *IEEE Transactions on Image Processing* 27.6 (June 2018), pp. 3025–3035. ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2018.2808770. arXiv: 1807.01874.

[33] Olivier Faugeras, Quang-Tuan Luong, and Théo Papadopoulo. *The geometry of multiple images: the laws that govern the formation of multiple images of a scene and some of their applications*. Cambridge, MA: MIT Press, 2001. 644 pp. ISBN: 978-0-262-06220-6.

[34] FGSV. *Arbeitspapiere zur Systematik der Straßenerhaltung - Reihe K: Kommunale Straßen - Unterabschnitt K 2.3: Schadenskatalog für die messtechnische und visuelle Zustandserfassung*. 490 AP 9 K 2.3. Köln, Germany: FGSV Verlag, 2015.

[35] FGSV. *Empfehlungen für das Erhaltungsmanagement von Innerortsstraßen - E EMI 2012*. 487. Köln, Germany: FGSV Verlag, 2012.

[36] FGSV. *Technische Prüfvorschriften für Ebenheitsmessungen auf Fahrbahnoberflächen in Längs- und Querrichtung - Teil: Berührungslose Messungen*. 404/2. Köln, Germany: FGSV Verlag, 2009.

[37] FGSV. *Zusätzliche Technische Vertragsbedingungen und Richtlinien zur Zustandserfassung und -bewertung von Straßen - ZTV ZEB-StB 06*. 489. Köln, Germany: FGSV Verlag, 2006.

[38] Martin A Fischler and Robert C Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". *Communications of the ACM* 24.6 (1981), pp. 381–395.

[39] Clive S. Fraser. "Automatic Camera Calibration in Close Range Photogrammetry". *Photogrammetric Engineering & Remote Sensing* 79.4 (Apr. 1, 2013), pp. 381–388. ISSN: 00991112. DOI: 10.14358/PERS.79.4.381.

[40] Yasutaka Furukawa and Jean Ponce. "Accurate Camera Calibration from Multi-View Stereo and Bundle Adjustment". *International Journal of Computer Vision* 84.3 (Sept. 2009), pp. 257–268. ISSN: 1573-1405. DOI: 10.1007/s11263-009-0232-2.

[41] David Gallup et al. "Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. 2007, pp. 1–8.

[42] Gang Li and Steven W. Zucker. "Surface Geometric Constraints for Stereo in Belief Propagation". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*. New York, NY, 2006, pp. 2355–2362. ISBN: 978-0-7695-2597-6. DOI: 10.1109/CVPR.2006.299.

[43] 3D Mapping Solutions GmbH. *Die Vermessungstechnik*. URL: https://www.3d-mapping.de/ueber-uns/unternehmensbereiche/data-acquisition/unser-vermessungssystem/ (visited on 12/01/2019).

[44] GPL software. *CloudCompare 2.8*. Version 2.8. 2017. URL: http://www.cloudcompare.org.

[45] Gunnar Gräfe. "Kinematische Anwendungen von Laserscannern im Straßenraum". Dissertation. Neubiberg, Germany: Universität der Bundeswehr München, 2007.

[46] Heinz Haferkorn. *Optik : Physikalisch-technische Grundlagen und Anwendungen*. Weinheim, Germany: Wiley, 2003. ISBN: 978-3-527-40372-1.

[47] A. Hanel, L. Hoegner, and U. Stilla. "Towards the Influence of a Car Windschield on Depth Calculation with a Stereo Camera System". *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLI-B5 (June 15, 2016), pp. 461–468. ISSN: 2194-9034. DOI: 10.5194/isprsarchives-XLI-B5-461-2016.

[48] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge, England; New York, NY: Cambridge University Press, 2003. ISBN: 978-0-511-18618-9.

[49] K. He et al. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.9 (2015), pp. 1904–1916.

[50] Eugene Hecht. *Optics*. 4th ed. San Francisco, CA: Addison Wesley, 2002. 698 pp. ISBN: 0-321-18878-0.

[51] D. Hernandez-Juarez et al. "Embedded Real-time Stereo Estimation via Semi-global Matching on the GPU". In: *Procedia Computer Science*. Vol. 80. 2016, pp. 143–153. DOI: 10.1016/j.procs.2016.05.305.

[52] Daniel Herrera, C. Juho Kannala, and Janne Heikkila. "Forget the checkerboard: Practical self-calibration using a planar scene". In: *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Lake Placid, NY, Mar. 2016, pp. 1–9. ISBN: 978-1-5090-0641-0. DOI: 10.1109/WACV.2016.7477641.

[53] H. Hirschmüller. "Stereo Processing by Semiglobal Matching and Mutual Information". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 (Feb. 2008), pp. 328–341. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2007.1166.

[54] H. Hirschmüller, M. Buder, and I. Ernst. "Memory Efficient Semi-Global Matching". *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* I-3 (July 23, 2012), pp. 371–376. ISSN: 2194-9050. DOI: 10.5194/isprsannals-I-3-371-2012.

[55] H. Hirschmüller and D. Scharstein. "Evaluation of Stereo Matching Costs on Images with Radiometric Differences". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.9 (Sept. 2009), pp. 1582–1599. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2008.221.

[56] Katsushi Ikeuchi, ed. *Computer Vision - A Reference Guide*. Boston, MA: Springer, 2014. ISBN: 978-0-387-30771-8 978-0-387-31439-6. DOI: 10.1007/978-0-387-31439-6.

[57] Intel Corporation. "Desktop 4th Generation Intel® Core™ Processor Family, Desktop Intel® Pentium® Processor Family, and Desktop Intel® Celeron® Processor Family - Datasheet – Volume 1 of 2". 1 (2015), p. 125.

[58] A. Irschara et al. "Efficient and Globally Optimal Multi View Dense Matching for Aerial Images". *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* I-3 (July 20, 2012), pp. 227–232. ISSN: 2194-9050. DOI: 10.5194/isprsannals-I-3-227-2012.

[59] Volodymyr Ivanchenko, Huiying Shen, and J. Coughlan. "Elevation-based MRF stereo implemented in real-time on a GPU". In: *2009 Workshop on Applications of Computer Vision (WACV)*. 2009, pp. 1–8.

[60] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. "Spatial transformer networks". In: *Advances in neural information processing systems*. 2015, pp. 2017–2025.

[61] Bernd Jähne. *Practical handbook on image processing for scientific and technical applications*. 2nd ed. Boca Raton, FLA: CRC Press, 2004. 610 pp. ISBN: 0-8493-1900-5.

[62] Jorg H. Kappes et al. "A Comparative Study of Modern Inference Techniques for Discrete Energy Minimization Problems". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. Portland, OR, June 2013, pp. 1328–1335. ISBN: 978-0-7695-4989-7. DOI: 10.1109/CVPR.2013.175.

[63] Alex Kendall et al. "End-to-End Learning of Geometry and Context for Deep Stereo Regression". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 66–75. arXiv: 1703.04309.

[64] Junhwan Kim, Vladimir Kolmogorov, and Ramin Zabih. "Visual correspondence using energy minimization and mutual information". In: *Proceedings Ninth IEEE International Conference on Computer Vision*. Vol. 2. 2003, pp. 1033–1040. ISBN: 0-7695-1950-4.

[65] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". *arXiv:1412.6980 [cs]* (Jan. 29, 2017). arXiv: 1412.6980.

[66] Granino A. Korn and Theresa M. Korn. *Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review*. Dover Books on Mathematics. Mineola, NY: Dover Publications, 2000. ISBN: 978-0-486-32023-6.

[67] Hongdong Li and Richard Hartley. "Plane-Based Calibration and Auto-calibration of a Fish-Eye Camera". In: *Computer Vision – ACCV 2006*. Vol. 3851. 2006, pp. 21–30. ISBN: 978-3-540-31219-2 978-3-540-32433-1. DOI: 10.1007/11612032_3.

[68] L. Li et al. "PMSC: PatchMatch-Based Superpixel Cut for Accurate Stereo Matching". *IEEE Transactions on Circuits and Systems for Video Technology* 28.3 (2018), pp. 679–692.

[69] D.G. Lowe. "Object recognition from local scale-invariant features". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Kerkyra, Greece, 1999, 1150–1157 vol.2. ISBN: 978-0-7695-0164-2. DOI: 10.1109/ICCV.1999.790410.

[70] David J. C. MacKay. *Information theory, inference and learning algorithms*. 3rd ed. Cambridge, England: Cambridge University Press, 2003.

[71] Ezio Malis and Manuel Vargas. *Deeper understanding of the homography decomposition for vision-based control*. RR-6303. inria-00174036v3. INRIA, 2007, p. 90.

[72] Christopher Manning, Prabhakar Raghavan, and Hinrich Schuetze. *Introduction to Information Retrieval*. Cambridge, England: Cambridge University Press, 2009. 581 pp.

[73] Nikolaus Mayer et al. "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4040–4048.

[74] Moritz Menze, Christian Heipke, and Andreas Geiger. "Joint 3D Estimation of Vehicles and Scene Flow". In: *ISPRS Workshop on Image Sequence Analysis (ISA)*. 2015.

[75] Joint Committee for Guides in Metrology. *JCGM 100:2008 Evaluation of measurement data - Guide to the expression of uncertainty in measurement*. JCGM, 2008.

[76] *MnDOT Pavement Preservation Manual*. MN: Minnesota Department of Transportation, 2020.

[77] Kevin P. Murphy. *Machine learning: a probabilistic perspective*. Cambridge, MA; London, England: MIT Press, 2012. 1067 pp. ISBN: 978-0-262-01802-9.

[78] Alejandro Newell, Kaiyu Yang, and Jia Deng. "Stacked Hourglass Networks for Human Pose Estimation". In: *European conference on computer vision*. 2016, pp. 483–499. arXiv: 1603.06937.

[79] C. Olsson, J. Ulén, and Y. Boykov. "In Defense of 3D-Label Stereo". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. June 2013, pp. 1730–1737. DOI: 10.1109/CVPR.2013.226.

[80] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035.

[81] Lutz Pinkofsky and Dirk Jansen. "Structural pavement assessment in Germany". *Frontiers of Structural and Civil Engineering* 12.2 (June 2018), pp. 183–191. ISSN: 2095-2430, 2095-2449. DOI: 10.1007/s11709-017-0412-z.

[82] Thomas Pock et al. "A Convex Formulation of Continuous Multi-label Problems". In: *Computer Vision – ECCV 2008*. Vol. 5304. Series Title: Lecture Notes in Computer Science. 2008, pp. 792–805. ISBN: 978-3-540-88690-7. DOI: 10.1007/978-3-540-88690-7_59.

[83] Marc Pollefeys et al. "Visual Modeling with a Hand-Held Camera". *International Journal of Computer Vision* 59.3 (Sept. 2004), pp. 207–232. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000025798.50602.3a.

[84] Stefania C. Radopoulou and Ioannis Brilakis. "Patch detection for pavement assessment". *Automation in Construction* 53 (May 2015), pp. 95–104. ISSN: 09265805. DOI: 10.1016/j.autcon.2015.03.010.

[85] Antonella Ragnoli, Maria De Blasiis, and Alessandro Di Benedetto. "Pavement Distress Detection Methods: A Review". *Infrastructures* 3.4 (Dec. 19, 2018), p. 58. ISSN: 2412-3811. DOI: 10.3390/infrastructures3040058.

[86] L. Roth and H. Mayer. "Reduction of the Fronto-Parallel Bias for Wide-Baseline Semi-Global Matching". *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* IV-2/W5 (May 29, 2019), pp. 69–76. ISSN: 2194-9050. DOI: 10.5194/isprs-annals-IV-2-W5-69-2019.

[87] I Scazziga. *Evaluation des Strassenzustandes*. 12/99. Eidgenössisches Departement für Umwelt, Verkehr, Energie und Kommunikation / Bundesamt für Strassen, 2000.

[88] D. Scharstein, R. Szeliski, and R. Zabih. "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms". In: *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*. Kauai, HI, 2001, pp. 131–140. ISBN: 978-0-7695-1327-0. DOI: 10.1109/SMBV.2001.988771.

[89] Daniel Scharstein, Tatsunori Taniai, and Sudipta N. Sinha. "Semi-global Stereo Matching with Surface Orientation Priors". In: *2017 International Conference on 3D Vision (3DV)*. Qingdao, China, Oct. 2017, pp. 215–224. ISBN: 978-1-5386-2610-8. DOI: 10.1109/3DV.2017.00033.

[90] S. N. Sinha, D. Scharstein, and R. Szeliski. "Efficient High-Resolution Stereo Matching Using Local Plane Sweeps". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. June 2014, pp. 1582–1589. DOI: 10.1109/CVPR.2014.205.

[91] Nikolai Smolyanskiy, Alexey Kamenev, and Stan Birchfield. "On the Importance of Stereo for Accurate Depth Estimation: An Efficient Semi-Supervised Deep Neural Network Approach". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 1007–1015.

[92] Noah Snavely, Steven M. Seitz, and Richard Szeliski. "Modeling the World from Internet Photo Collections". *International Journal of Computer Vision* 80.2 (Nov. 2008), pp. 189–210. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-007-0107-3.

[93] Bernhard Steinauer, Andreas Ueckermann, and Günther Maerschalk. *Analyse vorliegender messtechnischer Zustandsdaten und Erweiterung der Bewertungsparameter für Innerortsstraßen*. 46. Bremerhaven: Bundesanstalt für Straßenwesen, 2006.

[94] R. Szeliski et al. "A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.6 (June 2008), pp. 1068–1080. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2007.70844.

[95] Richard Szeliski. "Bayesian modeling of uncertainty in low-level vision". *International Journal of Computer Vision* 5.3 (1990), pp. 271–301.

[96] Richard Szeliski. *Computer Vision*. Texts in Computer Science. London: Springer London, 2011. ISBN: 978-1-84882-934-3 978-1-84882-935-0.

[97] Jean-Philippe Tardif, Peter Sturm, and Sebastien Roy. "Plane-based self-calibration of radial distortion". In: *2007 IEEE 11th International Conference on Computer Vision*. Rio de Janeiro, Brazil, 2007, pp. 1–8. ISBN: 978-1-4244-1630-1. DOI: 10.1109/ICCV.2007.4409113.

[98] The FFmpeg developers. *ffmpeg*. Version 3.4.1. 2017. URL: http://www.ffmpeg.org.

[99] The MathWorks. *MATLAB Computer Vision Toolbox*. Version 2019a. 2019.

[100] Bill Triggs. "Autocalibration from planar scenes". In: *Computer Vision — ECCV'98*. Vol. 1406. 1998, pp. 89–105. ISBN: 978-3-540-69354-3. DOI: 10.1007/BFb0055661.

[101] Bill Triggs et al. "Bundle Adjustment — A Modern Synthesis". In: *Vision Algorithms: Theory and Practice*. IWVA. Series Title: Lecture Notes in Computer Science. 2000, pp. 298–372. ISBN: 978-3-540-67973-8 978-3-540-44480-0. DOI: 10.1007/3-540-44480-7_21.

[102] Yi-Chang (James) Tsai and Anirban Chatterjee. "Pothole Detection and Classification Using 3D Technology and Watershed Method". *Journal of Computing in Civil Engineering* 32.2 (Mar. 2018), p. 04017078. ISSN: 0887-3801, 1943-5487. DOI: 10.1061/(ASCE)CP.1943-5487.0000726.

[103] Kerstin Vogel. "Iris und Stier auf den Straßen". *Sueddeutsche Zeitung* (Oct. 8, 2012).

[104] M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky. "MAP Estimation Via Agreement on Trees: Message-Passing and Linear Programming". *IEEE Transactions on Information Theory* 51.11 (Nov. 2005), pp. 3697–3717. ISSN: 0018-9448. DOI: 10.1109/TIT.2005.856938.

[105] Aloysius Wehr. "LIDAR: Airborne and terrestrial sensors". In: *Advances in Photogrammetry, Remote Sensing and Spatial Information Sciences: 2008 ISPRS Congress Book*. 2008.

[106] O. Woodford et al. "Global Stereo Reconstruction under Second-Order Smoothness Priors". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.12 (Dec. 2009), pp. 2115–2128. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2009.131.

[107] Ruigang Yang, Greg Welch, and Gary Bishop. "Real-time consensus-based scene reconstruction using commodity graphics hardware". In: *10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings.* Vol. 22. 2002, pp. 225–234.

[108] Ramin Zabih and John Woodfill. "Non-parametric local transforms for computing visual correspondence". In: *European conference on computer vision*. 1994, pp. 151–158. ISBN: 978-3-540-48400-4.

[109] Jure Zbontar and Yann LeCun. "Stereo matching by training a convolutional neural network to compare image patches". *Journal of Machine Learning Research* 17.1 (2016), p. 2.

[110] Z. Zhang. "A flexible new technique for camera calibration". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11 (Nov. 2000), pp. 1330–1334. ISSN: 01628828. DOI: 10.1109/34.888718.

[111] Huijie Zhao et al. "Calibration for stereo vision system based on phase matching and bundle adjustment algorithm". *Optics and Lasers in Engineering* 68 (May 2015), pp. 203–213. ISSN: 01438166. DOI: 10.1016/j.optlaseng.2014.12.001.

[112] Wenhao Zhao, Li Yan, and Yunsheng Zhang. "Geometric-constrained multi-view image matching method based on semi-global optimization". *Geo-spatial Information Science* 21.2 (Apr. 3, 2018), pp. 115–126. ISSN: 1009-5020, 1993-5153. DOI: 10.1080/10095020.2018.1441754.

Advances in Automation Engineering
Hrsg.: Prof. Dr.-Ing. Clemens Gühmann
ISSN 2509-8950 (print)
ISSN 2509-8969 (online)

---

1. Nowoisky, Sebastian: Verfahren zur Identifikation nichtlinearer dynamischer Getriebemodelle. - 2016. - VIII, 224 S.
ISBN 978-3-7983-2854-9 (print) 15,00 EUR
ISBN 978-3-7983-2855-6 (online)
DOI 10.14279/depositonce-5420

2. Huang, Hua: Model-based calibration of automated transmissions. - 2016. - XXIV, 134 S.
ISBN 978-3-7983-2858-7 (print) 14,00 EUR
ISBN 978-3-7983-2859-4 (online)
DOI 10.14279/depositonce-5461

3. Röper, Jan: Entwicklung eines virtuellen Getriebeprüfstands. - 2017. - xxvi, 133 S.
ISBN 978-3-7983-2951-5 (print) 14,00 EUR
ISBN 978-3-7983-2952-2 (online)
DOI 10.14279/depositonce-6073

4. Funck, Jürgen Helmut: Synchronous data acquisition with wireless sensor networks. - 2018. - xix, 327 S.
ISBN 978-3-7983-2980-5 (print) 19,50 EUR
ISBN 978-3-7983-2981-2 (online)
DOI 10.14279/depositonce-6716

5. Kiffe, Axel: Echtzeitsimulation leistungselektronischer Schaltungen für die Hardware-in-the-Loop-Simulation. - 2018. - x, 212 S.
ISBN 978-3-7983-3013-9 (print) 14,00 EUR
ISBN 978-3-7983-3014-6 (online)
DOI 10.14279/depositonce-7227

6. Lück, Rudolf: Überwachung hybrider Schrägkugellager in Luftfahrttriebwerken. - 2018. - XXIII, 169 S.
ISBN 978-3-7983-3021-4 (print) 18,50 EUR
ISBN 978-3-7983-3022-1 (online)
DOI 10.14279/depositonce-7283

7. Mokhtari, Noushin: Überwachung hydrodynamischer Gleitlager basierend auf der Körperschallanalyse. - 2020. - XXI, 167 S.
ISBN 978-3-7983-3183-9 (print) 18,50 EUR
ISBN 978-3-7983-3184-6 (online)
DOI 10.14279/depositonce-10642

8. Strommenger, Daniel: Verschleißprognose zur zuverlässigkeitsorientierten Regelung für trockene Reibkupplungen. - 2021. - xii, 236 S.
ISBN 978-3-7983-3196-9 (print) 19,50 EUR
ISBN 978-3-7983-3197-6 (online)
DOI 10.14279/depositonce-11299

Universitätsverlag der TU Berlin

ELEKTRONISCHE MESS- UND DIAGNOSETECHNIK

Stereo Vision-Based Road Condition Monitoring

When planning road construction measures, it is essential to have up-to-date information on road conditions. If this information is not to be obtained manually, it is currently obtained using laser scanners mounted on mobile mapping vehicles. In this thesis, the application of a stereo camera system, which is mounted behind the windshield of a vehicle, is investigated as an alternative.
For this purpose, a method based on plane-sweeping in combination with semi-global matching for the stereoscopic reconstruction of surfaces with little and repetitive textures is proposed. Furthermore, it is shown how the plane-sweep approach can be implemented in a neural network, which solves the stereo correspondence problem in a single step. Since cameras used in this context must be calibrated, a completely new approach for the self-calibration of mono cameras and stereo camera systems is introduced. It uses feature maps instead of feature points to compare multiple views of one and the same plane and employs backpropagation with gradient descent to infer unknown calibration parameters.

ISBN 978-3-7983-3205-8 (print)
ISBN 978-3-7983-3206-5 (online)

ISBN 978-3-7983-3205-8

https://verlag.tu-berlin.de