DGK Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 825

Niclas Zeller

Direct Plenoptic Odometry – Robust Tracking and Mapping with a Light Field Camera

München 2018

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5237-6

Diese Arbeit ist gleichzeitig als E-Dissertation auf mediaTUM veröffentlicht:
https://mediatum.ub.tum.de/?id=1402349

Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 825

Direct Plenoptic Odometry –
Robust Tracking and Mapping with a Light Field Camera

Von der Ingenieurfakultät Bau Geo Umwelt

der Technischen Universität München

zur Erlangung des Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

Vorgelegt von

M. Eng. Niclas Zeller

Geboren am 07.06.1987 in Stuttgart

München 2018

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5237-6

Diese Arbeit ist gleichzeitig als E-Dissertation auf mediaTUM veröffentlicht: https://mediatum.ub.tum.de/?id=1402349

Adresse der DGK:

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften (DGK)
Alfons-Goppel-Straße 11 ● D – 80 539 München
Telefon +49 – 331 – 288 1685 ● Telefax +49 – 331 – 288 1759
E-Mail post@dgk.badw.de ● http://www.dgk.badw.de

Prüfungskommission:

Vorsitzender: Prof. Dr. rer. nat. Thomas Kolbe

Referent: Prof. Dr.-Ing. Uwe Stilla

Korreferenten: Prof. Dr. rer. nat. Daniel Cremers
Prof. Dr.-Ing. Franz Quint (Hochschule Karlsruhe)

Tag der mündlichen Prüfung: 25.10.2018

© 2018 Bayerische Akademie der Wissenschaften, München

Alle Rechte vorbehalten. Ohne Genehmigung der Herausgeber ist es auch nicht gestattet,
die Veröffentlichung oder Teile daraus auf photomechanischem Wege (Photokopie, Mikrokopie) zu vervielfältigen

ISSN 0065-5325 ISBN 978-3-7696-5237-6

Abstract

Research and industry proceed to build autonomous driving cars, self-navigating unmanned aerial vehicles and intelligent, mobile robots. Furthermore, due to the demographic change, intelligent assistance devices for visually impaired and elderly people are in demand. For these tasks systems are needed which reliably map the 3D surroundings and are able to self-localize in these created maps. Such systems or methods can be summarized under the terms simultaneous localization and mapping (SLAM) or visual odometry (VO). The SLAM problem was broadly studied for different visual sensors like monocular, stereo and RGB-D cameras. During the last decade plenoptic cameras (or light field cameras) have become available as commercial products. They capture 4D light field information in a single image. This light field information can be used, for instance, to retrieve 3D structure.

This dissertation deals with the task of VO for plenoptic cameras. For this purpose, a new model for micro lens array (MLA) based light field cameras was developed. On the basis of this model a multiple view geometry (MVG) for MLA based light field cameras is derived.

An efficient probabilistic depth estimation approach for MLA based light field cameras is presented. The method establishes semi-dense depth maps directly from the micro images recorded by the camera. Multiple depth observations are merged in a probabilistic depth hypotheses. Disparity uncertainties resulting e.g. from differently focused micro images and sensor noise are taken into account. This algorithm is initially developed on the basis of single light field images and later extended by the introduced MVG.

Furthermore, calibration approaches for focused plenoptic cameras at different levels of complexity are presented. We begin with depth conversion functions, which convert the virtual depth estimated by the camera into metric distances, and then proceed to derive a plenoptic camera model on the basis of the estimated virtual depth map and the corresponding synthesized, totally focused image. This model takes into consideration depth distortion and a sensor which is tilted in relation to the main lens. Finally it leads to a plenoptic camera model which defines the complete projection of an object point to multiple micro images on the sensor. Based on this model we emphasize the importance of modeling squinting micro lenses. The depth conversion functions are estimated based on a series of range measurements while the parameters of all other models are estimated in a bundle adjustment using a 3D calibration target. The bundle adjustment based methods significantly outperform existing approaches.

The plenoptic camera based MVG, the depth estimation approach, and the model obtained from the calibration are combined in a VO algorithm called Direct Plenoptic Odometry (DPO). DPO works directly on the recorded micro images. Therefore, it does not have to deal with aliasing effects in the spatial domain. The algorithm generates semi-dense 3D point clouds on the basis of correspondences in subsequent light field frames. A scale optimization framework is used to adjust scale drifts and wrong absolute scale initializations. To the best of our knowledge, it is the first method that performs tracking and mapping for plenoptic cameras directly on the micro images. DPO is tested on a variety of indoor and outdoor sequences. With respect to the scale drift, it outperforms state-of-the-art monocular VO and SLAM algorithms. Regarding absolute accuracy, DPO is competitive to existing monocular and stereo approaches.

Kurzfassung

Sowohl in der Forschung als auch der Industrie werden autonome Fahrzeuge, selbstfliegende Drohnen und intelligente, mobile Roboter entwickelt. Außerdem wird aufgrund des demographischen Wandels die Nachfrage nach intelligenten Hilfsmitteln für sehbeeinträchtigte und ältere Menschen immer größer. Für die beschriebenen Anwendungen werden Systeme benötigt, welche zuverlässig die 3D Umgebung erfassen und sich selbst in dieser Umgebung lokalisieren können. Diese Systeme können unter den Begriffen Simultaneous Localization and Mapping (SLAM) oder visuelle Odometrie (VO) zusammengefasst werden. Bisher wurden SLAM-Methoden hauptsächlich basierend auf monokularen, Stereo- und RGB-D Kameras untersucht. In den letzten Jahren kamen plenoptische Kameras auf den Markt. Diese sind in der Lage, anhand des aufgenommenen 4D Lichtfelds, z.B. 3D Information zu erlangen.

In dieser Arbeit werden VO-Methoden basierend auf einer plenoptische Kamera untersucht. Dafür wurde ein neues Modell sowie eine Multi-View-Geometrie (MVG) für Lichtfeldkameras mit Mikrolinsengittern (engl.: micro lens arrays, MLA) hergeleitet.

Es wurde eine Methode zur Tiefenschätzung für MLA basierte Lichtfeldkameras entwickelt. Diese Methode berechnet, direkt aus den Mikrolinsenbildern der Kamera, eine teildichte Tiefenkarte. Mehrere Tiefenbeobachtungen werden hierbei in einem stochastischen Modell vereint. Das Modell berücksichtigt Disparitätsunsicherheiten welche aus z.B. unterschiedlich fokussierten Mikrolinsenbildern und Sensorrauschen resultieren.

Weiterhin werden verschiedene Kalibiermethoden für fokussierte plenoptische Kameras vorgestellt. Zunächst werden Funktionen zur Umrechnung der geschätzten virtuellen Tiefe in metrische Abstände definiert. Anschließend leiten wir Modelle basierend auf der geschätzten virtuellen Tiefenkarte und dem zugehörigen totalfokussierten Bild her. Diese Modelle berücksichtigen Tiefenverzeichnung und eine Sensorverkippung bezüglich der Hauptlinse. Schließlich wird ein Modell definiert, welches die komplette Projektion eines Objektpunkts in mehrere Mikrolinsenbilder auf dem Sensor beschreibt. Hier wird die Bedeutung der Modellierung von schielenden Mikrolinsen, wie sie in einer plenoptischen Kamera vorkommen, herausgearbeitet. Die Funktionen zur Tiefenumrechnung werden anhand einer Reihe von Abstandsmessungen bestimmt, während alle weiteren Modelle mit einem 3D Kalibrierobjekt im Bündelausgleich geschätzt werden. Die auf Bündelausgleichung basierende Kalibriermethoden erzielen robustere Ergebnisse als bisherige Methoden.

Die MVG für plenoptische Kameras, das Verfahren zur Tiefenschätzung und das während der Kalibrierung geschätzte Modell werden in einem VO-Algorithmus namens Direct Plenoptic Odometry (DPO) vereint. DPO arbeitet direkt auf den Mikrolinsenbildern und vermeidet damit Unterabtastungseffekte bei der Tiefenschätzung. Der Algorithmus erzeugt teildichte 3D Punktwolken basierend auf Korrespondenzen in aufeinanderfolgenden Lichtfeldaufnahmen. Nach unserem Wissen ist DPO die erste Methode welche die Kameraposition sowie eine 3D Karte aus den Aufnahmen einer plenoptischen Kamera bestimmt. DPO wurde mit verschiedenen Innenraum- und Außenbereichsequenzen getestet. Bezüglich des Skalierungsdrifts übertrifft DPO aktuellste monokulare VO und SLAM Algorithmen. Die absolute Genauigkeit liegt in der Größenordnung von aktuellen monokularen und stereobasierten Algorithmen.

für Miriam

Contents

List of Figures **xi**

List of Tables **xiii**

Abbreviations **xv**

1 Introduction **1**

1.1 Motivation . . . 1
1.2 Objectives . . . 3
1.3 Dissertation Overview . . . 3
1.4 Publications . . . 4
1.5 Collaborations . . . 5
1.6 Notations . . . 6

2 Fundamentals – Light Field Imaging and Visual SLAM **7**

2.1 The Light Field . . . 7
2.1.1 The Plenoptic Function . . . 7
2.1.2 Light Field Parametrization and Capturing . . . 8
2.2 The Plenoptic Camera . . . 9
2.2.1 The Unfocused Plenoptic Camera . . . 10
2.2.2 The Focused Plenoptic Camera . . . 11
2.3 Visualizations of 4D Light Field Recordings . . . 13
2.4 Depth Estimation for Plenoptic Cameras . . . 14
2.5 Visual SLAM . . . 16
2.5.1 Rigid Body and Similarity Transformation . . . 16
2.5.2 Lie Groups and Lie Algebra . . . 17
2.5.3 Nonlinear Optimization on Lie Manifolds . . . 21

3 Related Work **25**

3.1 Plenoptic Camera Calibration . . . 25
3.1.1 MLA Calibration & Light Field Resampling . . . 25
3.1.2 Unfocused Plenoptic Camera Calibration . . . 26
3.1.3 Focused Plenoptic Camera Calibration . . . 26
3.2 Depth Estimation in Light Fields . . . 27
3.3 Visual Localization and Mapping . . . 29
3.3.1 Monocular Algorithms . . . 29
3.3.2 RGB-D Sensor and Stereo Camera based Algorithms . . . 31
3.3.3 Light Field based Localization and Mapping . . . 33
3.4 Own Contribution . . . 34

4 The Plenoptic Camera from a Mathematical Perspective **37**

4.1 New Interpretation of a Plenoptic Camera . . . 37

4.2 Multiple View Epipolar Geometry 40
4.3 Properties and Limitations of Plenoptic Cameras 43
4.3.1 Effective Object Distance vs. Virtual Depth 43
4.3.2 Effective Stereo Baseline – Unfocused vs. Focused 44

5 Probabilistic Light Field based Depth Estimation 47
5.1 Probabilistic Virtual Depth 48
5.2 Graph of Stereo Baselines 48
5.3 Virtual Depth Estimation 50
5.3.1 Stereo Matching 50
5.4 Updating Virtual Depth Hypothesis 54
5.5 Calculating a Virtual Depth Map 55
5.6 Probabilistic Depth Map Filtering 56
5.6.1 Removing Outliers and Filling Holes in Micro Images 56
5.6.2 Regularization of the Virtual Image 57
5.7 Intensity Image Synthesis 59

6 Plenoptic Camera Calibration 63
6.1 Depth Conversion Functions 63
6.1.1 Physical Model 64
6.1.2 Behavioral Model 65
6.1.3 Polynomial based Curve Fitting 66
6.2 Intrinsic Models for Focused Plenoptic Cameras 67
6.2.1 Virtual Image Projection Model 67
6.2.2 Micro Images Projection Model 71
6.2.3 Extended Micro Images Projection Model 72
6.3 Plenoptic Camera based Bundle Adjustment 73
6.3.1 Initialization of Intrinsic and Extrinsic Parameters 74
6.3.2 Bundle Adjustment on Virtual Image Space 75
6.3.3 Full Bundle Adjustment on Micro Images 77

7 Full-Resolution Direct Plenoptic Odometry 79
7.1 Overview 79
7.2 Depth Map Representation 79
7.2.1 Micro Image Depth Map and Virtual Image Depth Map 81
7.3 Estimating Depth Hypotheses 81
7.3.1 Observation Uncertainty 81
7.3.2 Estimating In-Frame Depth 83
7.3.3 Estimating Inter-Frame Depth 83
7.3.4 Regularizing Depth Maps 84
7.4 Merging Depth Hypotheses 84
7.5 Tracking 84
7.5.1 Tracking on Pyramid Levels in the Micro Images 85
7.5.2 Variance of Tracking Residual 86
7.5.3 Compensation of Lighting Changes 87
7.5.4 Motion Prior for Tracking Robustness 87
7.6 Selecting Keyframes 88
7.7 Optimizing Global Scale 88
7.7.1 Keyframe based Scale Estimation 88
7.7.2 Scale Optimization 90

7.8 Finding Micro Lenses of Interest . . . 90

8 Experiments – Probabilistic Light Field based Depth Estimation 93
8.1 Data . . . 93
8.1.1 Checkerboard Recorded by the Plenoptic Camera Raytrix R5 . . . 93
8.1.2 Plenoptic Images of Real Scenes . . . 93
8.2 Results . . . 96
8.2.1 Quantitative Results . . . 96
8.2.2 Qualitative Results . . . 98
8.3 Conclusion . . . 101

9 Experiments – Plenoptic Camera Calibration 107
9.1 Data . . . 107
9.1.1 Series of Range Measurements . . . 107
9.1.2 Recordings of 3D Calibration Target . . . 108
9.2 Results . . . 109
9.2.1 Depth Conversion Functions . . . 110
9.2.2 Bundle Adjustment based Plenoptic Camera Calibration . . . 111
9.3 Conclusion . . . 116

10 Experiments – Full-Resolution Direct Plenoptic Odometry 119
10.1 Data . . . 119
10.1.1 Data Acquisition Setup . . . 119
10.1.2 A Synchronized Stereo and Plenoptic Visual Odometry Dataset . . . 120
10.2 Results . . . 128
10.2.1 Quantitative Results . . . 129
10.2.2 Qualitative Results . . . 134
10.3 Conclusion . . . 140

11 Summary and Prospects 141
11.1 Summary . . . 141
11.2 Future Work . . . 142

A Light Field based Depth Estimation 147
A.1 Relationship between Virtual Depth and Focus Parameter of Hog et al. [2017] . . . 147

B Direct Plenoptic Odometry 149
B.1 Pseudo Code of the DPO Algorithm . . . 149
B.2 Recursive Solution of the Global Scale Estimator . . . 151
B.3 Results from the Stereo and Plenoptic VO Dataset . . . 152
B.4 Trajectories Estimated by DPO . . . 153

Bibliography 159

Acknowledgments 167

Curriculum Vitae 169

List of Figures

1.1 Different light field sensors 2

2.1 Visualization of the plenoptic function 7
2.2 Two plane parametrization of a 4D light field 9
2.3 Different setups to acquire 4D light fields 10
2.4 Setup inside of an unfocused plenoptic camera 11
2.5 Setup inside of a focused plenoptic camera 12
2.6 4D light field visualizations 13
2.7 Cross section of a focused plenoptic camera in the Galilean mode 14
2.8 Depth estimation in a focused plenoptic camera based on the Galilean mode 16
2.9 Visualization of the tangent space of a smooth manifold 18

4.1 Imaging process for a focused plenoptic camera in the Galilean mode 38
4.2 Focused plenoptic camera interpreted as an array of pinhole cameras 39
4.3 Multiple view epipolar geometry for plenoptic cameras 41
4.4 Depth accuracy of plenoptic cameras based on plenoptic 1.0 and 2.0 rendering . . . 45

5.1 Five shortest stereo baseline distances in a hexagonal micro lens grid 49
5.2 Stereo matching cost function for different baseline distances and orientations . . . 49
5.3 Visualization of the photometric disparity error 53
5.4 Visualization of the focus disparity error 54
5.5 Projection of a raw image point $\boldsymbol{x}_R$ to a virtual image point $\boldsymbol{x}_V$ 55
5.6 Synthesis of the intensity value for a point in the virtual image 59
5.7 Exemplary result of the probabilistic depth estimation 61

6.1 Relationship between micro image and micro lens centers 72
6.2 3D target used for plenoptic camera calibration 73
6.3 Single calibration marker mapped to multiple micro images 74

7.1 Workflow of the Direct Plenoptic Odometry (DPO) algorithm 80
7.2 Keyframe depth maps and intensity images 82
7.3 Tracking residual after various numbers of iterations 86
7.4 Workflow of the scale optimization framework 88

8.1 Checkerboard dataset recorded by a Raytrix R5 camera 94
8.2 Own focused plenoptic camera dataset recorded by a Raytrix R5 camera 94
8.3 Focused plenoptic camera dataset published by Hog et al. [2017] 95
8.4 Raytrix R11 dataset 95
8.5 Virtual depth maps calculated for the checkerboard dataset 97
8.6 Virtual depth histograms of unfiltered virtual depth maps 98
8.7 Results for our own Raytrix R5 dataset 99
8.8 Results for the Raytrix R5 dataset of Hog et al. [2017] 100

8.9 Processed raw image of the “Lion” dataset of Hog et al. [2017] 101
8.10 Results for the Raytrix R11 dataset 102
8.11 Further results for the Raytrix R11 dataset 103
8.12 Estimation uncertainty obtained by our depth estimation algorithm 104

9.1 Setups for plenoptic camera calibration 108
9.2 Raw image samples showing a checkerboard pattern used for the calibration method of Bok et al. [2017] 109
9.3 Estimated depth conversion functions 110
9.4 Depth accuracy of a focused plenoptic camera for different model parameters and camera setups 113
9.5 Reprojection error map with and without the consideration of pixel distortion . . . 116
9.6 Position offset between camera model without and with squinting micro lenses . . . 117

10.1 Handheld platform to acquire time synchronized image sequences from a focused plenoptic camera and a stereo camera system 119
10.2 White balance filter used to record white images for the plenoptic camera and the stereo camera system 122
10.3 Estimated attenuation image for the plenoptic camera 123
10.4 Estimated attenuation images for the left camera of the stereo camera system . . . 124
10.5 Semi-dense point cloud generated by DPO overlaid with the ground truth loop closure trajectory 125
10.6 Sample images for one sequence of the stereo and plenoptic VO dataset 126
10.7 Example sequence of the stereo and plenoptic VO dataset 127
10.8 Example of the drift accumulated by DPO over an entire sequence 127
10.9 Measured absolute scale errors d'_s and scale drifts e'_s 130
10.10 Measured and optimized scale for DPO as a function of the keyframe index 132
10.11 Measured rotational drift and alignment error 133
10.12 Sample images of sequence #5 recorded by the plenoptic camera and the stereo camera 134
10.13 Comparison of point clouds received from DPO and LSD-SLAM 135
10.14 Comparison of image details for the plenoptic and the monocular camera 135
10.15 Point cloud reconstructed by DPO with different thresholds on the inverse effective depth variance 136
10.16 Complete trajectory and point cloud calculated by DPO for a scene with abundant vegetation 136
10.17 Point cloud of a hallway reconstructed by DPO 137
10.18 Point clouds of stairways reconstructed by DPO 138
10.19 Samples of the point clouds calculated by DPO 139

B.1 Top views of the trajectory of sequence #1 153
B.2 Top views of the trajectory of sequence #2 154
B.3 Top views of the trajectory of sequence #3 154
B.4 Top views of the trajectory of sequence #4 155
B.5 Top views of the trajectory of sequence #6 155
B.6 Top views of the trajectory of sequence #7 156
B.7 Top views of the trajectory of sequence #9 156
B.8 Top views of the trajectory of sequence #10 157
B.9 Top views of the trajectory of sequence #11 157

List of Tables

8.1 Inverse virtual depth statistics obtained from the checkerboard dataset 96

9.1 Estimated intrinsic camera parameters and reprojection errors 112
9.2 Accuracy of 3D object coordinates estimated during the bundle adjustment 112
9.3 Robustness of the estimated intrinsic camera parameters 114

10.1 Camera specifications for the plenoptic camera and the stereo camera system . . . 120

B.1 All results from the stereo and plenoptic VO dataset 152
B.2 Path lengths for all sequences in the stereo and plenoptic VO dataset 152

Abbreviations

1D	one dimensional	18
2D	two dimensional	2
2PP	two plane parametrization	8
3D	three dimensional	1
4D	four dimensional	2
AWGN	additive white Gaussian noise	48
CNN	convolutional neural network	28
CPU	central processing unit	31
DFT	discrete Fourier transform	26
DOF	depth of field	11
DPO	Direct Plenoptic Odometry	34
DSLR	digital single-lens reflex	
DSO	Direct Sparse Odometry	129
EGI	extended Gaussian image	32
EKF	extended Kalman filter	30
EPI	epipolar plane image	13
FIR	finite impulse response	90
FOV	field of view	44
GPS	Global Positioning System	
GPU	graphics processing unit	31
ICP	Iterative Closest Point	32
IMU	inertial measurement unit	33
log-scale	logarithmized scale	89
LRF	laser rangefinder	64
MAE	maximum absolute error	112
MLA	micro lens array	2
MVG	multiple view geometry	3
PCA	principal component analysis	28
RGB	red, green, blue	
RGB-D	red, green, blue - depth	
RMS	root mean square	111
RMSE	root mean square error	111

SfM	structure from motion	29
SGM	Semi-Global Matching	129
SLAM	simultaneous localization and mapping	1
SNR	signal to noise ratio	60
SSID	sum of squared intensity differences	81
TCP	total covering plane	51
UAV	unmanned aerial vehicle	1
VO	visual odometry	2

1 Introduction

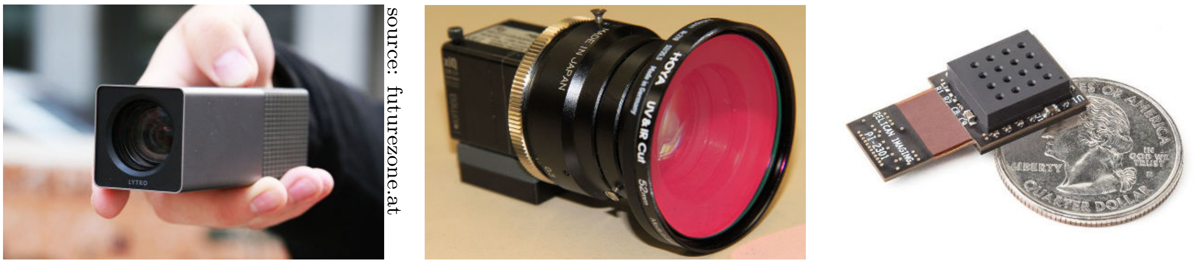
1.1 Motivation

The sense of vision allows us humans to navigate in any kind of environment without interacting with it. Furthermore, we are able to build a three dimensional (3D) map of the surroundings in our brain, based on what we see. In this way vision, in combination with other senses, enable us to navigate through and interact with our environment. In the century of automation, where we want to build autonomous cars, self-navigating unmanned aerial vehicles (UAVs) as well as mobile and intelligent robots, teaching a computer the sense of vision became a very important and broadly studied field both in research and industry. Aside from its use in automation, 3D vision can also be used to assist for instance visually impaired and elderly people in order to enhance their quality of life.

For some of the mentioned tasks, like localizing a car on an existing road map, external infrastructure can be established to achieve this goal. However, this is not always possible. Inaccuracies in existing maps or even the lack of maps are examples of such cases. Furthermore, it might be difficult to keep maps of transient environments (e.g. moving objects and people) updated. Depending on the application, establishing such an infrastructure might simply be too expensive. Therefore, even though various systems which allow localization on a global and local scale already exist, a growing demand for closed systems which do not rely on any infrastructure for navigation and mapping can be observed.

Such tasks based on closed systems can be summarized under the research topic of simultaneous localization and mapping (SLAM). This means that a computer-based movable system is able to build a map, usually in 3D, of the environment and to localize itself within this map. Furthermore, tracking and mapping are supposed to be performed online since the resulting map and trajectory have to be used e.g. for obstacle avoidance.

Depending on the application, the SLAM problem can be tackled by incorporating many different active and passive sensors, which balance each other's capabilities and thereby are able to compensate for the failure of individual sensors. Furthermore, autonomous cars can combine the information gained through their own sensor system with that of external infrastructure like GPS (Global Positioning System). However, there are other applications where the number, size, and weight of the sensors as well as the power consumption matters. UAVs and small household robots for instance can only carry a limited weight. Furthermore, for indoor scenarios there usually are no external localization systems available and existing maps are not detailed enough and outdated since they generally do not include furniture, etc. Similar concerns hold true for assistance devices, which at best are so small and light that they are almost invisible. Besides, it is crucial that such devices operate reliably both indoors and outdoors and therefore must not be dependent on external systems. Furthermore, while some active sensors (e.g. structured light sensors) perform reliably in indoor environments with artificial lighting, they completely fail in sunlight conditions.



source: futurezone.at

source: Pelican Imaging

(a) Lytro camera (1. generation) (b) Raytrix R5 camera (c) PiCam, Pelican Imaging

Figure 1.1: Different light field sensors. While (a) and (b) are both micro lens array based plenoptic cameras, (c) is a miniaturized 4×4 camera array.

In these scenarios, where weight and power consumption matter, the SLAM problem is generally solved based on a single monocular camera. A monocular camera captures only two dimensional (2D) images of the 3D world. Therefore, a 3D map can not be build without observing the scene multiple times from different perspectives. Since no absolute depth can be measured using a monocular camera, tracking and mapping can only be performed up to an unknown scale factor.

Over the last few decades, plenoptic cameras, which are also called light field cameras, with a size comparable to standard industry cameras have emerged (Adelson and Wang [1992]; Ng et al. [2005]; Lumsdaine and Georgiev [2009]; Perwaß and Wietzke [2012]) and were developed up to a commercial stage. Unlike monocular cameras, which record a 2D image of a scene, plenoptic cameras capture the light field of a scene as a four dimensional (4D) function in a single image. The additional information in two angular dimensions allows the reconstruction of the 3D scene from a single image to a certain degree, and therefore to measure the absolute scale of a scene. Figure 1.1 shows three different, portable light field sensors. While the cameras from Lytro and Raytrix are plenoptic cameras which gather angular information based on a micro lens array (MLA) in front of the image sensor, the PiCam from Pelican Imaging is a miniaturized camera array. Here, an array of 4×4 cameras is realized on a single image sensor.

In plenoptic cameras the additional angular information is obtained at the expense of spatial resolution. However, as pixel sizes continue to shrink (currently around 1 µm) and image resolutions of over 40 million pixels on a sensor of about 17 mm × 17 mm in size can be manufactured, it seems to be a good compromise to reduce spatial resolution in order to gain additional information about the environment. This claim is supported by Engel et al. [2016] who showed that, at least for direct SLAM approaches, increasing the image resolution does not necessarily lead to more accurate tracking results.

The motivation of this work was to study the pros and cons of plenoptic cameras with respect to visual odometry (VO) and SLAM. One question in particular, which is hoped to be answered within the context of this work, was whether or not the absolute scale of a scene can be recovered from a sequence of plenoptic images. It was also supposed to be examined whether the additional angular information gained by a plenoptic camera leads to improved tracking capabilities in comparison to monocular cameras.

In this dissertation often the two terms SLAM and VO are used to describe at first glance the same task. However, these tasks are different by definition. VO is the task of estimating the motion and therefore the trajectory of a camera (or multiple cameras) based on the recorded images. SLAM furthermore establishes a map of its environment which aims for global consistency. Hence, a SLAM system stores map information of previously explored regions and will recognize these regions when they are revisited. This is called loop closure. A VO system stores only transient map information since it is needed to estimate the camera trajectory. Therefore, loop-closures will not be recognized.

1.2 Objectives

The objective of this dissertation is to explore plenoptic cameras, built on the basis of standard industrial cameras, with respect to the task of 3D tracking and mapping and evaluate the suitability of these cameras for applications such as robot navigation or vision assistance for visually impaired persons.

The overarching goal of this dissertation can therefore be stated as the development and investigation of a plenoptic camera based 3D tracking and mapping algorithm. This problem is supposed to be solved by way of an example for a focused plenoptic camera by the manufacturer Raytrix (model R5). While the previous sentences define this work's intent entirely, there are a number of secondary problems that arise on the way to achieving this goal.

One can consider the problem of camera calibration as solved for monocular cameras and stereo cameras. So far, this is not the case for plenoptic cameras. Hence, one goal was to solve the problem of plenoptic camera calibration while simultaneously finding a mathematical model for the camera that is suitable for a multiple view system.

On the basis of the camera model, tracking and depth estimation algorithms, which are able to process sequences of plenoptic images, had to be developed and combined in a plenoptic camera based VO framework.

1.3 Dissertation Overview

Following the introduction, Chapter 2 introduces some fundamentals with respect to light field imaging as well as visual SLAM. This dissertation combines these two topics which have been handled as two more or less independent fields of research so far. Mathematical terms which are needed throughout this dissertation will be defined.

Chapter 3 gives an overview about related work and the state-of-the-art in the field of this thesis. Publications are divided into the categories of light field based depth estimation, plenoptic camera calibration, and VO (or visual SLAM). Furthermore, the main contributions of this thesis with respect to the state-of-the-art are listed.

Chapter 4 investigates the plenoptic camera from a mathematical perspective. A new mathematical model for MLA based light field cameras is introduced. Based on this mathematical model a multiple view geometry (MVG) can be defined directly for the micro images recorded by the camera. Furthermore, the two concepts of focused and unfocused plenoptic cameras are investigated with respect to the task of depth estimation.

Chapter 5 introduces a probabilistic depth estimation algorithm based on a single recording of a focused plenoptic camera. The algorithm performs depth estimation without any metric calibration of the plenoptic camera. Multiple depth observations are incorporated into a single depth estimate based on a probabilistic model. A filtering method is presented to refine the probabilistic depth map. Based on this refined depth map a so called totally focused image of the recorded scene can be synthesized.

Chapter 6 presents intrinsic camera models and calibration approaches for focused plenoptic cameras on different abstraction levels. In a first model, the camera is considered to follow a pinhole camera model, while the estimated virtual depth is transformed into object distances based on a depth conversion function. In a second model, the projection from object space to the image created by the main lens is defined. Here, in addition to lateral distortion a depth distortion model has to be defined, since depth estimation is performed prior to any image correction. The last model defines the plenoptic camera in the most complete way. It defines the projection of a 3D object point to multiple micro images on the sensor. Since distortion is applied directly to the recorded raw image, it is no longer necessary to consider any depth distortion.

Chapter 7 formulates a direct and semi-dense VO algorithm based on a focused plenoptic camera. This algorithm combines the mathematical plenoptic camera model and multiple view geometry (Chapter 4), the probabilistic depth estimation (Chapter 5) and the metric camera calibration (Chapter 6).

Chapter 8, Chapter 9, and Chapter 10 present the experiments which were performed to evaluate the methods developed in this thesis. To evaluate the probabilistic depth estimation algorithm two public datasets as well as light field images recorded by us are used. For the plenoptic camera calibration there are no suitable public datasets available. The used dataset was recorded based on a 3D calibration target. Similarly, up to now there exist no public datasets for light field based VO. Therefore, a synchronized dataset was recorded to compare light field based VO with VO algorithms relying on other sensors. The results obtained for the proposed methods based on the datasets are presented and extensively discussed.

Chapter 11 summarizes this dissertation and presents prospects for further improvements and future work.

1.4 Publications

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

1. Zeller N, Quint F, and Guan L (2014a). Hinderniserkennung mit Microsoft Kinect. In *34. Wissenschaftlich-Technische Jahrestagung der DGPF (DGPF Tagungsband 23 / 2014)*.

2. Zeller N, Quint F, and Guan L (2014b). Kinect based 3D Scene Reconstruction. In *International Conference on Computer Graphics, Visualization and Computer Vision (WSCG)*, volume 22, pages 73–81.

3. Zeller N, Quint F, and Stilla U (2014c). Applying a Traditional Calibration Method to a Focused Plenoptic Camera. In *BW-CAR Symposium on Information and Communication Systems (SInCom)*.

4. Zeller N, Quint F, and Stilla U (2014d). Calibration and Accuracy Analysis of a Focused Plenoptic Camera. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences (Proc. PCV 2014)*, II-3:205–212.

5. Zeller N, Quint F, and Stilla U (2014e). Kalibrierung und Genauigkeitsuntersuchung einer fokussierten plenoptischen Kamera. In *34. Wissenschaftlich-Technische Jahrestagung der DGPF (DGPF Tagungsband 23 / 2014)*.

6. Zeller N, Quint F, Zangl C, and Stilla U (2014f). Edge Segmentation in Images of a Focused Plenotic Camera. In *International Symposium on Electronics and Telecommunications (ISETC)*, pages 269–272.

7. Zeller N, Quint F, and Stilla U (2015a). Establishing a Probabilistic Depth Map from Focused Plenoptic Cameras. In *International Conference on 3D Vision (3DV)*, pages 91–99.
8. Zeller N, Quint F, and Stilla U (2015b). Filtering Probabilistic Depth Maps Received from a Focused Plenoptic Camera. In *BW-CAR Symposium on Information and Communication Systems (SInCom)*.
9. Zeller N, Quint F, and Stilla U (2015c). Narrow Field-of-View Visual Odometry based on a Focused Plenoptic Camera. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences (Proc. PIA 2015)*, II-3/W4:285–292.
10. Vasko R, Zeller N, Quint F, and Stilla U (2015). A Real-Time Depth Estimation Approach for a Focused Plenoptic Camera. In *Advances in Visual Computing (Proc. ISVC 2015)*, volume 9475 of *Lecture Notes in Computer Science*, pages 70–80.
11. Zeller N, Noury CA, Quint F, Teulière C, Stilla U, and Dhôme M (2016a). Metric Calibration of a Focused Plenoptic Camera based on a 3D Calibration Target. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences (Proc. ISPRS Congress 2016)*, III-3:449–456.
12. Zeller N, Quint F, and Stilla U (2016b). Depth Estimation and Camera Calibration of a Focused Plenoptic Camera for Visual Odometry. *ISPRS Journal of Photogrammetry and Remote Sensing*, 118:83 – 100.
13. Zeller N, Quint F, Sütterlin M, and Stilla U (2016c). Investigating Mathematical Models for Focused Plenoptic Cameras. In *International Symposium on Electronics and Telecommunications (ISETC)*, volume 12, pages 301–304.
14. Kühefuß A, Zeller N, and Quint F (2016). Feature Based RGB-D SLAM for a Plenoptic Camera. In *BW-CAR Symposium on Information and Communication Systems (SInCom)*, pages 25–29.
15. Konz J, Zeller N, and Quint F (2016). Depth Estimation from Micro Images of a Plenoptic Camera. In *BW-CAR Symposium on Information and Communication Systems (SInCom)*, pages 17–23.
16. Zeller N, Quint F, and Stilla U (2017). From the Calibration of a Light-Field Camera to Direct Plenoptic Odometry. *IEEE Journal of Selected Topics in Signal Processing (JSTSP)*, 11(7):1004–1019.
17. Zeller N, Quint F, and Stilla U (2018). Scale-Awareness of Light Field Camera based Visual Odometry. In *European Conference on Computer Vision (ECCV)*, pages 715–730.

1.5 Collaborations

Parts of this thesis have been developed in collaboration with other colleagues. The real time implementation of the probabilistic depth estimation algorithm (Vasko et al. [2015]) was done by Ross Vasko from the Ohio State University, USA, as a summer intern at Karlsruhe University of Applied Sciences. The plenoptic camera calibration approach which was published at the ISPRS Congress 2016 in Prague (Zeller et al. [2016a] (parts of Sections 6.2.1, 6.3.1, and 6.3.2)) was developed in collaboration with Charles-Antoine Noury from Institut Pascal in Clermont-Ferrand, France, during his stay in Karlsruhe.

1.6 Notations

In this section important notations, which are used throughout this dissertation, are introduced. Matrices are denoted as bold, capital letters ($\boldsymbol{G}$), vectors as bold, lower case letters ($\boldsymbol{x}$) and scalars as normal letters, either capital or lower case (d). It is not differentiated between homogeneous ($\boldsymbol{x} = [x, y, z, 1]^T$) and non-homogeneous ($\boldsymbol{x} = [x, y, z]^T$) representations. Nevertheless, the meaning should be clear from the context. The notation $[\boldsymbol{t}]_i$ defines the i-th element of a vector $\boldsymbol{t}$ and $[\boldsymbol{R}]_j$ the j-th row of a matrix $\boldsymbol{R}$. The determinate of a matrix $\boldsymbol{R}$ is denoted by $|\boldsymbol{R}|$ and $\mathrm{tr}(\boldsymbol{R})$ denotes the trace of a matrix $\boldsymbol{R}$.

2 Fundamentals – Light Field Imaging and Visual SLAM

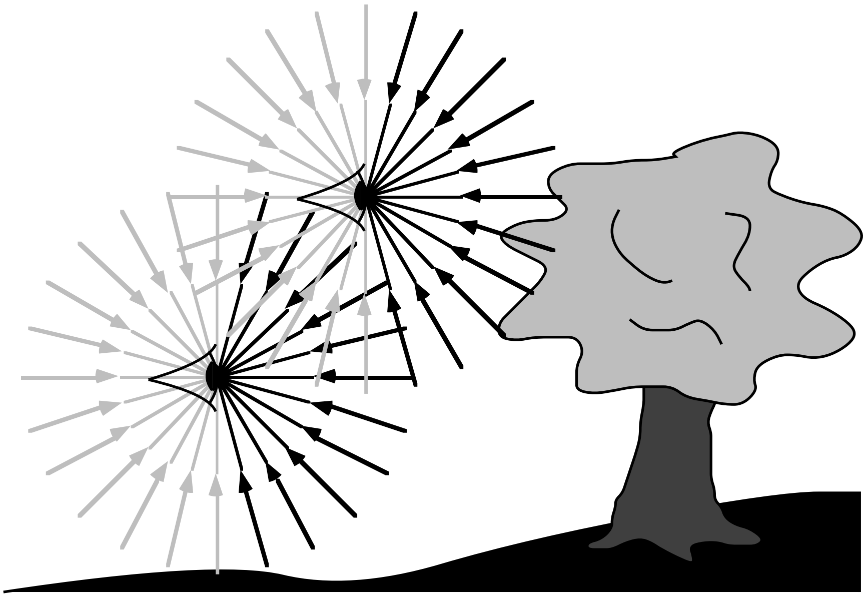
2.1 The Light Field

While a regular camera captures only the amount of light hitting a 2D image sensor, the light field is what describes the entire distribution of light in a 3D volume. First explorations of the light field were made by Gershun [1936], more than eighty years ago.

2.1.1 The Plenoptic Function

A general definition of the light field, as it is used today in image processing, was given by Adelson and Bergen [1991]. They described the entire distribution of light in 3D space by a seven dimensional function which they called the *plenoptic function*. Here, the intensity distribution of light is defined for any point (V_x, V_y, V_z) in the 3D space, with respect to the angle ($\mathit{\Phi}$, $\mathit{\Theta}$) of the respective light ray, at a certain wavelength λ, for any time t:

$$P(\mathit{\Phi}, \mathit{\Theta}, \lambda, V_x, V_y, V_z, t). \tag{2.1}$$



source: Adelson and Bergen [1991]

Figure 2.1: Original drawing by Adelson and Bergen [1991], which is a quite famous visualization of the plenoptic function. The plenoptic function describes the amount of light penetrating any point in space (V_x, V_y, V_z) from any possible angle ($\mathit{\Phi}$, $\mathit{\Theta}$). This is shown here schematically with two eyes representing two different observer positions. While a real observer does not gather light rays coming from behind, these are included in the plenoptic function.

While there might be even more dimensions, describing, for instance, the light's polarization, the plenoptic function stated in eq. (2.1) gives a quite generalized definition of the light field. Anyhow, as one will see in the following, several assumptions can be made with respect to computational imaging. The plenoptic function considered here will only be a subset of the function given in eq. (2.1).

When processing images, one is not interested in the absolute time t. Even for the task of visual odometry as discussed in this thesis, the assumption of a light field which is stationary, at least in a wide sense, has to be made. Furthermore, when considering a monochromatic image sensor, the plenoptic function is integrated along a certain bandwidth of the wavelength domain. Therefore, the function will be independent of the wavelength λ itself. Hence eq. (2.1) shrinks down to a five dimensional function:

$$P(\mathit{\Phi}, \mathit{\Theta}, V_x, V_y, V_z). \tag{2.2}$$

For RGB sensors one can simply consider the three colors as separate channels of the plenoptic function. The function given in eq. (2.2) can be visually interpreted as shown in Figure 2.1, which is the original drawing by Adelson and Bergen [1991]. Here, the coordinates (V_x, V_y, V_z) can be considered as the position of an observer, while $(\mathit{\Phi}, \mathit{\Theta})$ are the angles corresponding to a light ray going through the observer's pupil. While a real-eye observer would only gather the light rays coming from in front, the plenoptic function considers all rays penetrating from every direction.

Another simplification of the plenoptic function can be made when considering the intensity distribution P along a ray being independent of the absolute position along the respective light ray. This assumption holds true as long as the ray is traveling through free space without hitting any object or being attenuated by some medium. One can parametrize the light field on a surface outside the convex hull of the scene as given in eq. (2.3) (Gortler et al. [1996]).

$$P(\mathit{\Phi}, \mathit{\Theta}, x, y) \tag{2.3}$$

This 4D plenoptic function has its limitation in that only the light field around a scene bounded by a convex hull can be parameterized. While this assumption implies certain limitations, the 4D plenoptic function represents the nature of light fields captured by common imaging systems.

2.1.2 Light Field Parametrization and Capturing

While in the previous section the plenoptic function was considered as a continuum, what we capture is just a sampled version of it, where the intensity distribution is integrated over a certain area (e.g. the photosensitive are of a sensor pixel). There are several ways to parametrize this sampled 4D light field. The most popular one is the two plane parametrization (2PP) introduced by Levoy and Hanrahan [1996]. A ray in the 4D light field is defined by its intersections with two parallel planes $\mathit{\Omega}$ and $\mathit{\Pi}$, where the intersections are defined by the coordinates $(u, v) \in \mathit{\Omega}$ and $(s, t) \in \mathit{\Pi}$ respectively:

$$L(u, v, s, t). \tag{2.4}$$

Each ray (u, v, s, t) in the light field is described by its corresponding intensity value L. Here, it is important to notice that the variable t does not represent time. Figure 2.2 visualizes this two plane parametrization. The figure illustrates the example of two light rays (L_1 and L_2) of the 4D light field. Both rays are emitted from the same point $O(V_x, V_y, V_z)$ in 3D space and therefore intersect in this point. Since both rays are emitted from the same point, they will have the same intensity value ($L_1 = L_2$), as long as Lambertian reflectance is assumed.

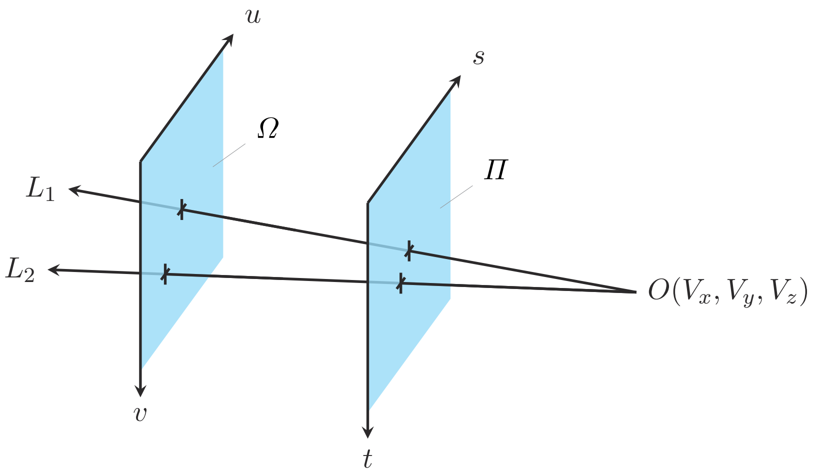


Figure 2.2: Two plane parametrization (2PP). Light rays are defined based on their intersections with two parallel planes and have a certain intensity value L. The figure shows the example of two light rays L_1 and L_2 emitted from the same point $O(V_x, V_y, V_z)$ in the 3D space.

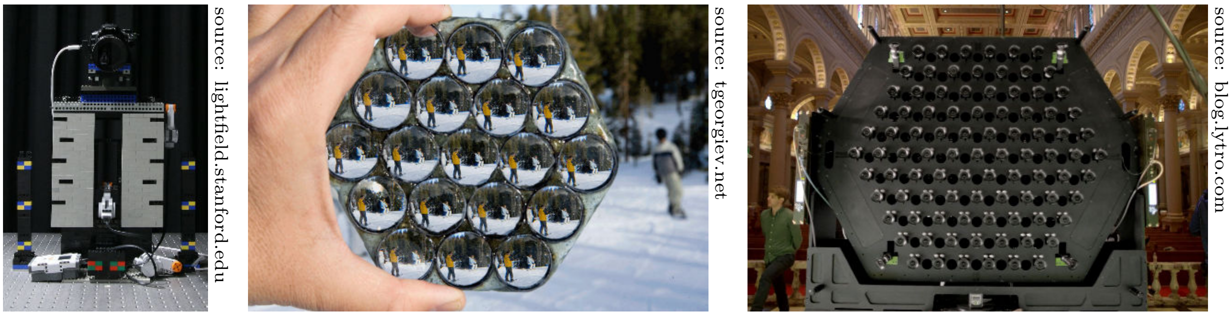
At first glance this parametrization seems to have quite a few limitations. Only the light field outside a convex hull and on a planar cut through the 3D space is covered. However, the two plane parametrization is sufficient for the data gathered by most of today's light field capturing setups. Some of these setups are shown in Figure 2.3, which all basically form arrays of real or virtual cameras. Figure 2.3a shows a gantry, which is able to capture the 4D light field of a static scene by moving a camera in horizontal and vertical direction. Figure 2.3b shows an array of lenses and prisms, which can be used to capture a light field using a single camera. Figure 2.3c shows a large camera array, based on which high resolution light field videos of dynamic scenes can be captured.

A convenient feature of 4D light fields captured by camera arrays is that they already are recorded in a format conforming to the two plane parametrization. Consider a camera array, for instance, where all cameras are arranged on a plane and are pointing in the same direction. The plane Ω can be defined as the plane of the camera lenses, while the plane Π defines the respective image plane. Hence, for the conventional pinhole camera model, the plane Π is defined in distance f in front of the plane Ω, where f is the cameras' focal length. Selecting certain coordinates (u, v) on the plane Ω is equivalent to the selection of a certain camera in the array. In contrast selecting coordinates (s, t) would be equivalent to the selection of a respective pixel in the rectified images. Even though both coordinates (u, v) and (s, t) define positions on the respective plane, the coordinates (s, t) are referred to as spatial coordinates while (u, v) are referred to as angular coordinates. The coordinates (s, t) define the spatial samples in a single image of the camera array, while the coordinates (u, v) define the position of the respective pinhole camera and hence the viewing angle on the scene.

2.2 The Plenoptic Camera

For camera arrays it is quite intuitive to see that they capture a 4D light field which conforms to the two plane parametrization. However, it is not as obvious that a MLA based light field camera, as drawn in Figure 2.4, does so, too.

In this section we will discuss both concepts of MLA based light field cameras: the traditional, unfocused plenoptic camera (or plenoptic camera 1.0) proposed by Adelson and Wang [1992] and



source: lightfield.stanford.edu

source: tgeorgiev.net

source: blog.lytro.com

(a) camera gantry (b) lens and prism array (c) Lytro Immerge camera array

Figure 2.3: Different setups to acquire 4D light fields. (a) Stanford's Lego Mindstorms gantry. Here an assembled DSLR camera can be moved in horizontal and vertical direction to capture images from different viewpoints on a plane. (b) Optical device by Georgiev et al. [2006] consisting of 19 lenses and 18 prisms. It can be placed in front of a regular camera lens to capture 4D light fields. (c) Lytro Immerge. Large camera array to capture 4D light field videos in cinema quality.

developed further by Ng et al. [2005] and the focused plenoptic camera (or plenoptic camera 2.0) introduced by Lumsdaine and Georgiev [2009] and commercialized by Perwaß and Wietzke [2012].

2.2.1 The Unfocused Plenoptic Camera

There are different ways to look at plenoptic cameras. In contrast to the perspective presented here, there are plenty of other ways to describe the concept of plenoptic imaging (e.g. Dansereau [2013] (Sec. 2.3)). Here, the plenoptic camera is considered to be an array of cameras that captures the light field inside the camera and therefore behind the main lens. As will be shown in the following, this light field conforms to the two plane parametrization. Hahne et al. [2014] show how this array can be transformed into an array outside the camera. However, here the data conforms to the two plane parametrization only under certain conditions.

For an unfocused plenoptic camera, the focal length f_M of the micro lenses is chosen to be equal to the distance B between sensor and MLA (see Fig. 2.4). Thus, each micro lens is focused to infinity and thus each pixel under a micro lens integrates intensities over a bundle of rays which incident in parallel on the micro lens. For the light field parametrization one plane of the two plane parametrization can be chosen to be the MLA plane, where each micro lens is one sample, while the second plane is the main lens plane sampled by the pixels under a micro lens. Hence, the plane Ω is defined to be on the MLA and the plane Π on the main lens.

This setup of the unfocused plenoptic camera is shown in Figure 2.4. As one can see from the figure, each micro lens gathers light across the complete aperture of the main lens, while each pixel under the micro lens captures only rays from a small sub-aperture. From the previous section one can remember that picking a point (u, v) on the plane Ω can be considered as selecting a specific camera from the camera array, while the central perspective image of this camera is formed on the plane Π. This can be done in a similar way for the plenoptic camera. By selecting a point (u, v) one chooses a certain sub-aperture on the main lens and respectively a certain pixel position under the micro lenses. Hence, by arranging the pixels with certain coordinates from each micro image in an array, according to the corresponding micro lens position (s, t), again, a central perspective view is obtained (see Figure 2.6a). This is a simplified perspective on the image recorded by a plenoptic camera, where it is assumed that the MLA consists of rectangular micro lenses and each micro image consists of an integer number of pixels. For real data, interpolation and resampling have to be performed. However, the principle of the concept is as stated.

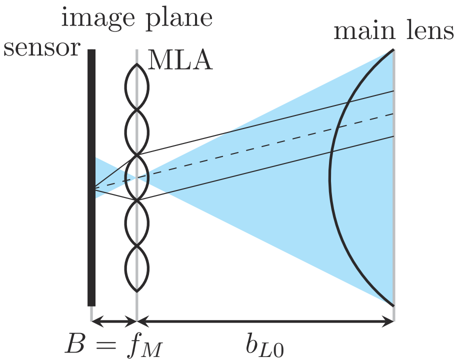


Figure 2.4: Setup inside of an unfocused plenoptic camera. While each micro lens captures light penetrating the whole of the main lens aperture (shown in blue), a single pixel on the sensor only gathers a bundle of rays penetrating a small subsection of the main lens aperture.

In this thesis, plenoptic cameras are mainly considered from a geometric perspective. But for interested readers who seek further insight into the topic of plenoptic imaging, the author would like to refer them to the work of Dansereau [2013]. In his PhD thesis (Dansereau [2013], Section 2.3), he compares a traditional camera to a plenoptic camera with respect to the captured depth of field (DOF) as well as the amount of light which is gathered.

2.2.2 The Focused Plenoptic Camera

One limitation of unfocused plenoptic cameras is that the spatial resolution, i.e. the resolution of the plane Ω in the 2PP, is limited by the number of micro lenses in the MLA. The idea behind the focused plenoptic camera or plenoptic camera 2.0 is to find an imaging concept which decouples this connection and therefore adds more degrees of freedom to the design of a plenoptic camera. Instead of placing the MLA in the image plane of the main lens, it is placed either in front of or behind that image plane. Furthermore, the micro lenses are not focused at infinity, but at the image plane of the main lens. This is shown in Figure 2.5 for an image plane in front of the MLA, called Keplerian mode (Figure 2.5a), and an image plane behind the MLA, called Galilean mode (Figure 2.5b). A micro lens with focal length $f_M < B$ has to be chosen for the Keplerian mode and $f_M > B$ for the Galilean mode. More precisely, for the Keplerian mode the micro lens focal length is given by

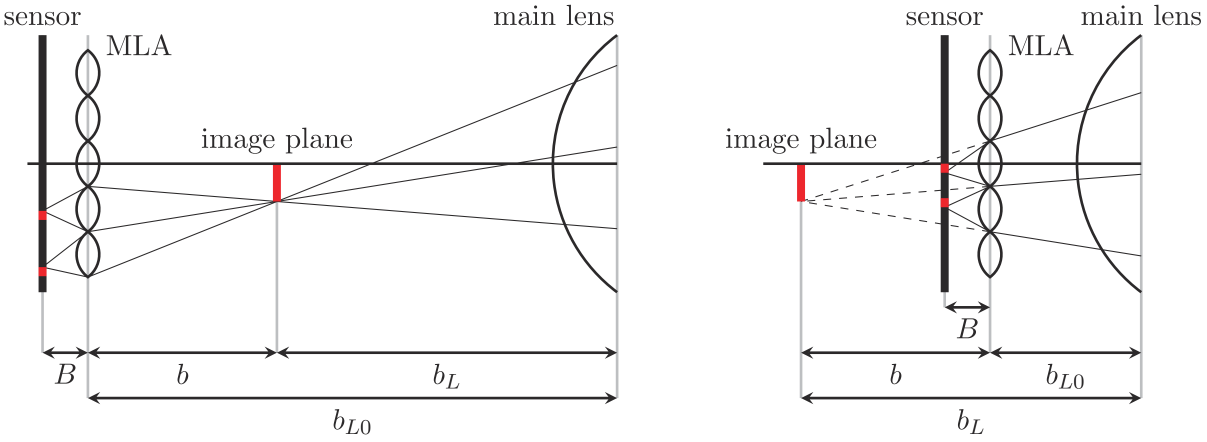
$$f_M = \frac{b \cdot B}{b + B}, \tag{2.5}$$

while for the Galilean mode a micro lens focal length of

$$f_M = \frac{b \cdot B}{b - B} \tag{2.6}$$

results from the thin lens equation. While in the Keplerian mode a real main lens image is formed and captured by the MLA, the image of the main lens in the Galilean mode exists only as virtual image behind the image sensor.

In the setup of a focused plenoptic camera, the micro images are focused subsections of the main lens image. Hence, this main lens image plane is sampled with a spatial resolution which is decoupled from the number of micro lenses in the MLA. Instead, the spatial resolution on the image plane is defined by the ratio $\frac{b}{B}$ and the pixel pitch. Increasing the distance b between image plane and MLA will decrease the spatial resolution, while reducing the distance will increase it accordingly. At the same time that the spatial resolution is increased, the number of angular



(a) Keplerian configuration (b) Galilean configuration

Figure 2.5: Setup inside of a focused plenoptic camera. (a) Keplerian configuration: the image plane of the main lens is in front of the MLA. (b) Galilean configuration: the image plane of the main lens is behind the MLA. In both cases the micro lens' focal lengths are matched such that the main lens image is focused on the sensor. In this schematic drawing aperture matching between micro lenses and main lens was not considered.

samples decreases, since an image point is seen by fewer micro lenses. Thus, since the total number of rays is given by the number of sensor pixels, a tradeoff between spatial and angular resolution has to be found. This phenomena was discussed exhaustively by Georgiev et al. [2006].

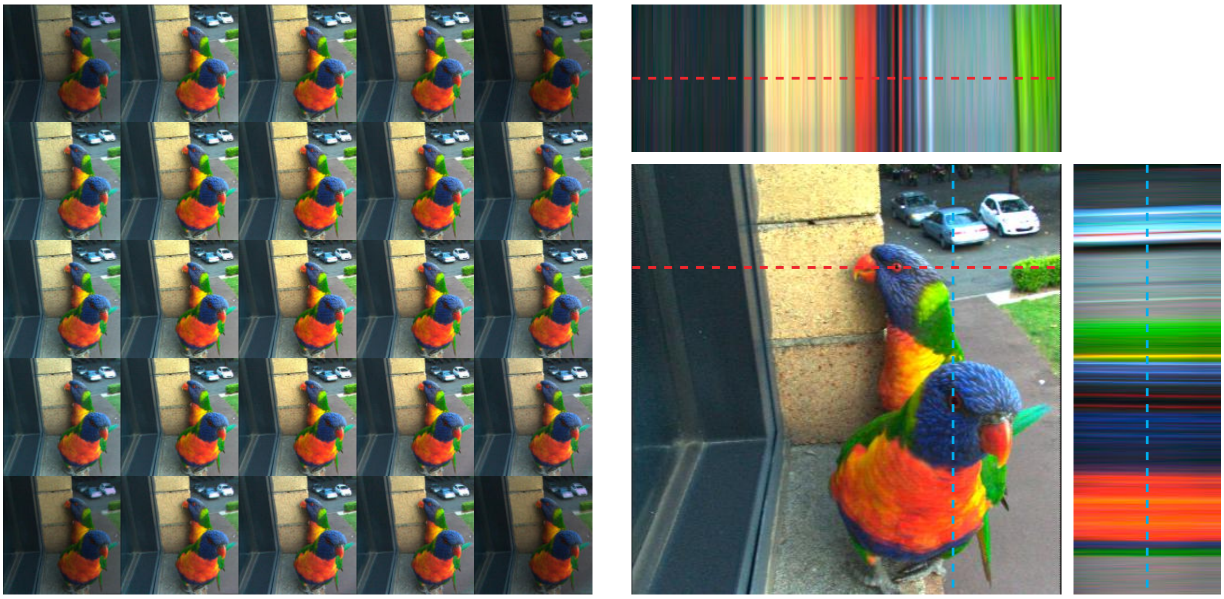
Assuming $\frac{b}{B} = 1$, for instance, would imply that each image point is seen by only a single micro lens. Therefore, for each image point there would exist only one angular sample, while the spatial domain is sampled at full sensor resolution. For this case the focused plenoptic camera would, in theory, act as the equivalent of a monocular camera.

Similar to the way that the number of captured light rays is directly coupled to the number of pixels on the sensor, the depth of field (DOF) captured by the camera is coupled to the angular resolution. Hence, increasing the spatial resolution will result in a decreased DOF in comparison to the equivalent unfocused plenoptic camera.

In conclusion, the main difference between an unfocused and a focused plenoptic camera is the manner of sampling. In an unfocused plenoptic camera, the MLA samples the spatial domain of the light field and the sensor pixels sample the angular domain. In the case of the focused plenoptic camera, the reverse is true. Generally speaking, an unfocused plenoptic camera exhibits more vagueness in the spatial domain, while a focused plenoptic camera exhibits more vagueness in the angular domain. This can also be seen when considering the ray bundle of a single pixel in Figure 2.4 and Figure 2.5, respectively.

With regards to the 2PP, the unfocused plenoptic camera already samples the light field in a uniform fashion, whereas the focused plenoptic camera performs a nonuniform sampling of the light field, which cannot be arranged directly in an equally spaced 2PP. Therefore, obtaining a uniformly sampled 2PP from a focused plenoptic camera requires depth estimation and interpolation of the light field (Wanner et al. [2011]).

A further extension to the focused plenoptic camera is the multi-focus plenoptic camera proposed by Perwaß and Wietzke [2012]. Here a MLA consisting of different interlaced types of micro lenses is used. Each micro lens type has a different focal length and therefore is focused on a different image plane. This way the lost depth of field can be compensated for, to some degree.


(a) sub-aperture images

(b) center sub-aperture image and EPIs

Figure 2.6: 4D light field visualizations. (a) Sub-aperture images: each image corresponds to a specific sample (u, v) on the plane Ω and hence is showing the scene from a slightly different perspective. (b) Epipolar plane images (EPIs): top and right part show slices through the (v, t) and (u, s) domain of the light field respectively. The resulting EPIs represent a point of the scene as a straight line of constant color. Here, the slope of the line is directly correlated to the depth of the scene point. The image in the center shows the central sub-aperture image. The light field was resampled from the recoding of a Lytro camera using the methods of Dansereau et al. [2013].

However, this setup makes the interpretation of the recorded data in the sense of a 4D light field even more complicated.

2.3 Visualizations of 4D Light Field Recordings

There are mainly two popular ways to visualize recorded 4D light fields. They will be discussed in the following. These are the so-called *sub-aperture images* and the *epipolar plane images (EPIs)*. Both visualizations represent the light field as a set of 2D functions and therefore as a set of 2D slices through the two plane parametrization (2PP).

The sub-aperture images are the most obvious visualization of light fields. As discussed before, selecting coordinates (u, v) on the plane Ω in the 2PP can be interpreted as picking a pinhole camera located on this plane. The sub-aperture images are simply the images of these pinhole cameras and thus 2D slices along the plane Π. Figure 2.6a shows the sub-aperture images extracted from a Lytro camera. Here, each of the images shows the same scene from a slightly different perspective and hence from different coordinates (u, v). Since each image is captured through only a subsection of the complete aperture, each image by itself has a very large depth of field. The set of sub-aperture images represent the complete 4D light field. This set can be used for disparity estimation and to synthesize a single full aperture image focused at different focal planes.

Figure 2.6b exemplary shows two EPIs as well as the center sub-aperture image. The EPIs represent slices along the (u, s) and (v, t) domains of the 2PP, respectively. As one can see from the figure, a single point in the 3D space corresponds to a straight line of constant color in the

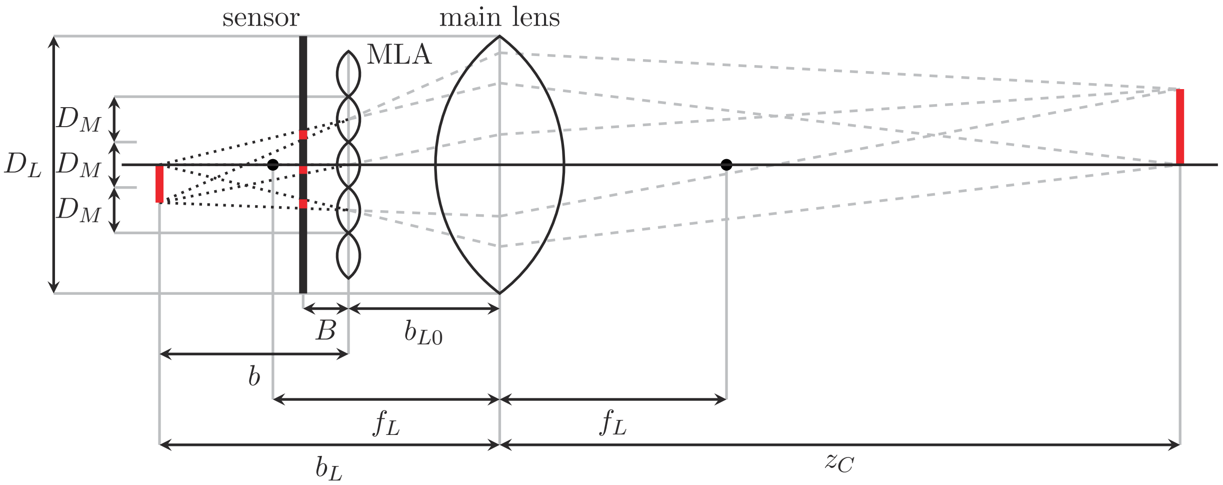


Figure 2.7: Cross section of a focused plenoptic camera in the Galilean mode. Main lens creates a virtual image of a real object behind the sensor. The virtual image is captured in multiple micro images on the sensor.

EPI. The distance of this point is encoded in the slope of the line. Let us consider a point lying on the plane Π, for instance. All rays corresponding to this point have the same coordinates s and t, respectively, while u and v can be swept across the entire range. Therefore, a point lying on the plane Π will result in a line of constant color which is perpendicular to the s and t axis, respectively. A point lying between the two planes Ω and Π will result in a negative slope along the u and v axis respectively, while a point behind the plane Π (referred to Ω) will result in a positive slope.

Considering this depth encoding in the 2PP, disparity estimation from the light field can be considered as the estimation of the slope of lines of constant color through the EPI (Wanner and Goldlücke [2014]), while image refocusing can be performed by shearing the EPIs and integrating along the (u, v) domains.

2.4 Depth Estimation for Plenoptic Cameras

This thesis deals with light fields captured by focused plenoptic cameras. Therefore, the idea of depth estimation is discussed exclusively for this type of camera. Depth estimation from the image of an unfocused plenoptic camera can be considered either as finding pixel correspondences in the set of sub-aperture images (e.g. Jeon et al. [2015]) or as estimating the slope of straight lines of constant color in the EPIs (e.g. Wanner and Goldlücke [2012]). The first task basically can be solved by standard stereo matching algorithms since the sub-aperture images form a multiple view stereo configuration. However, disparities have to be estimated with sub-pixel accuracy due to the small stereo baselines.

As discussed before, the light fields captured by focused plenoptic cameras do not conform to a uniformly sampled 2PP. Hence, sub-aperture images can not be extracted directly. Instead, the task of depth estimation is solved by finding pixel correspondences directly in the recorded micro images. In the following, the concept of depth estimation is described for the Galilean mode. However, it can be applied in a similar way to the Keplerian mode.

Figure 2.7 shows again a complete cross section of a focused plenoptic camera in the Galilean mode. The main lens produces a virtual image of an object, which is located at a distance z_c in front of the main lens, at the distance b_L behind the main lens. The relationship between image distance b_L and object distance z_C is defined by the thin lens equation:

$$\frac{1}{f_L} = \frac{1}{z_C} + \frac{1}{b_L}. \tag{2.7}$$

In general, a virtual image point is projected to multiple micro images. This can be guaranteed by placing the MLA with respect to the main lens' focal length f_L ($\min(b_L) = f_L$). Since the minimum image distance $\min(b_L)$ is equivalent to the main lens focal length f_L, the parameter b_{L0} must be chosen in such a way that an image point at distance f_L to the main lens is still visible in multiple micro images. Perwaß and Wietzke [2012] discuss this problem extensively for different MLA arrangements.

Figure 2.7 shows how the virtual image is projected by the three middle micro lenses onto different pixels of the sensor. If it is known which pixels on the sensor correspond to the same virtual image point, the distance b between MLA and virtual image can be calculated by triangulation.

To derive how the distance b can be calculated, Figure 2.8 shows an example of the triangulation for a virtual image point, based on the corresponding points in two micro images. In Figure 2.8 μ_i (for $i \in \{1,2\}$) define the distances of the points in the micro images with respect to the principal points of their micro images. Similarly, d_i (for $i \in \{1,2\}$) define the distances of the respective principal points to the orthogonal projection of the virtual image point on the MLA. All distances μ_i, as well as d_i, are defined as signed values. Distances with an upwards pointing arrow in Figure 2.8 have positive values and those with a downward pointing arrow have negative values. Triangles which have equal angles are similar and therefore the following relations hold:

$$\frac{\mu_i}{B} = \frac{d_i}{b} \qquad \longrightarrow \qquad \mu_i = \frac{d_i \cdot B}{b} \qquad \text{for} \qquad i \in \{1,2\}. \tag{2.8}$$

The baseline distance $\varDelta c_{ML}$ between the two micro lenses can be calculated as given in eq. (2.9).

$$\varDelta c_{ML} = d_2 - d_1 \tag{2.9}$$

The disparity μ of the virtual image point is defined as the difference between μ_2 and μ_1, and the relation given in eq. (2.10) is obtained.

$$\mu = \mu_2 - \mu_1 = \frac{(d_2 - d_1) \cdot B}{b} = \frac{\varDelta c_{ML} \cdot B}{b} \tag{2.10}$$

After rearranging eq. (2.10), the distance b between a virtual image point and the MLA can be described as a function of the baseline distance $\varDelta c_{ML}$, the distance B between MLA and sensor, and the disparity μ, as given in eq. (2.11).

$$b = \frac{\varDelta c_{ML} \cdot B}{\mu} \tag{2.11}$$

The number of micro images in which a virtual image point will appear depends on its distance b to the MLA. Thus, the length of the longest baseline $\varDelta c_{ML}$, which can be used for triangulation, changes. It can be defined as a multiple of the micro lens diameter $\varDelta c_{ML} = \kappa \cdot D_M$ ($\kappa \geq 1$). Here, κ is not necessarily an integer, due to e.g. the 2D arrangement of the micro lenses in the MLA. Instead, for a hexagonally arranged MLA, which is the case for most plenoptic cameras, the first 10 values of κ are as follows: 1.00, 1.73, 2.00, 2.65, 3.00, 3.46, 3.61, 4.00, 4.36, 4.58.

The baseline distance $\varDelta c_{ML}$ and the disparity μ are both defined in pixels and can be measured from the recorded micro lens images, while the distance B between MLA and sensor is a metric

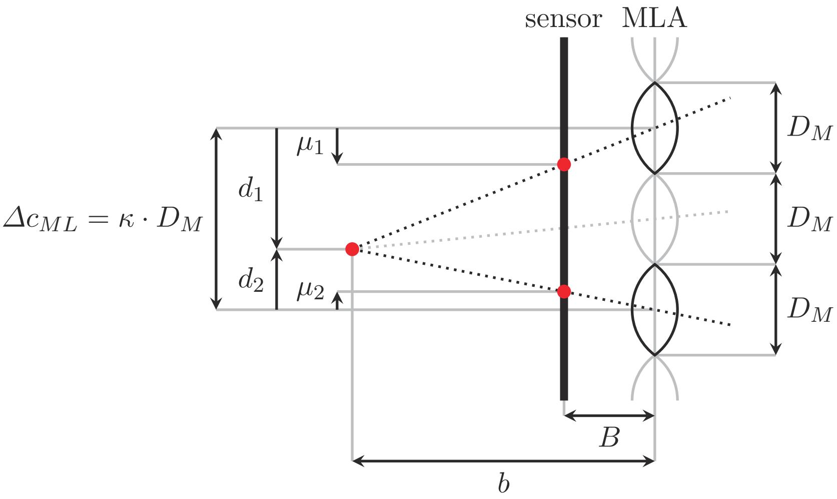


Figure 2.8: Depth estimation in a focused plenoptic camera based on the Galilean mode. The distance b between a virtual image point and the MLA can be calculated based on its projection in two or more micro images.

dimension which cannot be measured precisely. Thus, the distance b is estimated relatively to the distance B. This relative distance, which is free of any unit and is called *virtual depth*, will be denoted by v throughout this thesis:

$$v = \frac{b}{B} = \frac{\Delta c_{ML}}{\mu}. \tag{2.12}$$

To retrieve the real depth, i.e. the object distance z_C between an observed point and the camera, one has to estimate the relation between the virtual depth v and the image distance b_L (which relies on B and b_{L0}) in a calibration process. Then, one can use the thin lens equation (eq. (2.7)) to calculate the object distance z_C.

Once the virtual depth v has been estimated from the micro images, one is able to project the corresponding points into the image space and therefore to reconstruct a convex surface of the virtual image, including the corresponding intensities. It will be focused on that in more detail in Chapter 5.

2.5 Visual SLAM

The task of visual SLAM is about finding the position and orientation of a camera moving through 3D space while simultaneously building a 3D map of its surroundings. This section introduces basic components of SLAM such as transformations in 3D Euclidean space and its representations or nonlinear optimization, which the work presented in this dissertation builds upon.

2.5.1 Rigid Body and Similarity Transformation

The orientation and position of a camera in 3D space can be described by a rigid body transformation (or rigid body motion) which is also called a 3D special Euclidean transformation. Special means that reflections are excluded from the set of Euclidean transformations and therefore the respective transformation matrix has a determinate of +1. The set of all special Euclidean transformations in 3D is denoted by SE(3) and is also referred to as the special Euclidean group. The group of 3D rotational transformations is a subgroup of the special Euclidean group SE(3).

This group is also referred to as the special orthogonal group SO(3), as it is the group of orthogonal matrices with a positive determinate.

It is well known that with respect to homogeneous coordinates the rigid body transformation can be defined by a 4 × 4 matrix which will be denoted by $\boldsymbol{G} \in \mathrm{SE}(3)$:

$$\begin{aligned} \boldsymbol{x}' &= \boldsymbol{G} \cdot \boldsymbol{x}, \\ \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} &= \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad \text{with } \boldsymbol{R} \in \mathrm{SO}(3) \text{ and } \boldsymbol{t} \in \mathbb{R}^3. \end{aligned} \tag{2.13}$$

Here $\boldsymbol{R} \in \mathrm{SO}(3)$ describes a 3D rotation, which is also called a special orthogonal transformation. The vector $\boldsymbol{t}$ describes a 3D translation.

In a mathematical sense, the rigid body transformation is a transformation which preserves the inner product as well as the cross product. In other words, the rigid body transformation preserves distances as well as angles in the 3D space and thereby the complete 3D structure.

However, since in visual SLAM the camera orientation is estimated as a concatenation of multiple rigid body transformations, there will be scale drifts present in the reconstructed scene. To cover these scale drifts in the 3D map, the special Euclidean group can be generalized to the group of 3D similarity transformations Sim(3). While the 3D rigid body transformation has six degrees of freedom, one degree of freedom is added for the scale in the case of the 3D similarity transformation. Using homogeneous coordinates of 3D points, this transformation still can be described by a 4 × 4 matrix which will be denoted by $\boldsymbol{S} \in \mathrm{Sim}(3)$:

$$\begin{aligned} \boldsymbol{x}' &= \boldsymbol{S} \cdot \boldsymbol{x}, \\ \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} &= \begin{bmatrix} s\boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad \text{with } \boldsymbol{R} \in \mathrm{SO}(3),\ \boldsymbol{t} \in \mathbb{R}^3 \text{ and } s \in \mathbb{R}^+. \end{aligned} \tag{2.14}$$

In comparison to the rigid body transformation, the similarity transformation is simply extended by a positive scale s. $\boldsymbol{S} \in \mathrm{Sim}(3)$ is a transformation which preserves angles in the 3D space, while distances are scaled dependent on s.

For both matrices, $\boldsymbol{G}$ as well as $\boldsymbol{S}$, the fourth row is constant and therefore can be omitted. This leaves 12 real numbers to represent either a rigid body or a similarity transformation. Hence, the matrix representation is quite a redundant representation.

In this dissertation we chose to represent the transformations by the vector of the respective tangent space received from its Lie algebra. The tangent spaces give representations with (almost) no redundancy, since the transformations are defined in $\mathbb{R}^6$ and $\mathbb{R}^7$. 'almost' is added as a disclaimer, because the rotation angles are still not limited to a range of 2π. The way to formulate this tangent spaces is outlined in the following section.

2.5.2 Lie Groups and Lie Algebra

Lie groups are special types of manifolds. A Lie group is a group which is at the same time a smooth manifold. A smooth manifold defines a global differentiable structure, which can be approximated locally by a Eucildean space, as the surface of a sphere, for instance. The surface of a sphere can be approximated locally by a 2D Euclidean space, as maps are a local Euclidean

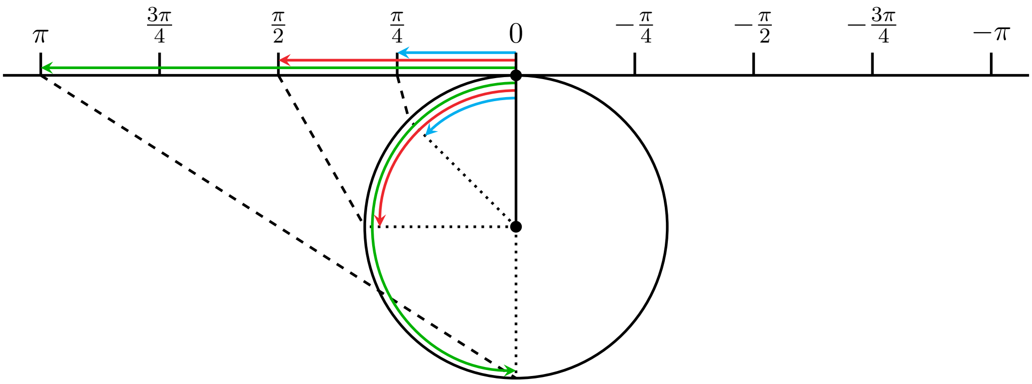


Figure 2.9: Schematic visualization of the tangent space of a smooth manifold, illustrated by the example of a circle. This circle defines the Lie group of 1D rotations. While the group of 1D rotations is a commutative Lie group, the Lie groups presented here are not commutative.

approximation of the globe. However, on a global scale, the sphere is a structure quite different from a Euclidean space. The sphere does not have the properties of a group with any operation and therefore is also not a Lie group. Figure 2.9 shows a simple example of a Lie group, a circle, which defines the group of one dimensional (1D) rotations. While the 1D rotation locally forms a 1D Euclidean space, it globally describes a circular hull. In contrast to the Lie groups introduced later, the group of 1D rotations is a commutative group which means that the order of a concatenation of transformations is interchangeable.

In fact, the transformations described above (SE(3) and Sim(3)) satisfy the properties of a Lie group together with the matrix multiplication as group operation. In the following, at first Lie groups will be discussed using the example of the special orthogonal group SO(3), after which we generalize our insight and extend it to apply to SE(3) and Sim(3).

While the Lie algebras for the groups of rotational transformations (SO(3)), rigid body transformations (SE(3)) and similarity transformations Sim(3) are introduced here, at this point, the respective derivations are skipped. The derivations in question are stated in different publications (e.g. Murray et al. [1994] (Chapter 2, Appendix A), Ma et al. [2004] (Chapter 2), Strasdat [2012] (Section 2.4, Appendix A). Therefore, the author refers readers who have a greater interest in Lie Manifolds and their applications in robotics and computer vision to the publications given above.

3D Rotational Transformation

It is well known that a rotational transformation (or special orthogonal transformation) can be described by a 3 × 3 matrix:

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \in \mathrm{SO}(3). \tag{2.15}$$

While this matrix has 9 entries it has only three degrees of freedom to fulfill the properties of a 3D rotation matrix. The matrix $\boldsymbol{R}$ must be an orthogonal matrix ($\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}$) and must have a positive determinate of one ($|\boldsymbol{R}| = +1$).

A 3D vector $\boldsymbol{w} \in \mathbb{R}^3$ is defined which describes the three degrees of freedom of the matrix $\boldsymbol{R} \in \mathrm{SO}(3)$. Furthermore, the $\hat{\ }$-operator is defined, which maps the vector $\boldsymbol{w}$ to a skew symmetric matrix:

$$\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \mapsto \widehat{\boldsymbol{w}} = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} \qquad \text{with } \boldsymbol{w} \in \mathbb{R}^3 \text{ and } \widehat{\boldsymbol{w}} \in \mathfrak{so}(3). \tag{2.16}$$

Here, the matrix $\widehat{\boldsymbol{w}}$ defines an element of the Lie algebra $\mathfrak{so}(3)$.

For the matrix $\widehat{\boldsymbol{w}}$ which has just three degrees of freedom a so-called *exponential map* is defined which maps the tangent space element $\widehat{\boldsymbol{w}} \in \mathfrak{so}(3)$ to the corresponding element of the Lie group SO(3).

$$\exp : \mathfrak{so}(3) \to \mathrm{SO}(3); \qquad \widehat{\boldsymbol{w}} \mapsto e^{\widehat{\boldsymbol{w}}} \tag{2.17}$$

It is not as obvious that the matrix exponential of a skew symmetric matrix $\widehat{\boldsymbol{w}}$ defines a rotation matrix. However, from the definition of the matrix exponential, one can easily prove that the matrix $\boldsymbol{R} = e^{\widehat{\boldsymbol{w}}}$ is actually a rotation matrix. Eq. (2.18) proves that $\boldsymbol{R}^{-1} = \boldsymbol{R}^T$ holds true and thus $\boldsymbol{R}$ defines an orthogonal matrix.

$$\boldsymbol{R}^{-1} = \left(e^{\widehat{\boldsymbol{w}}}\right)^{-1} = e^{-\widehat{\boldsymbol{w}}} = e^{\widehat{\boldsymbol{w}}^T} = \left(e^{\widehat{\boldsymbol{w}}}\right)^T \tag{2.18}$$

One can prove that the determinate of $\boldsymbol{R}$ is equal to $+1$ by Jacobi's formula as follows:

$$|e^{\widehat{\boldsymbol{w}}}| = e^{\mathrm{tr}(\widehat{\boldsymbol{w}})} = e^0 = 1, \tag{2.19}$$

where $\mathrm{tr}(\widehat{\boldsymbol{w}})$ defines the trace of the matrix $\widehat{\boldsymbol{w}}$.

Instead of solving the infinite series which defines the matrix exponential, the exponential map $\boldsymbol{R} = e^{(\widehat{\boldsymbol{w}})}$ can be solved based on Rodrigues' formula:

$$\boldsymbol{R} = e^{\widehat{\boldsymbol{w}}} = \boldsymbol{I} + \frac{\widehat{\boldsymbol{w}}}{\|\boldsymbol{w}\|} \sin(\|\boldsymbol{w}\|) + \frac{\widehat{\boldsymbol{w}}^2}{\|\boldsymbol{w}\|^2} \left(1 - \cos(\|\boldsymbol{w}\|)\right). \tag{2.20}$$

The logarithm which is the inverse operation of the exponential map is denoted by $\widehat{\boldsymbol{w}} = \log(\boldsymbol{R})$.

$$\log : \mathrm{SO}(3) \to \mathfrak{so}(3); \qquad e^{\widehat{\boldsymbol{w}}} \mapsto \widehat{\boldsymbol{w}} \tag{2.21}$$

The logarithm from SO(3) to the tangent space element $\mathfrak{so}(3)$ is defined as follows:

$$\|\boldsymbol{w}\| = \cos^{-1}\left(\frac{\mathrm{tr}(\boldsymbol{R}) - 1}{2}\right), \qquad \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} = \frac{1}{2\sin(\|\boldsymbol{w}\|)} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}. \tag{2.22}$$

Rodrigues' formula (eq. 2.20) immediately shows that there is no one-to-one mapping between $\widehat{\boldsymbol{w}} \in \mathfrak{so}(3)$ and $\boldsymbol{R} \in \mathrm{SO}(3)$ as the function is periodic in 2π with respect to the vector norm $\|\boldsymbol{w}\|$. However, each vector $\boldsymbol{w}$ defines only one rotation matrix $\boldsymbol{R}$. Furthermore, the group of rotational transformations, and therefore the exponential map, is not commutative. For two elements of the tangent space $\boldsymbol{w}_1$ and $\boldsymbol{w}_1 \in \mathfrak{so}(3)$ it is,

$$e^{\widehat{\boldsymbol{w}}_1} e^{\widehat{\boldsymbol{w}}_2} \neq e^{\widehat{\boldsymbol{w}}_2} e^{\widehat{\boldsymbol{w}}_1} \neq e^{\widehat{\boldsymbol{w}}_1 + \widehat{\boldsymbol{w}}_2}, \tag{2.23}$$

unless $\widehat{\boldsymbol{w}}_1 \widehat{\boldsymbol{w}}_2 = \widehat{\boldsymbol{w}}_2 \widehat{\boldsymbol{w}}_1$. Therefore, the Lie algebra for non-commutative Lie groups is completed by the introduction of the so-called *Lie bracket*, denoted by:

$$[\widehat{\boldsymbol{w}}_1, \widehat{\boldsymbol{w}}_2] = \widehat{\boldsymbol{w}}_1 \widehat{\boldsymbol{w}}_2 - \widehat{\boldsymbol{w}}_1 \widehat{\boldsymbol{w}}_2, \qquad \widehat{\boldsymbol{w}}_1, \widehat{\boldsymbol{w}}_2 \in \mathfrak{so}(3). \tag{2.24}$$

The Lie bracket can be considered as a measure for how strong the commutativity is violated by the Lie group. Recalling the example of 1D rotations for instance, shown in Figure 2.9, the tangent space element is only a scalar, defining the rotation angle ϕ. Since the multiplication with respect to a scalar is commutative, the respective Lie bracket will always be zeros. This implies that the Lie group of 1D rotations is commutative.

3D Rigid Body Transformation

For the Lie group of 3D rigid body transformations, in a similar way as for the special orthogonal group, a tangent space can be defined which is connected to the group by an exponential map.

In homogeneous coordinates the rigid body transformation is defined by a 4 × 4 matrix (see eq. (2.13)). One can define a map from a six dimensional vector $\boldsymbol{\xi}$ to the corresponding matrix $\widehat{\boldsymbol{\xi}}$ as follows:

$$\widehat{\boldsymbol{\xi}} = \begin{bmatrix} \widehat{\boldsymbol{w}} & \boldsymbol{v} \\ \boldsymbol{0} & 0 \end{bmatrix} \in \mathfrak{se}(3), \tag{2.25}$$

with

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{bmatrix} \in \mathbb{R}^6. \tag{2.26}$$

For the tangent space defined by the vector $\boldsymbol{\xi}$ again an exponential map can be defined, in order to map to the respective element in the Lie group:

$$\exp : \mathfrak{se}(3) \to \mathrm{SE}(3); \qquad \widehat{\boldsymbol{\xi}} \mapsto e^{\widehat{\boldsymbol{\xi}}} \tag{2.27}$$

This exponential is defined as follows:

$$\boldsymbol{G} = e^{\widehat{\boldsymbol{\xi}}} = \begin{bmatrix} e^{\widehat{\boldsymbol{w}}} & \frac{(\boldsymbol{I} - e^{\widehat{\boldsymbol{w}}})\widehat{\boldsymbol{w}}\boldsymbol{v} + \boldsymbol{w}\boldsymbol{w}^T\boldsymbol{v}}{\|\boldsymbol{w}\|} \\ \boldsymbol{0} & 1 \end{bmatrix}. \tag{2.28}$$

While the upper left part of the transformation matrix $\boldsymbol{G}$ simply defines a special orthogonal transformation, the upper right part defines the 3D translation vector $\boldsymbol{t} \in \mathbb{R}^3$:

$$\boldsymbol{t} = \frac{(\boldsymbol{I} - e^{\widehat{\boldsymbol{w}}})\widehat{\boldsymbol{w}}\boldsymbol{v} + \boldsymbol{w}\boldsymbol{w}^T\boldsymbol{v}}{\|\boldsymbol{w}\|}. \tag{2.29}$$

The logarithm in SE(3), defining the inverse of the exponential map, can be calculated straight forward from eqs. (2.22) and (2.29).

For the special Euclidean group the Lie bracket is defined analogously to the special orthogonal group:

$$\left[\widehat{\boldsymbol{\xi}}_1, \widehat{\boldsymbol{\xi}}_2\right] = \widehat{\boldsymbol{\xi}}_1\widehat{\boldsymbol{\xi}}_2 - \widehat{\boldsymbol{\xi}}_1\widehat{\boldsymbol{\xi}}_2, \qquad \widehat{\boldsymbol{\xi}}_1, \widehat{\boldsymbol{\xi}}_2 \in \mathfrak{se}(3). \tag{2.30}$$

3D Similarity Transformation

The generalization of the tangent space and the exponential map from SE(3) to Sim(3) is quite simple. The tangent space vector $\boldsymbol{\xi}_s$ is simply extended by one entry with respect to $\boldsymbol{\xi}$ as follows:

$$\boldsymbol{\xi}_s = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{w} \\ \rho \end{bmatrix} \in \mathbb{R}^7. \tag{2.31}$$

In contrast to the 3D rigid body transformation, for the 3D similarity transformation the rotation matrix $\boldsymbol{R}$ is multiplied by a positive scalar s unequal to zero (see eq. (2.14)). This is equivalent to an additive scalar in the exponent. Hence, the respective tangent space element $\widehat{\boldsymbol{\xi}}_s$ is defined by the following 4 × 4 matrix:

$$\widehat{\boldsymbol{\xi}}_s = \begin{bmatrix} \widehat{\boldsymbol{w}} + \boldsymbol{I}\rho & \boldsymbol{v} \\ \boldsymbol{0} & 0 \end{bmatrix} \in \mathfrak{sim}(3). \tag{2.32}$$

Finally, the exponential map for the Lie group of similarity transformations:

$$\exp : \mathfrak{sim}(3) \rightarrow \mathrm{Sim}(3); \qquad \widehat{\boldsymbol{\xi}}_s \mapsto e^{\widehat{\boldsymbol{\xi}}_s}, \tag{2.33}$$

is defined as follows:

$$\boldsymbol{S} = e^{\widehat{\boldsymbol{\xi}}_s} = \begin{bmatrix} e^{(\widehat{\boldsymbol{w}}+\boldsymbol{I}\rho)} & \frac{(\boldsymbol{I}-e^{\widehat{\boldsymbol{w}}})\widehat{\boldsymbol{w}}\boldsymbol{v}+\boldsymbol{w}\boldsymbol{w}^T\boldsymbol{v}}{\|\boldsymbol{w}\|} \\ \boldsymbol{0} & 1 \end{bmatrix} = \begin{bmatrix} e^{\rho}e^{\widehat{\boldsymbol{w}}} & \frac{(\boldsymbol{I}-e^{\widehat{\boldsymbol{w}}})\widehat{\boldsymbol{w}}\boldsymbol{v}+\boldsymbol{w}\boldsymbol{w}^T\boldsymbol{v}}{\|\boldsymbol{w}\|} \\ \boldsymbol{0} & 1 \end{bmatrix}. \tag{2.34}$$

Here, obviously the scale is defined by $s = e^{\rho}$.

For the sake of simplicity we will later omit the $\widehat{\ }$-operator and use the notations $\boldsymbol{w} \in \mathfrak{so}(3)$, $\boldsymbol{\xi} \in \mathfrak{se}(3)$, and $\boldsymbol{\xi}_s \in \mathfrak{sim}(3)$. Furthermore, the following notations will be used for the exponential maps and logarithms respectively:

$$\boldsymbol{R} = \exp_{\mathfrak{so}(3)}(\boldsymbol{w}) \qquad \boldsymbol{w} = \log_{\mathrm{SO}(3)}(\boldsymbol{R}) \qquad \text{with } \boldsymbol{w} \in \mathbb{R}^3 \text{ and } \boldsymbol{R} \in \mathrm{SO}(3) \tag{2.35}$$

$$\boldsymbol{G} = \exp_{\mathfrak{se}(3)}(\boldsymbol{\xi}) \qquad \boldsymbol{\xi} = \log_{\mathrm{SE}(3)}(\boldsymbol{G}) \qquad \text{with } \boldsymbol{\xi} \in \mathbb{R}^6 \text{ and } \boldsymbol{G} \in \mathrm{SE}(3) \tag{2.36}$$

$$\boldsymbol{S} = \exp_{\mathfrak{sim}(3)}(\boldsymbol{\xi}_s) \qquad \boldsymbol{\xi}_s = \log_{\mathrm{Sim}(3)}(\boldsymbol{S}) \qquad \text{with } \boldsymbol{\xi}_s \in \mathbb{R}^7 \text{ and } \boldsymbol{S} \in \mathrm{Sim}(3) \tag{2.37}$$

2.5.3 Nonlinear Optimization on Lie Manifolds

This section describes, by way of example, the Gauss-Newton optimization based on the Lie group of rigid body transformations SE(3). However, this optimization can be applied in the same way to any other Lie group.

In the interest of readability, the $\circ$-operator, as the concatenation of two rigid body transformation in the respective tangent space, is defined:

$$\circ : \mathfrak{se}(3) \times \mathfrak{se}(3) \rightarrow \mathfrak{se}(3). \tag{2.38}$$

The concatenation of the two elements $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ is defined as follows:

$$\boldsymbol{\xi} := \boldsymbol{\xi}_2 \circ \boldsymbol{\xi}_1 := \log_{\mathrm{SE}(3)}\left(\exp_{\mathfrak{se}(3)}(\boldsymbol{\xi}_2) \cdot \exp_{\mathfrak{se}(3)}(\boldsymbol{\xi}_1)\right). \tag{2.39}$$

As the Lie group SE(3) is not commutative, the concatenation of course is also not commutative.

Nonlinear optimizations rely on a Euclidean space parameterization of the function to be optimized with respect to the function argument. In this way the function can be optimized iteratively by adding an optimization increment $\delta\boldsymbol{x}^{(n)}$ to the current estimate $\boldsymbol{x}^{(n)}$:

$$\boldsymbol{x}^{(n+1)} = \delta\boldsymbol{x}^{(n)} + \boldsymbol{x}^{(n)}. \tag{2.40}$$

However, 3D rigid body transformations do not span a Euclidean space. Nevertheless, as discussed before, rigid body transformations form a smooth manifold and therefore can be approximated locally by a Euclidean space. Assuming that the optimization increment $\delta\boldsymbol{\xi}$ is small enough, optimization can be performed in a way similar to a Euclidean space.

The energy function $E(\boldsymbol{\xi})$ is generally defined as the sum over as set of squared residuals, as given in eq. (2.41). The set of residuals is denoted by the vector $\boldsymbol{r}(\boldsymbol{\xi})$.

$$E(\boldsymbol{\xi}) = \boldsymbol{r}(\boldsymbol{\xi})^T\boldsymbol{r}(\boldsymbol{\xi}) \tag{2.41}$$

The goal of the optimization is to find the vector $\boldsymbol{\xi}$ which minimizes the energy function $E(\boldsymbol{\xi})$.

Iterative optimization is performed by minimizing the second order Gauss-Newton approximation with respect to the tangent space:

$$\delta\boldsymbol{\xi}(n) = -(\boldsymbol{J}^T\boldsymbol{J})^{-1}\boldsymbol{J}^T\boldsymbol{r}(\boldsymbol{\xi}^{(n)}). \tag{2.42}$$

Here, $\boldsymbol{J}$ is the Jacobian of $\boldsymbol{r}(\boldsymbol{\xi}^{(n)})$ at the current estimate $\boldsymbol{\xi}^{(n)}$ with respect to the tangent space.

$$\boldsymbol{J} = \left.\frac{\partial\boldsymbol{r}(\boldsymbol{\epsilon}\circ\boldsymbol{\xi}^{(n)})}{\partial\boldsymbol{\epsilon}}\right|_{\boldsymbol{\epsilon}=0} \tag{2.43}$$

The optimization increment $\delta\boldsymbol{\xi}(n)$ is defined as a left-side increment to the current estimate $\boldsymbol{\xi}^{(n)}$. The error made by formulating the optimization of the tangent space is compensated by applying the increment using the concatenation operator $\circ$, which was defined previously:

$$\boldsymbol{\xi}^{(n+1)} = \delta\boldsymbol{\xi}^{(n)} \circ \boldsymbol{\xi}^{(n)}. \tag{2.44}$$

The optimization can be extended by a weighting scheme, as will be done in Section 7.5, for instance. Here, the energy function is extended by a weighting matrix $\boldsymbol{W}$ as follows:

$$E(\boldsymbol{\xi}) = \boldsymbol{r}(\boldsymbol{\xi})^T\boldsymbol{W}\boldsymbol{r}(\boldsymbol{\xi}), \tag{2.45}$$

where $\boldsymbol{W}$ is a diagonal matrix of positive entries defining the respective weights for the squared residuals. Furthermore, the definition of the update changes as follows:

$$\delta\boldsymbol{\xi}(n) = -(\boldsymbol{J}^T\boldsymbol{W}\boldsymbol{J})^{-1}\boldsymbol{J}^T\boldsymbol{W}\boldsymbol{r}(\boldsymbol{\xi}^{(n)}). \tag{2.46}$$

The matrix $\boldsymbol{W}$ generally does not have to be constant but is recalculated for each iteration. To formulate a robust estimator for instance, $\boldsymbol{W}$ is chosen such that high residuals are down weighted.

Levenberg-Marquardt Extension

In the implementations related to this thesis the Levenberg-Marquardt algorithm, which is a variation of Gauss-Newton optimization described above, is used. For the Levenberg-Marquardt algorithm the weighted Gauss-Newton approximation of the Hessian $\boldsymbol{N} = \boldsymbol{J}^T\boldsymbol{W}\boldsymbol{J}$ is modified. This modified matrix will be denoted by $\boldsymbol{N}'$. More precisely, each diagonal element of the matrix $\boldsymbol{N}$, denoted by $[\boldsymbol{N}]_{ii}$, is multiplied by $(1+\lambda)$. Hence, the matrix $\boldsymbol{N}'$ is defined as follows:

$$[\boldsymbol{N}']_{ij} = \begin{cases} (1+\lambda)\,[\boldsymbol{N}]_{ij} & \text{for } i = j, \\ [\boldsymbol{N}]_{ij} & \text{for } i \neq j. \end{cases} \tag{2.47}$$

In the update step given in eq. (2.46), the approximation of the Hessian $\boldsymbol{N}$ is replaced by its extension $\boldsymbol{N}'$. If the optimization increment $\delta\boldsymbol{\xi}$, obtained by performing the update step, leads to a reduction in the energy function $E(\boldsymbol{\xi})$, then the increment is $\delta\boldsymbol{\xi}$ accepted and λ is decreased (usually divided by a factor > 1). If the obtained $\delta\boldsymbol{\xi}$ leads to an increased value of the energy function, then the increment is rejected and λ is increased (multiplied by the same factor). The parameter λ is increased until a vector $\delta\boldsymbol{\xi}$ is found which leads to a decreased value of the energy function.

For $\lambda = 0$, the Levenberg-Marquardt algorithm behaves equivalently to the Gauss-Newton optimization and converges fast for an energy function which is close to a quadratic function. For a large λ the matrix $\boldsymbol{N}'$ is dominated by its diagonal elements. As the matrix $\boldsymbol{N}'$ nears the state of a diagonal matrix, its inverse does so, too. This can be interpreted as if each parameter of the vector $\boldsymbol{\xi}$ is optimized separately without considering the cross-correlation between the individual parameters. At the same time, for an increasing λ the determinate of the matrix $\boldsymbol{N}'$ increases as well. This results in a diminishing of the optimization increment $\delta\boldsymbol{\xi}$. Given that the Levenberg-Marquardt algorithm only accepts increments which lead to an improved energy function, or else requires that the step size is reduced, the algorithm always guarantees a decrease in the energy function and therefore always leads to a local minimum of that energy function.

More about the Levenberg-Marquardt optimization can be found e.g. in the book of Hartley and Zisserman [2003] (Appendix 6).

3 Related Work

Plenoptic camera based localization and mapping is a very young research field and as such, only few articles covering this topic have been published so far. In order to assess the advances that this dissertation contributes to the topics of plenoptic camera based VO and 3D scene reconstruction this section presents related work by other research groups. For a better overview, related work is categorized into three different research fields:

- plenoptic camera calibration
- depth estimation in light fields
- visual localization and mapping

After the related work the main contributions of this thesis are summarized.

3.1 Plenoptic Camera Calibration

Finding an appropriate model for the plenoptic camera is crucial for the success of scale aware 3D scene reconstruction. In this section camera models and calibration approaches which are known to date are discussed.

Work in the field of plenoptic camera calibration can be divided into three categories. In the first category, approaches are presented which do not perform a full geometric calibration of plenoptic cameras, but instead perform only a partial calibration to obtain optimized resampled light fields (Section 3.1.1). The second category contains calibration approaches for unfocused plenoptic cameras (Section 3.1.2), while the third category contains those for focused plenoptic cameras (Section 3.1.3).

Since the models for unfocused and focused plenoptic cameras are quite similar, some of the presented geometric calibration approaches apply for both concepts. Nevertheless, they are generally only tested on either of both cameras.

3.1.1 MLA Calibration & Light Field Resampling

The method presented by Ng and Hanrahan [2006] corrects the aberration of a real lens. Thereby it enhances the imaging quality by re-sorting the sampled light rays captured in a plenoptic camera but is strictly speaking not a calibration approach.

Yu et al. [2012] propose a color demosaicing approach for plenoptic cameras. Instead of performing demosaicing directly in the recorded raw image, it is performed after the processes of resampling and image synthesis. In this way, higher frequency components in the sampled 4D light field are preserved, while aliasing artifacts are minimized.

A complete pipeline for the calibration of the MLA in a plenoptic camera is presented by Cho et al. [2013]. Their method is demonstrated based on the specific example of a Lytro camera (Lytro [2013], 1. generation). They propose a discrete Fourier transform (DFT) based method to estimate the micro lens pitch and the MLA orientation. Furthermore, the regular hexagonal grid of the MLA is estimated by Delauney triangulation between priorly detected micro lens centers. Aside from the calibration of the MLA, they also propose several methods to resample the 4D light field.

Thomason et al. [2014] describe another approach to calibrate the MLA of a plenoptic camera. They model the deviation between the centers of the micro images, obtained from a recorded white image, and the actual micro lens centers and a misalignment of the MLA.

In a more recent approach Hog et al. [2017] propose an image rendering pipeline for focused plenoptic cameras. This pipeline includes the calibration of the MLA. While Cho et al. [2013] calculate only the micro lens pitch from a Fourier domain representation of a white image, Hog et al. [2017] derive all MLA parameters from the DFT of the white image.

3.1.2 Unfocused Plenoptic Camera Calibration

The first complete mathematical model of a plenoptic camera was proposed by Dansereau et al. [2013]. Here, the MLA is modeled as a grid of pinholes, while the main lens is modeled as a thin lens. Distortion of the main lens is corrected in the decoded 4D light field. The camera model is estimated from a planar checkerboard pattern detected in the sub-aperture images.

While the calibration approach put forward by Dansereau et al. [2013] requires that sub-aperture images are extracted from the distorted raw image of the plenoptic camera, Bok et al. [2014, 2017] define a distortion model directly in the micro images and resample the 4D light field from the rectified raw image. To overcome the problem of feature point detection in the small micro images, rather than detecting point features, they detect line features of a planar checkerboard directly in the micro images.

Both Dansereau et al. [2013] and Bok et al. [2014, 2017] estimate the intrinsic plenoptic camera parameters as well es the extrinsic orientation with respect to the planar checkerboard in a non-linear optimization approach.

3.1.3 Focused Plenoptic Camera Calibration

A first method for the calibration of focused plenoptic cameras was presented by Johannsen et al. [2013]. In contrast to the methods listed in Section 3.1.2, this calibration approach does not work directly on recorded raw images or 4D light fields. Instead, the camera model is estimated based on a synthesized view of the virtual main lens image (totally focused image) in combination with a virtual depth map received from disparities estimated in the micro images. Similar to the approaches described in Section 3.1.2, the main lens is modeled as a thin lens. In addition to the intrinsic camera model, imperfections of the main lens are corrected by a distortion model. Due to the fact that disparities are estimated based on the distorted raw image, a depth distortion model consisting of five coefficients is defined in addition to lateral distortions.

On the basis of the work from Johannsen et al. [2013], Heinze et al. [2015, 2016] propose an improved calibration approach. Here, the camera model is extended by modeling the effect of a plenoptic sensor (MLA and image sensor) which is tilted with respect to the main lens. However, the effect of a misaligned MLA with respect to the image sensor, as described by Thomason et al. [2014], is neglected. The depth distortion model is reduced to a three-coefficient model. This three-coefficient model shows an improved accuracy when compared to the five-coefficient model of Johannsen et al. [2013]. The calibration approach proposed by Heinze et al. [2015, 2016] is designed specifically for Raytrix cameras (Raytrix GmbH [2013]). A distinct feature of plenoptic cameras from the manufacturer Raytrix is a MLA which consists of three interlaced types of micro lenses with different focal lengths to extend the DOF of the camera. For each type of micro lens Heinze et al. [2015, 2016] estimate a different distance with respect to the image sensor.

In both approaches, Johannsen et al. [2013] and Heinze et al. [2015, 2016], a planar target is used for calibration. This target is recorded from multiple perspectives to estimate the intrinsic and extrinsic parameters.

Strobl and Lingenauber [2016] present a stepwise calibration approach for focused plenoptic cameras which is based on a planar checkerboard target. In their approach, the first step is to estimate a lateral camera model based on the synthesized totally focused virtual image of the plenoptic camera and disregarding the estimated depth map. In a second step, a conversion function from estimated virtual depths to metric object distances is estimated.

Zhang et al. [2016] propose a novel model and calibration approach for focused plenoptic cameras, consisting of 10 intrinsic parameters. In addition to the standard extrinsic parameters, they model a misalignment of the MLA with respect to the image sensor in a similar way as proposed by Thomason et al. [2014]. A misalignment between MLA and main lens is not considered. Similar to Bok et al. [2014], they consider the aberration between micro image centers, estimated from a recorded white image, and the real micro lens centers. Zhang et al. [2016] estimate the complete model, including the micro lens centers, based on a recorded white image and recordings of two parallel planes which provide depth and scale priors. However, their model does not consider lens distortion. Aside from that, the extrinsic orientation is not estimated during optimization. Therefore, their approach does not allow for the combination of information of multiple images captured from different views for calibration.

3.2 Depth Estimation in Light Fields

In the last years extensive research on light-field-based depth estimation has been published. This section gives an overview of published algorithms which work on images recorded by a plenoptic cameras and other light field data.

A first algorithm for depth estimation based on the recordings of a plenoptic camera was published more than 20 years ago by Adelson and Wang [1992]. They assembled one of the first plenoptic cameras and proposed an algorithm for disparity estimation in the sub-aperture images received from the camera.

Bishop and Favaro [2009, 2012] show that for plenoptic cameras the low sampling frequency in the spatial domain results in aliasing effects which generate artifacts in the estimated disparity map. Therefore, they propose an iterative depth estimation algorithm which performs filtering to remove those aliasing artifacts. One drawback of this method is that depth estimation is performed on low resolution sub-aperture images. Therefore, the spatial resolution of the depth map is limited by the number of micro lenses in the MLA. Bishop and Favaro [2011] developed their algorithm further to overcome the limitation of low spatial resolution. This is achieved by rendering full-resolution views from the recorded micro images.

Wanner and Goldlücke [2012, 2014] propose a method for globally consistent depth estimation in 4D light fields. Their algorithm estimates disparities by finding line slopes in the EPIs. The algorithm is tested on simulated data, light fields recorded from camera arrays as well as those reconstructed from images of a focused plenoptic camera (Wanner et al. [2011]).

Kim et al. [2013] propose an efficient method for 3D scene reconstruction from high spatio-angular resolution Giga-ray light fields. Similar to Wanner and Goldlücke [2014] their algorithm works on EPIs. The algorithm is broken down in to several steps to efficiently process the large amount of data. They start with estimating the depth around object boundaries. Homogeneous regions are filled in a fine-to-coarse (rather than coarse-to-fine) approach.

Heber et al. [2013] describe a method for high quality depth map calculation from a set of sub-aperture images extracted from the recording of a plenoptic camera. The sub-aperture images are extracted in such a way that the view points are arranged on circles around a center view. Thus, depth map calculation is solved by estimating the virtual rotation of a scene point over the set of sub-aperture images.

In another paper, Heber and Pock [2014] show that a multiple view stereo problem can be formulated as a principal component analysis (PCA).

Yu et al. [2013] propose a light field triangulation and stereo matching approach. They analyze 3D line segments in the light field's ray space and perform Delaunay triangulation afterwards. The extracted line segments are used to perform light field based stereo matching using a line-assisted graph cut algorithm.

Chen et al. [2014] published a light field stereo matching algorithm which is based on the surface camera (Scam) parametrization, introduced by Yu et al. [2004], to handle significant occlusions. They use a bilateral statistics concept to model the probability of occlusions. Furthermore, they show that bilateral Scam analysis can be used to distinguish between on-surface and free space, textured and non-textured, as well as Lambertian and Specular reflection. Another approach, which is able to detect and remove specular reflection, is presented by Tao et al. [2014], while Wang et al. [2015] propose an additional depth estimation method which explicitly models occlusions in the recorded scene.

Tosic and Berkner [2014] present a so-called scale-depth space by convolving a recorded light field with a special kernel. In the scale-depth space, regions of constant depth are represented by local extrema and are used to calculate dense depth maps.

Jeon et al. [2015] make use of the phase-shift theorem of the Fourier transform to interpolate the sub-aperture images and thereby obtain disparity estimates with sub-pixel accuracy.

Tao et al. [2013, 2015] and Lin et al. [2015] present light field based depth estimation algorithms which combine different types of cues. These include correspondence, focus, and shading information.

Yucer et al. [2016] propose an approach to calculate per-pixel depth, based on local gradient information and thereby they calculate accurate depth for thin features. Furthermore, using a two-sided photoconsistency measure, they are able to detect whether a pixel lies on a texture or silhouette edge and use the gained information to fill untextured regions.

Heber and Pock [2016] utilize convolutional neural networks (CNNs) to predict depth information for given light field data. The proposed method learns an end-to-end mapping between the 4D light field and a representation of the corresponding 4D depth field in terms of 2D hyperplane orientations. The obtained prediction is then further refined in a post processing step by applying a higher-order regularization.

Recently, Hog et al. [2017] published an algorithm which estimates depth directly from the raw data of a focused plenoptic camera. Disparities are estimated from a set of stereo images, called focal stack, extracted directly from the recorded micro images.

As the algorithm of Hog et al. [2017] is customized to raw images recorded by focused plenoptic cameras it is probably the one most closely related to the work presented in this dissertation (Chapter 5).

Furthermore, the author refers interested readers to the publication of Wu et al. [2017] which tries to offer a complete overview of light field image processing.

3.3 Visual Localization and Mapping

This section summarizes recent advances in visual SLAM and odometry estimation based on monocular (Section 3.3.1), stereo and RGB-D sensors (Section 3.3.2). Section 3.3.3 presents light field based VO and structure from motion (SfM) algorithms which are the most closely related to the work on plenoptic odometry (Chapter 7) presented in this dissertation.

SLAM and VO algorithms can be categorized in different groups with regards to the sensor used, the type of measurements fitted in the optimization approach, the structure of the estimated point cloud, and the form in which the camera orientation is represented.

With respect to the type of measurements, VO can be divided into indirect (or feature based) and direct methods. Indirect methods perform an intermediate step to extract a certain type of geometric primitives (i.e. points, line segments, vectors), based on which a cost function is defined. This cost function is minimized to find the optimal model parameters. Direct methods, by contrast, skip this intermediate step and formulate the cost function directly for the measurements supplied by the sensor (i.e. intensities or depth measurements).

Regarding the structure of the point cloud, the algorithms can be divided into sparse and dense (or semi-dense) approaches. While sparse methods calculate only a sparse and unstructured set of uncorrelated 3D points, dense and semi-dense methods provide a dense or partly dense reconstruction. Neighboring points are connected to each other in the sense of a probabilistic smoothness term.

Regarding the way the camera motion or the camera position is represented, there are essentially two different concepts, which are filter-based and keyframe-based methods. Filter-based approaches estimate a densely sampled or even continuous representation of the camera trajectory over time. However, these filter-based approaches drop past measurements. Keyframe-based methods, by contrast, reduce the number of samples over time by selecting a subset of frames (keyframes), for which the camera orientations and 3D structure are optimized. Past keyframes are kept, which allows for optimization of the complete map in terms of global consistency.

The task of vision-based navigation has been studied for almost fifty year now. However, approaches which estimated all six degrees of freedom of camera motion first emerged around the turn of the millennium.

3.3.1 Monocular Algorithms

One of the main challenges for monocular VO is that depth can not be gained from a single frame. Therefore, in an initial stage, monocular VO results in an ill-conditioned optimization problem. While indirect methods generally try to find an initial solution by estimating the essential matrix between subsequent frames, direct methods generally start from a random depth map and rely on only little motion between initial frames to assure convergence. Here, existing algorithms will be divided into these two categories.

Indirect Methods

One of the first algorithms which incrementally estimates camera motion and 3D structure of a predefined set of feature points was proposed by Chiuso et al. [2002]. Feature points are tracked from frame to frame, while the algorithm tries to handle occlusion and different scale factors of the features.

Davison et al. [2007] propose an algorithm called MonoSLAM, which is one of the first fully automated visual SLAM algorithms. Features are tracked based on patch correlation and the algorithm is able to perform local loop closures.

A significant breakthrough in monocular SLAM was achieved by Klein and Murray [2007, 2008] with their algorithm called PTAM (Parallel Tracking and Mapping). They split the SLAM problem into two separate tasks. On one hand, the current camera pose is tracked based on a given scene model for every single frame. On the other hand, the 3D map and camera poses are optimized based on a carefully selected set of keyframes. In this way, map optimization does not have to be performed at camera frame rate and therefore the number of landmarks used for mapping can be increased. Later PTAM was shown to be able to run on modern cell phones (Klein and Murray [2009]).

While most indirect SLAM and VO algorithms are based on corner points, which in turn are matched based on feature descriptors (e.g. SIFT or SURF), Eade and Drummond [2009] propose a monocular SLAM approach which makes use of edge landmarks instead of corners. The defined landmarks, called edgelets, are still point features which are well defined only along one dimension.

Accumulated scale drifts are a problem of monocular SLAM algorithms. They may lead to convergence issues at loop closures. Strasdat et al. [2010b] propose a pose graph optimization scheme, which is based on the Lie group of Sim(3). After a loop closure detection, the complete pose graph is optimized, including the scale of each keyframe.

Furthermore, Strasdat et al. [2010a, 2012] perform extensive evaluations based on Monte Carlo simulations to compare extended Kalman filter (EKF) based VO with keyframe-based approaches, which perform local pose graph optimization. They come to the conclusion that keyframe-based approaches always outperform filter-based approaches while requiring the same computational effort.

ORB-SLAM, presented by Mur-Artal et al. [2015], can be considered today's state-of-the-art in feature-based real-time SLAM. They present a full SLAM framework which performs tracking and keyframe-based mapping on extracted ORB-features (Rublee et al. [2011]). Relocalization and loop closure detection is performed in a bag-of-words approach, using a vocabulary learned from a large set of images (Galvez-López and Tardós [2012]).

While indirect dense approaches are still rather rare, a few algorithms considering this approach have been published in recent years.

Valgaerts et al. [2012] lay out how the estimation of the fundamental matrix, mostly used as initialization in indirect SfM and SLAM, can benefit from dense optical flow estimates. Furthermore, they propose a new model to recover the fundamental matrix and dense optical flow simultaneously.

Instead of using only point features, Concha and Civera [2014] propose a monocular SLAM approach based on superpixels, which are features consisting of image regions with homogeneous texture and are assumed to be planar surfaces in the scene.

Ranftl et al. [2016] propose a dense depth estimation approach from a monocular moving camera. Their approach estimates a dense depth for static parts of the scene and detects moving objects based on motion segmentation in the optical flow field received from two consecutive images.

Direct Methods

Traditionally sparse indirect approaches were developed for odometry estimation to reduce the amount of data. Today's computers are capable of handling large optimization problems by implementing them in a parallel structure. This makes it possible to skip the time-consuming feature extraction part and perform tracking and mapping directly on pixel intensities.

Newcombe et al. [2011] propose a first algorithm called DTAM (Dense Tracking and Mapping). They estimate a dense 3D map of the scene and the camera motion based on a moving monocular camera by minimizing the photometric error between consecutive frames. Using modern graphics processing units (GPUs) their approach is able to run in real-time.

Engel et al. [2013] propose a semi-dense formulation which ignores homogeneous regions and considers only image regions with a sufficiently high intensity gradient. This reduction of the amount of data allows direct VO approaches to run in real time on standard central processing units (CPUs) and even on modern smart phones (Schöps et al. [2014]). Engel et al. [2014] developed their approach further and propose a full SLAM framework, based on this semi-dense formulation, called LSD-SLAM. LSD-SLAM performs relocalization, loop closure detection and Sim(3) pose graph optimization without any feature extraction.

Beside dense and semi-dense formulations, there also exist some sparse direct formulations. A first sparse and semi-direct approach for structure from motion is proposed by Jin et al. [2003]. Semi-direct means that the algorithm does not completely relinquish the extraction of geometric features. Instead, it performs a combined optimization based on geometric feature points and image patches.

Forster et al. [2014, 2017] propose another semi-direct approach. While they use direct methods for tracking of new frames, structure and motion optimization is performed based on geometric feature points. They successfully incorporate point features as well as edgelets (Eade and Drummond [2009]) into their algorithms and are able to track and map features in regions where traditional indirect methods would fail. Gomez-Ojeda et al. [2016] extend the algorithm by Forster et al. [2014] to line segment based features.

In a recent publication, Engel et al. [2018] propose the first completely direct sparse approach which is able to run in real-time on a standard CPU. While previous sparse and direct approaches still relied on geometric features in the optimization back-end, Engel et al. [2018] perform pose optimization and sparse scene structure estimation in a completely direct approach. The optimization is performed over a sliding window of keyframes without building a pose graph.

3.3.2 RGB-D Sensor and Stereo Camera based Algorithms

For RGB-D sensors and stereo cameras the task of VO is significantly simplified in comparison to monocular approaches. Since absolute depth is received from a static recording, these systems do not have to deal with scale drifts. Furthermore, initialization becomes much easier since scene structure is available from the first frame. Here, publications are grouped into RGB-D sensor and stereo camera based approaches.

RGB-D Sensor based VO and SLAM

Extensive research on RGB-D SLAM and VO began with the release of the Microsoft Kinect sensor in 2010. The Kinect offers a low-cost sensor which is able to capture dense RGB-D data at video frame rate. The SLAM methods for RGB-D sensors are mostly a combination of direct and indirect approaches. Hence, this section offers a more or less chronological list of algorithms.

Henry et al. [2010, 2012] present one of the first algorithms for RGB-D based tracking in combination with dense 3D mapping. They use a combination of visual features and 3D structures received from the depth sensor to perform frame alignment in a modified Iterative Closest Point (ICP) algorithm (Chen and Medioni [1992]; Besl and McKay [1992]). Furthermore, loop closures are detected based on visual and depth information.

Huang et al. [2011] extend the algorithm of Henry et al. [2010] with a feature-based VO front-end. While the VO algorithm is able run at a high frame rate, accurate mapping can be performed at a much lower frame rate.

Pomerleau et al. [2011] present another ICP-based algorithm. In contrast to Henry et al. [2010], they do not rely on any visual features.

Osteen et al. [2012] present an odometry estimation algorithm for RGB-D cameras which does not extract any feature points. They estimate an initial pure rotational motion by the correlation of extended Gaussian images (EGIs) (Horn [1984]), created from the local normal estimates of the point cloud. Afterwards, the initial estimate is refined and translation is added in an ICP approach.

In a manner similar to Henry et al. [2010], Dryanovski et al. [2012] perform an initial estimation based on features extracted from the images and in a second step, refined the estimate using ICP. However, in contrast to Henry et al. [2010], they use edge features instead of points.

Dryanovski et al. [2013] present another feature- and ICP-based VO algorithm. In contrast to previous approaches, rather than performing alignment with respect to a reference frame, alignment is performed referring to a global model dataset of 3D features. While this model is gradually growing, it is upper bounded by the number of points.

Hu et al. [2012] propose an algorithm which switches between pure intensity-based bundle adjustment and optimization based on full RGB-D data. Two separate local maps are built which are joined afterwards.

The famous KinectFusion algorithm by Izadi et al. [2011] also performs six degrees of freedom pose estimation based on the ICP algorithm. The depth maps of the single camera frames are incorporated into a volumetric 3D model, which is based on truncated signed distance functions (Curless and Levoy [1996]). There exist several extensions to KinecFusion. Kintinuous (Whelan et al. [2012]), for instance, allows the algorithm to perform on a large scale by shifting the volumetric space. Furthermore, the tracking approach is extended to use color information in combination with depth (Whelan et al. [2013b]). Whelan et al. [2013a] add a pose graph for global pose optimization.

Yet another extension to the work of Whelan et al. [2013a] is ElasticFusion (Whelan et al. [2015]). Here, the pose graph is omitted while global optimization is achieved by the fusion of windowed surface elements.

Endres et al. [2014] present a feature-based RGB-D-SLAM algorithm. In a front-end stage the egomotion of the camera is estimated based on features extracted from the RGB image and validated by the point cloud received from the depth map. Based on these estimates, a pose graph is built and optimized.

Steinbrücker et al. [2011] propose an odometry estimation method where the intensity image of one frame is warped into the next frame, based on the corresponding depth map. Then the relative pose between the two frames is optimized by minimizing the photometric error between the warped and the real image. Kerl et al. [2013b] and Klose et al. [2013] propose quite similar approaches which extend the method of Steinbrücker et al. [2011] by the robust Huber-norm (Huber [1964]). Furthermore, Kerl et al. [2013a] combine the odometry estimation with a keyframe-based pose graph to obtain globally consistent maps.

Bylow et al. [2013] present a tracking approach which is based on the alignment of signed distance functions (Curless and Levoy [1996]).

Kerl et al. [2015] model the effect of a rolling shutter for RGB-D cameras. Furthermore, instead of representing the camera trajectory as sampled in the time domain, the algorithm estimates a continuous trajectory representation using cubic B-splines (Lovegrove et al. [2013]).

Stereo Camera based VO and SLAM

Monocular SLAM algorithms can generally be directly extended to stereo based approaches and therefore both types are very closely related to each other.

Mei et al. [2011] present a feature-based large-scale relative SLAM system based on a binocular stereo camera. By using a continuous relative representation of the trajectory and map they are able to perform robust tracking and mapping in constant time.

Engel et al. [2015] describe a stereo extension of their previously published monocular LSD-SLAM algorithm. Static stereo observations from the images of both cameras are combined with motion stereo. Similar to LSD-SLAM, its stereo version establishes semi-dense depth maps as well as a keyframe pose graph. However, optimization is performed with respect to SE(3) instead of Sim(3), as no accumulated scale drifts occur.

Mur-Artal and Tardós [2017] present a generalization of the ORB-SLAM algorithm (Mur-Artal et al. [2015]) for stereo camera systems and RGB-D sensors.

While all algorithms listed above are only based on camera data, there are other algorithms which fuse visual information with additional sensor data such as those from inertial measurement units (IMUs): Mourikis and Roumeliotis [2007]; Li and Mourikis [2013]; Leutenegger et al. [2015]; Usenko et al. [2016].

3.3.3 Light Field based Localization and Mapping

As mentioned before, there have been no plenoptic camera based VO and SLAM algorithms published thus far. However, this section presents the algorithms most closely related to the method proposed in this thesis. These are either VO algorithms, which are based on any kind of light field representations, as well as SfM and 3D reconstruction approaches based on plenoptic images and other light fields.

Dansereau et al. [2011] published a paper called Plenoptic Flow. They describe three different closed-form solutions for six degrees of freedom odometry estimation based on 4D light field representations. Nevertheless, the proposed methods do not use multiple light field frames to improve scene structure. Furthermore, the methods have been tested only on light fields captured from camera arrays and simulated data.

Similar to Dansereau et al. [2011], Dong et al. [2013] propose a light field based VO system to estimate the trajectory of a moving robot. Their system has been tested only on light fields received from a 3×3 camera array and is likely to fail on plenoptic cameras which suffer from small stereo baselines.

Johannsen et al. [2015] propose a ray-feature based SfM approach for 4D light field representation. They formulate a linear solution for the rigid body transformation from one 4D light field into the other, based on ray correspondences. This initial solution is refined in a non-linear optimization process.

Skinner and Johnson-Roberson [2016] present a method for underwater 3D reconstruction based on a plenoptic camera. Instead of working on the light field data captured by the plenoptic camera, they apply an underwater light propagation model to the virtual depth maps calculated with standard software and build a 3D model from a set of depth maps.

Recently Zhang et al. [2017] published a work called “the light field 3D scanner”. This work presents a feature-based SfM approach working on 4D light field representations. Similar to Johannsen et al. [2015] the method is able to fuse a large light field of a static scene from a collection of light field images. Their approach relies on the well known 2PP of the light field, which firstly has to be extracted from the raw data of a plenoptic camera.

3.4 Own Contribution

Plenoptic Camera Model

In this thesis a new mathematical model for a focused plenoptic camera is proposed (Section 4.1). In contrast to all previous models, this model shows that a focused plenoptic camera actually forms a virtual camera array in the object space where each micro lens in the MLA represents a single pinhole camera.

Based on the new plenoptic camera model a multiple view geometry (MVG) for plenoptic cameras is established (Section 4.2). This MVG allows to define epipolar constraints between micro images of different light field frames.

Furthermore, based on the model, one can show that focused plenoptic cameras are generally superior to unfocused plenoptic cameras with respect to single frame depth estimation (Section 4.3.2).

Light Field based Depth Estimation

Even though there already exists a variety of light field based depth estimation algorithms, a new algorithm was developed throughout the course of this doctoral thesis. Most of the algorithms listed in Section 3.2 perform dense and consistent depth estimation from some 4D light field representation. These approaches are computationally complex and must calculate EPIs or sub-aperture images in advance. Thus, these approaches supply highly accurate and consistent depth information but are impractical for real-time processing.

This dissertation presents an efficient depth estimation approach for focused plenoptic cameras (Chapter 5), which is able to run in real-time. In contrast to other algorithms, it works directly on the recorded raw data. It calculates semi-dense depth maps by working only on image regions with a sufficiently high intensity gradient.

The algorithm works incrementally using a graph of stereo baselines (Section. 5.2). New depth observations are incorporated in a probabilistic manner, where different stochastic error sources are taken into account. Furthermore, a probabilistic filtering approach is presented, which models the correlation between neighboring points (Section 5.6).

Due to its incremental nature, the algorithm can be extended to multiple frames of the plenoptic camera, as it is done in the proposed Direct Plenoptic Odometry (DPO) algorithm (Chapter 7).

Plenoptic Camera Calibration

In this dissertation the plenoptic camera is modeled at various levels of abstraction leading to plenoptic camera calibration approaches of various complexities (Chapter 6).

Starting from simple depth conversion functions (Section 6.1), we proceed to define a camera model based on the totally focused image and the virtual depth calculated from the raw data of the plenoptic camera (Section 6.2.1). Ultimately, we arrive at a complete model of a plenoptic camera which defines the projection from a 3D point in object space to multiple points in the micro images (Sections 6.2.2 and 6.2.3). In contrast to most previous models, here the fact is considered that the micro lens centers are actually not at the center of the micro images (obtained from a white image). Instead, the micro lenses in a plenoptic camera are squinting. Not considering the squinting micro lenses in the camera model results in a bias in the estimated object distance.

The methods presented in this theses perform plenoptic camera calibration based on a 3D calibration target which consists of unordered calibration points. While the 3D nature of our calibration target leads to a well-conditions optimization task, the method does not rely on any prior constraints, which is due to the reason that a set of unordered points is used for calibration. To estimate the camera parameters a full plenoptic bundle adjustment is formulated which optimizes the camera parameters as well as the 3D structure of the calibration target. (Sections 6.3).

Direct Plenoptic Odometry

A VO algorithm named Direct Plenoptic Odometry (DPO) is proposed, which combines the MVG for plenoptic cameras, the light field based depth estimation algorithm, and the intrinsic camera parameters received from the calibration (Chapter 7).

DPO is formulated as a keyframe-based algorithm which generates semi-dense probabilistic point clouds. New recorded light field frames are tracked based on direct image alignment with respect to the micro images (Section 7.5).

Depth estimates in the current keyframe are refined based on stereo correspondences within subsequent frames (Section 7.3.3). Correspondences are established on the basis of the introduced plenoptic MVG (Section. 4.2).

By working directly with the recorded micro images (raw image) one is able to find stereo correspondences using full sensor resolution instead of low-resolution sub-aperture images. This avoids aliasing effects due to undersampling in the spatial domain and results in significantly higher resolved depth maps.

Different approaches, e.g. lighting compensation (Section 7.5.3) and motion priors (Section 7.5.4), are implemented to improve the tracking robustness of the algorithm. Scale drifts are accounted for in a keyframe-based active scale optimization framework (Section 7.7).

To the best of our knowledge, DPO is the first VO algorithm for plenoptic cameras of its type, which works directly with the recorded raw data of a plenoptic camera and improves the static depth by plenoptic motion stereo.

Datasets

To perform plenoptic camera calibration, a customized 3D calibration target was assembled. Based on this target, calibration datasets for various main lens focal lengths were recorded. Furthermore, calibration datasets based on a standard checkerboard pattern were recorded to compare the proposed methods to state-of-the-art algorithms (Chapter 9).

To date, there are no public datasets available for plenoptic camera based VO. For this thesis a handheld platform was developed to record synchronized data from a plenoptic camera and a stereo camera system. Using the platform, a versatile dataset consisting of indoor and outdoor scenes was recorded. A ground truth for the sequences was obtained by detecting a large loop closure between the beginning and the end of a sequence (Chapter 10).

4 The Plenoptic Camera from a Mathematical Perspective

Light fields recorded by plenoptic cameras have been extensively studied in the past. These studies have provided us with insight about how a 4D light field representation can be extracted from the image captured by the camera. It is known, for instance, that from a plenoptic camera one can extract a set of sub-aperture images which depict the scene from slightly different perspectives. Using these sub-aperture images, one is able to estimate disparity maps representing the 3D structure of the recorded scene. However, if it is not known where the synthesized views are located in the real world and what intrinsic parameters these synthesized views rely on, one is not able to transform the disparity maps into metric depth maps or register them to the real world. But such information is needed if one wants to perform odometry estimation based on the images of a plenoptic camera.

This geometric relationship has been studied by Christopher Hahne (Hahne [2016]; Hahne et al. [2014]) for traditional, unfocused plenoptic cameras in his PhD thesis. However, the matter changes slightly for focused plenoptic cameras, because rather than merely representing a light ray with a certain incident angle, each pixel also represents a focused image point. Here, high resolution sub-aperture images or EPI representations cannot be extracted directly from the recorded raw data. Instead, disparity estimation must be performed in advance (Wanner et al. [2011]). Therefore, in the following, the geometrical relationship between the micro images recorded by a focused plenoptic camera and the real world is investigated. This leads to a new interpretation of a MLA based light field camera in which each micro lens acts as a single virtual pinhole camera. This interpretation allows to formulate a multiple view epipolar geometry for plenoptic cameras.

The new camera interpretation does not only allow to perform multiple view stereo vision based on a plenoptic camera, it also gives insight into the structure of the depth data received from the camera. Furthermore, it will demonstrated why, when performing 3D reconstruction based on plenoptic cameras, one should generally choose a focused plenoptic camera over a traditional unfocused one.

In the following mathematical models, the plenoptic camera is always considered a perfect optical system, where the main lens is an ideal thin lens and the MLA is a grid of pinholes. Imperfections in the optical system are completely neglected in this chapter and will be handled during the camera calibration presented in Chapter 6.

4.1 New Interpretation of a Plenoptic Camera

Traditionally the focused plenoptic camera is modeled in the following way. The plenoptic sensor (image sensor and MLA) is considered an array of pinhole cameras which observe the image created by the main lens. This model is visualized in Figure 4.1 for the Galilean mode of a

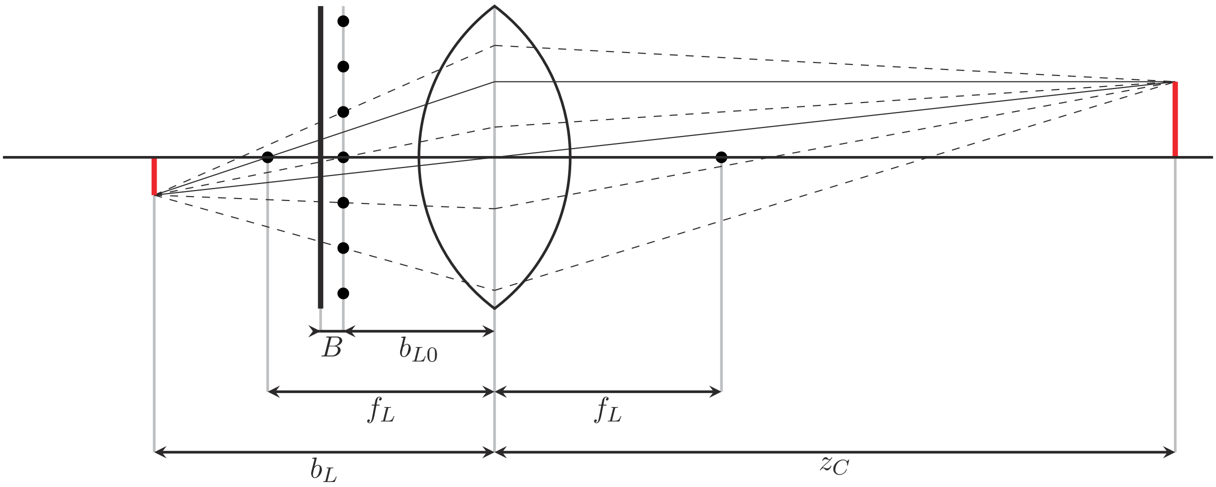


Figure 4.1: Imaging process for a focused plenoptic camera in the Galilean mode. The main lens forms a virtual image behind the sensor. This image is a warped representation of the object space. The virtual image is projected by the MLA on the sensor where multiple focused micro images are formed.

focused plenoptic camera. In the Galilean mode, the main lens image is formed behind the sensor and thus results in a virtual image which is observed by the micro lenses.

Due to the concatenated projections of the main lens and the MLA, defining a multiple view geometry (MVG) for a plenoptic camera is not a straightforward matter. To obtain depth estimates from corresponding points in two micro images which belong to separate recordings from different positions one first must establish this correspondence. For traditional cameras, corresponding points can be found in a linear 1D space (standard epipolar geometry), since the camera can be simplified to be a pinhole camera. Therefore, the aim was to find an equivalent mathematical representation of the plenoptic camera which no longer includes the deflection of light rays by the main lens.

Similar to the way that an object point is projected from the object to the image space through the main lens, a micro lens center, which is a 3D point in image space, can be projected through the main lens to the object space.

Thus, the resulting object distance z'_{C0} of a micro lens center can be calculated using the thin lens equation and is defined as given in eq. (4.1).

$$z'_{C0} = \frac{f_L \cdot b_{L0}}{b_{L0} - f_L} \tag{4.1}$$

Here f_L is the main lens' focal length and b_{L0} is the distance between MLA and main lens plane, as shown in Figure 4.1. Figure 4.2 shows the projected micro lens centers, resulting in a virtual camera array. From the specific example shown in Figure 4.2 one can see that for the given setup ($f_L > b_{L0}$), the micro lenses are projected behind the main lens. For later use we define z_{C0} as the negative value of z'_{C0}:

$$z_{C0} := -z'_{C0} = \frac{f_L \cdot b_{L0}}{f_L - b_{L0}}. \tag{4.2}$$

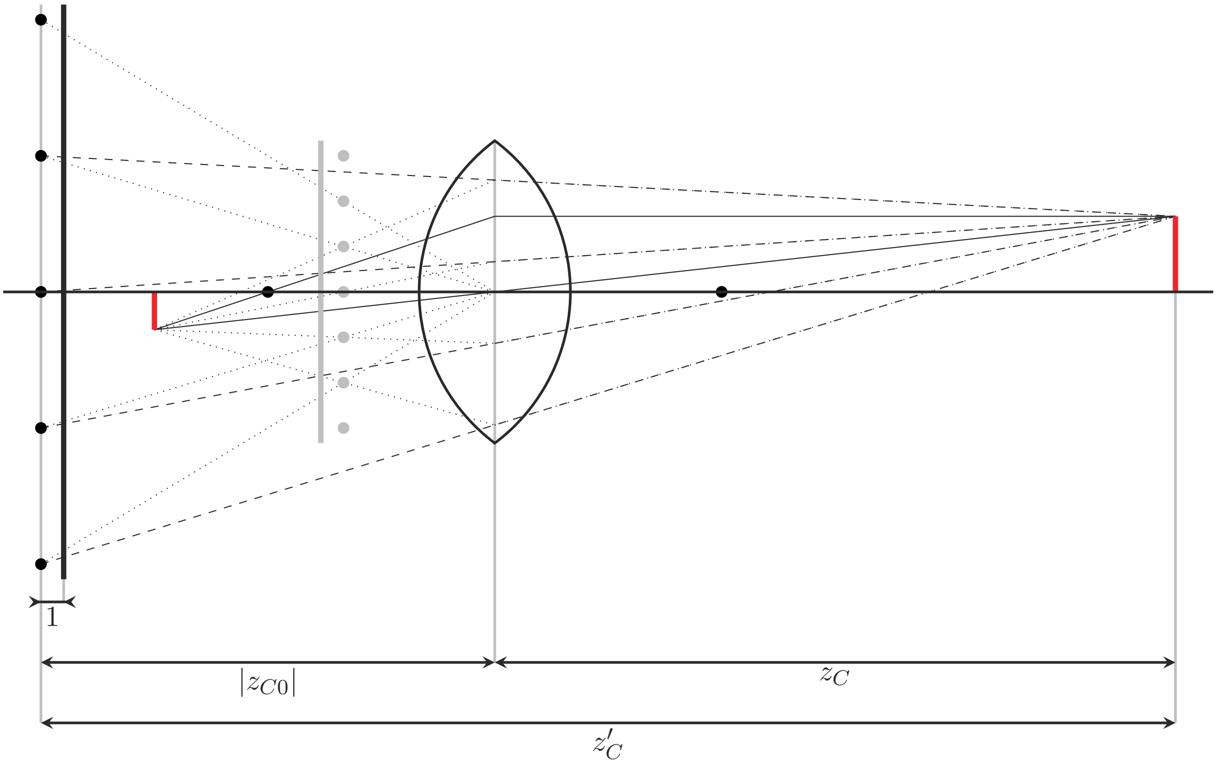


Figure 4.2: New interpretation of a focused plenoptic camera. MLA is projected from image space to object space and results in a virtual array of very narrow FOV pinhole cameras at distance $|z_{C0}|$ to the main lens. The figure shows the case $f_L > b_{L0}$, which results in a virtual array behind the main lens, whereas for the case of $f_L < b_{L0}$, the array would be projected in front of the main lens. The distance between projected micro lens centers and virtual image sensor has been normalized to 1.

With this definition, z'_C given in eq. (4.3) represents the object distance with respect to the virtual camera array.

$$z'_C := z_C + z_{C0} \tag{4.3}$$

The coordinates in x- and y-direction of a projected micro lens center are received as the intersection of the projected MLA plane with the main lens' central ray through the corresponding real micro lens. Thus, one receives a projected micro lens center in camera coordinates $\boldsymbol{p}_{ML}$ from the coordinates of the real micro lens center $\boldsymbol{c}_{ML}$ as given in eq. (4.4).

$$\begin{aligned}\boldsymbol{p}_{ML} = \begin{bmatrix} p_{MLx} \\ p_{MLy} \\ -z_{C0} \end{bmatrix} &= -\boldsymbol{c}_{ML}\frac{z_{C0}}{b_{L0}} = -\begin{bmatrix} c_{MLx} \\ c_{MLy} \\ b_{L0} \end{bmatrix}\frac{z_{C0}}{b_{L0}} \\ &= -\boldsymbol{c}_{ML}\frac{f_L}{f_L - b_{L0}} = \boldsymbol{c}_{ML}\frac{f_L}{b_{L0} - f_L}\end{aligned} \tag{4.4}$$

Here, the minus in front of $\boldsymbol{c}_{ML}$ results due to the transformation from image coordinates to object coordinates (see Section 1.6 for the definition of different coordinate systems).

The projected micro images are defined in such a way that they have a normalized focal length and are parallel to the x-y-plane at $z_C = 1 - z_{C0}$. This way, the central projection from homogeneous 2D to 3D coordinates is defined by simply a scaling in all three dimensions with the effective object distance z'_C. A point $\boldsymbol{x}_p$ in the projected micro image is calculated based on the respective point $\boldsymbol{x}_{ML}$ in the real micro image as follows:

$$\begin{aligned}\boldsymbol{x}_p = \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} &= \boldsymbol{x}_{ML} \frac{f_L - b_{L0}}{f_L \cdot B} + \frac{\boldsymbol{c}_{ML}}{f_L} \\ &= \begin{bmatrix} x_{ML} \\ y_{ML} \\ B \end{bmatrix} \frac{f_L - b_{L0}}{f_L \cdot B} + \frac{\boldsymbol{c}_{ML}}{f_L}. \end{aligned} \tag{4.5}$$

In addition to the regular camera coordinates $\boldsymbol{x}_C$, with their origin being at the center of the main lens, separate camera coordinates $\boldsymbol{x}'_C$ are defined for each micro lens, i.e. for each virtual camera. This is done because each projected micro lens results in different center coordinates $\boldsymbol{p}_{ML}$, which in turn define the origin of the respective camera coordinate systems. The coordinates $\boldsymbol{x}'_C$ will subsequently be called *effective camera coordinates*. Therefore, one obtains the following relation between regular (main lens centered) camera coordinates $\boldsymbol{x}_C$ and effective (projected micro lens centered) camera coordinates $\boldsymbol{x}'_C$:

$$\boldsymbol{x}_C := \boldsymbol{x}'_C + \boldsymbol{p}_{ML} = z'_C \boldsymbol{x}_p + \boldsymbol{p}_{ML} \tag{4.6}$$

Even though Figure 4.1 and 4.2 only show a setup for $f_L > b_{L0}$, the presented projection is valid for almost any setup. The only setup for which no projection is possible is for $f_L = b_{L0}$. Here, the micro lenses are projected to infinity. However, for this case the micro images no longer represent central perspective images of the real world. Instead, each micro image yields an orthographic projection of the object space and because of this, depth estimation becomes independent from the absolute object distance. Here, each of the orthographic images has a different viewing angle. For $f_L < b_{L0}$ it could even be the case that the MLA is projected behind the recorded scene and effectively observes it from the rear. Hence, the depth accuracy will increase with increasing object distance.

A plenoptic camera model has been found which no longer includes the deflection of light rays by the main lens. As this new model represents the mathematical equivalent of a standard camera array, multiple view epipolar geometry can be defined directly for the micro images recorded by a plenoptic camera. Furthermore, projection from object to image space and vice versa can be performed in a single step, rather than performing an intermediate step of calculating the virtual depth and virtual image coordinates.

4.2 Multiple View Epipolar Geometry

This section defines a multiple view geometry (MVG) for plenoptic cameras based on the model introduced in Section 4.1. While in the following the plenoptic camera based epipolar geometry is derived for the specific case of two views, generalizing it to apply to more views is quite a straightforward matter.

Considering a set of images (views) taken with the plenoptic camera from different locations. Let i and j be two particular views of this set. Equation (4.7) defines the transformation of a 3D object point with camera coordinates $\boldsymbol{x}_C^{(i)}$ in the i-th view to camera coordinates $\boldsymbol{x}_C^{(j)}$ in the j-th view based on the rigid body transformation $\boldsymbol{G}_{ij} \in \mathrm{SE}(3)$:

$$\boldsymbol{x}_C^{(j)} = \boldsymbol{G}_{ij} \cdot \boldsymbol{x}_C^{(i)} = \begin{bmatrix} \boldsymbol{R}_{ij} & \boldsymbol{t}_{ij} \\ \boldsymbol{0} & 1 \end{bmatrix} \cdot \boldsymbol{x}_C^{(i)}. \tag{4.7}$$

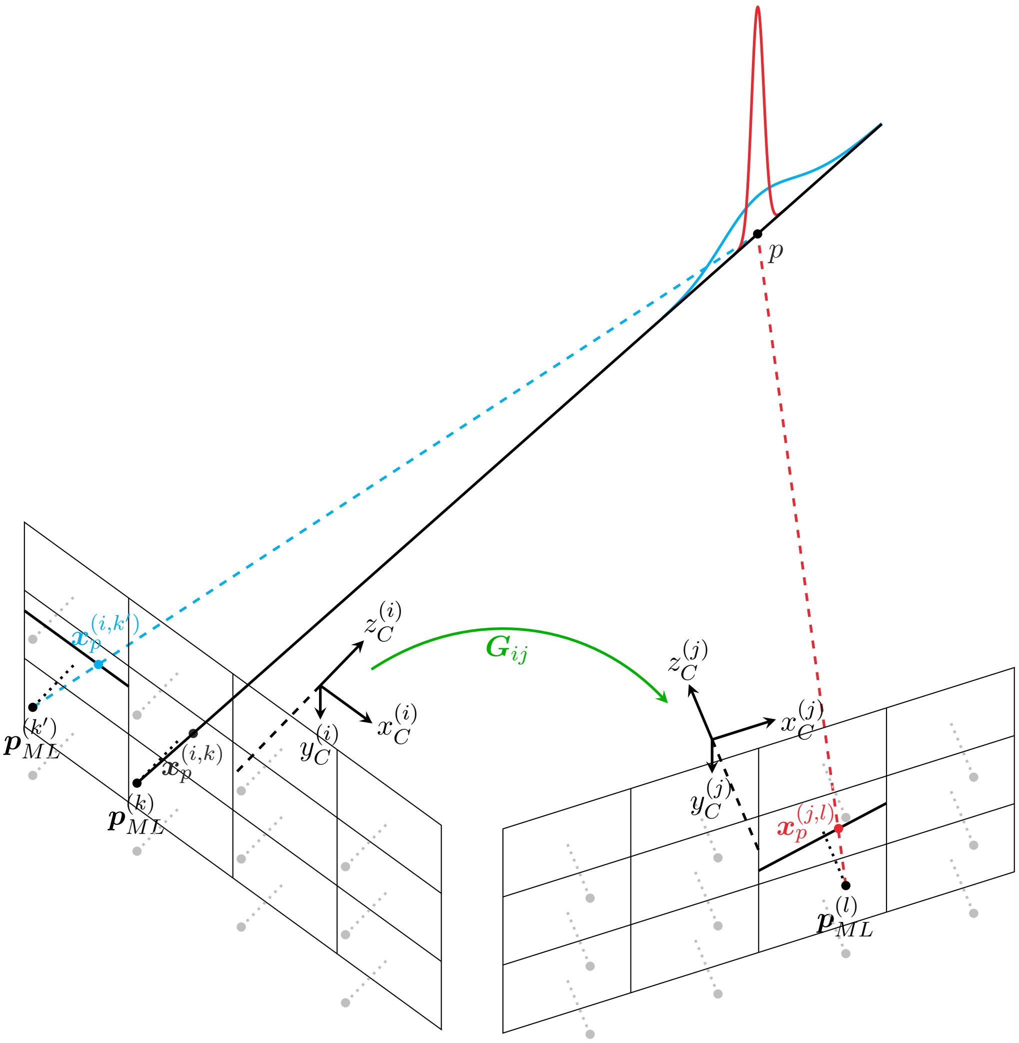

p
$\boldsymbol{x}_p^{(i,k')}$
$\boldsymbol{p}_{ML}^{(k')}$
$\boldsymbol{x}_p^{(i,k)}$
$\boldsymbol{p}_{ML}^{(k)}$
$z_C^{(i)}$
$x_C^{(i)}$
$y_C^{(i)}$
$\boldsymbol{G}_{ij}$
$z_C^{(j)}$
$x_C^{(j)}$
$y_C^{(j)}$
$\boldsymbol{x}_p^{(j,l)}$
$\boldsymbol{p}_{ML}^{(l)}$

Figure 4.3: Multiple view epipolar geometry for plenoptic cameras in the case of two distinct views. The short baseline between micro images of the same view results in a large depth variance. However, increasing the stereo baseline by finding stereo correspondences in the micro images of a second view significantly reduces the depth variance. Due to the defined epipolar geometry, seeking corresponding points from multiple views results in a linear search along the epipolar line.

Similarly to a point in regular camera coordinates, a point in effective camera coordinates $\boldsymbol{x}_C'^{(i,k)}$ of the k-th micro lens in the i-th view can be transformed into the effective camera coordinates $\boldsymbol{x}_C'^{(j,l)}$ of the l-th micro lens in the j-th view:

$$\boldsymbol{x}_C'^{(j,l)} = \boldsymbol{G}_{ij}'^{(kl)} \cdot \boldsymbol{x}_C'^{(i,k)}. \tag{4.8}$$

Here, $\boldsymbol{G}_{ij}'^{(kl)} \in \mathrm{SE}(3)$ is again a rigid body transformation defined as follows:

$$\boldsymbol{G}_{ij}'^{(kl)} = \begin{bmatrix} \boldsymbol{R}_{ij} & \boldsymbol{t}_{ij}'^{(kl)} \\ \boldsymbol{0} & 1 \end{bmatrix} \tag{4.9}$$

$$\boldsymbol{t}_{ij}'^{(kl)} = \boldsymbol{t}_{ij} - \boldsymbol{p}_{ML}^{(l)} + \boldsymbol{R}_{ij} \cdot \boldsymbol{p}_{ML}^{(k)}. \tag{4.10}$$

In eq. (4.10) $\boldsymbol{p}_{ML}^{(k)}$ and $\boldsymbol{p}_{ML}^{(l)}$ are the coordinates of the respective projected micro lens centers, as defined in eq. (4.4). With respect to non-homogeneous effective camera coordinates eq. (4.8) can be restated as follows:

$$\boldsymbol{x}_C^{\prime(j,l)} = \boldsymbol{R}_{ij} \cdot \boldsymbol{x}_C^{\prime(i,k)} + \boldsymbol{t}_{ij}^{\prime(kl)}. \tag{4.11}$$

Based on this definition, one can derive the epipolar geometry between one micro image in the i-th view and another micro image in the j-th view. From projected image coordinates $\boldsymbol{x}_p^{(i,k)}$ (i-th view) and $\boldsymbol{x}_p^{(j,l)}$ (j-th view) one obtains the effective camera coordinates $\boldsymbol{x}_C^{\prime(i,k)}$ and $\boldsymbol{x}_C^{\prime(j,l)}$ as follows:

$$\boldsymbol{x}_C^{\prime(i,k)} = \lambda_i \boldsymbol{x}_p^{(i,k)} \quad \text{with} \quad \lambda_i := z_C^{\prime(i,k)}, \tag{4.12}$$

$$\boldsymbol{x}_C^{\prime(j,l)} = \lambda_j \boldsymbol{x}_p^{(j,l)} \quad \text{with} \quad \lambda_j := z_C^{\prime(j,l)}. \tag{4.13}$$

In eqs. (4.12) and (4.13) λ_i and λ_j are the effective object distances of the respective views. The parameters λ_i and λ_j are the scaling factors mentioned in the previous section, which define the projection from 2D image to 3D object coordinates.

In order to enhance readability, in the following all indices are omitted which do not lead to ambiguous definitions ($\boldsymbol{x}_C^{\prime(i)} := \boldsymbol{x}_C^{\prime(i,k)}$; $\boldsymbol{x}_C^{\prime(j)} := \boldsymbol{x}_C^{\prime(j,l)}$; $\boldsymbol{R} := \boldsymbol{R}_{ij}$; $\boldsymbol{t}' := \boldsymbol{t}_{ij}^{\prime(kl)}$). Inserting eq. (4.12) and (4.13) into eq. (4.11) leads to the following relation:

$$\boldsymbol{x}_C^{\prime(j)} = \lambda_j \boldsymbol{x}_p^{(j)} = \lambda_i \boldsymbol{R} \cdot \boldsymbol{x}_p^{(i)} + \boldsymbol{t}', \tag{4.14}$$

where λ_j can be written as follows:

$$\lambda_j = \lambda_i \left[\boldsymbol{R}\right]_3 \cdot \boldsymbol{x}_p^{(i)} + \left[\boldsymbol{t}'\right]_3. \tag{4.15}$$

In eq. (4.15) $[\boldsymbol{R}]_3$ is the third row of the matrix $\boldsymbol{R}$ and $[\boldsymbol{t}']_3$ is the third element of the vector $\boldsymbol{t}'$ (see Section 1.6).

After combining eq. (4.14) and eq. (4.15) one receives a linear function in λ_i, as given in eq. (4.16).

$$\lambda_i \left([\boldsymbol{R}]_3 \cdot \boldsymbol{x}_p^{(i)} \right) \boldsymbol{x}_p^{(j)} + \left[\boldsymbol{t}'\right]_3 \cdot \boldsymbol{x}_p^{(j)} = \lambda_i \boldsymbol{R} \cdot \boldsymbol{x}_p^{(i)} + \boldsymbol{t}' \tag{4.16}$$

Therefore, a point on the epipolar line in the micro image of the j-th frame is defined as follows:

$$\begin{aligned} \boldsymbol{x}_p^{(j)} &= \frac{\lambda_i \boldsymbol{R} \cdot \boldsymbol{x}_p^{(i)} + \boldsymbol{t}'}{\lambda_i \left[\boldsymbol{R}\right]_3 \cdot \boldsymbol{x}_p^{(i)} + [\boldsymbol{t}']_3} \\ &= \frac{\lambda_i \boldsymbol{R} \cdot \boldsymbol{x}_p^{(i)} + \boldsymbol{t} - \boldsymbol{p}_{ML}^{(l)} + \boldsymbol{R} \cdot \boldsymbol{p}_{ML}^{(k)}}{\lambda_i \left[\boldsymbol{R}\right]_3 \cdot \boldsymbol{x}_p^{(i)} + [\boldsymbol{t}]_3 + z_{C0} + [\boldsymbol{R}]_3 \cdot \boldsymbol{p}_{ML}^{(k)}}. \end{aligned} \tag{4.17}$$

Using this epipolar geometry, one is able to deal with stereo observations from different micro images of different views.

Equation (4.5) shows a linear relationship between micro image coordinates $\boldsymbol{x}_{ML}$ and projected micro image coordinates $\boldsymbol{x}_p$. Hence, the epipolar geometry defined for projected micro images can be applied directly to the micro images on the sensor.

Figure 4.3 visualizes the epipolar geometry for a specific micro lens with respect to two different reference micro lenses. On the one hand a reference micro lens (with index k') which is located in the same light field frame and on the other hand a reference micro lens (with index l) which is located in a different frame. Since the micro lens located in the same frame only results in a very short stereo baseline, the estimate will result in a low depth accuracy. The baseline to the micro lens located in a different frame, on the other hand, is much larger, and therefore the depth accuracy will also increase.

For the sake of simplicity, the micro lenses in Figure 4.3 are placed on a rectangular grid instead of a hexagonal grid, as is the case for most plenoptic cameras.

4.3 Properties and Limitations of Plenoptic Cameras

Compared to a monocular camera or stereo cameras, the plenoptic camera builds a more complex optical system. Therefore, the plenoptic camera is analyzed from a mathematical perspective in more detail with respect to 3D reconstruction.

In Section 4.3.1, the relationship between the virtual depth v and the effective object distance z'_C is derived. While the virtual depth v can be estimated directly from the images of a focused plenoptic camera, defines z'_C the metric object distance for the new camera interpretation introduced in Section 4.1.

In Section 4.3.2 a comparison between the theoretical depth accuracy of an unfocused plenoptic camera (plenoptic 1.0 rendering) and the one of a focused plenoptic camera (full resolution or plenoptic 2.0 rendering) is carried out.

4.3.1 Effective Object Distance vs. Virtual Depth

For the case that two micro lenses are within the same frame ($\boldsymbol{t} = \boldsymbol{0}$ and $\boldsymbol{R} = (\delta_{mn})_{m,n\in\{1,2,3\}}$), the epipolar geometry defined by eq. (4.17) simplifies to the following:

$$\boldsymbol{x}_p^{(j)} = \lambda_i^{-1}\left(\boldsymbol{p}_{ML}^{(k)} - \boldsymbol{p}_{ML}^{(l)}\right) + \boldsymbol{x}_p^{(i)}. \tag{4.18}$$

From eq. (4.18) one receives the disparity in the projected micro images μ_p as given in eq. (4.19).

$$\mu_p := \frac{\langle \boldsymbol{x}_p^{(j)} - \boldsymbol{x}_p^{(i)}, \boldsymbol{p}_{ML}^{(k)} - \boldsymbol{p}_{ML}^{(l)}\rangle}{\|\boldsymbol{p}_{ML}^{(k)} - \boldsymbol{p}_{ML}^{(l)}\|} = \lambda_i^{-1}\|\boldsymbol{p}_{ML}^{(k)} - \boldsymbol{p}_{ML}^{(l)}\| = z'^{-1}_{Ci}\|\boldsymbol{p}_{ML}^{(k)} - \boldsymbol{p}_{ML}^{(l)}\| \tag{4.19}$$

From eq. (4.19) one can see that for in-frame depth estimation (micro images used for stereo matching lie within the same light field frame) the disparity μ_p is proportional to the inverse effective object distance $\lambda^{-1} = z'^{-1}_C$. From eq. (2.12) in Section 2.4 one can see that the disparity μ in the real micro images is proportional to the inverse virtual depth v^{-1}.

From eq. (4.5) it follows that the coordinates of a point in a real micro image $\boldsymbol{x}_{ML}$ and the coordinates of the corresponding point in the projected micro images $\boldsymbol{x}_p$ have a linear relationship. Hence, the respective disparities μ and μ_p are also linearly connected. This leads to the conclusion that there has to be a linear relationship between z'^{-1}_C and v^{-1}, both of which are proportional to the respective disparities μ_p and μ (see eqs. (4.19) and (2.12)). This is proven by replacing the metric object distance with its definition based on the thin lens equation:

$$\begin{aligned} z'_C = z_C + z_{C0} &= \left(\frac{1}{f_L} - \frac{1}{b_L}\right)^{-1} + z_{C0} \\ &= \left(\frac{1}{f_L} - \frac{1}{v\cdot B + b_{L0}}\right)^{-1} + \frac{f_L\cdot b_{L0}}{f_L - b_{L0}}. \end{aligned} \tag{4.20}$$

Rearranging eq. (4.20) gives the following definition of z'^{-1}_C with respect to v^{-1}:

$$z'^{-1}_C = -\frac{(f_L - b_{L0})^2}{B\cdot f_L^2}\cdot v^{-1} + \frac{f_L - b_{L0}}{f_L^2}. \tag{4.21}$$

Since both z'^{-1}_C and v^{-1} linearly depend on the disparity μ, estimated from the micro images, both will have similar probability distributions. One can assume that v^{-1} is Gaussian distributed as will be shown in Section 5.1.

4.3.2 Effective Stereo Baseline – Unfocused vs. Focused

Previously it was shown that a focused plenoptic camera can be interpreted as an array of very narrow field of view (FOV) virtual cameras with high resolution (full resolution or plenoptic 2.0 rendering). An unfocused plenoptic camera can be interpreted as an array of wide FOV virtual cameras with low resolution (Hahne et al. [2014]) (plenoptic 1.0 rendering). Here, the resolution of a sub-aperture image, observing the complete scene, is limited by the number of micro lenses in the MLA.

Even though the images of both concepts have different characteristics, the camera setups differ only by the focal length of the micro lenses. For some cases the plenoptic 1.0 and 2.0 concept can even be applied to the same raw image (Lumsdaine and Georgiev [2009]).

In the following, the effective stereo baselines for both concepts are compared to each other and therefore the benefits that each concept yields with regards to 3D reconstruction are depicted. The calculations for plenoptic 1.0 rendering are based on the model of Hahne et al. [2014], while the calculations for plenoptic 2.0 rendering rely on the camera interpretation described in Section 4.1.

For plenoptic 1.0 rendering the maximum stereo baseline $\varDelta B_{1.0\,\mathrm{max}}$ results from the distance B between MLA and sensor, the micro lens diameter D_M, and the main lens focal length f_L:

$$\varDelta B_{1.0\,\mathrm{max}} = \frac{D_M \cdot f_L}{B}. \tag{4.22}$$

Since the virtual camera array is formed at distance f_L in front of the main lens, the object distance z_C can be calculated based on the pixel disparity μ in the sub-aperture images as given in eq. (4.23).

$$z_C = \frac{\varDelta B_{1.0\,\mathrm{max}} \cdot f_L}{D_M \cdot \mu} + f_L \tag{4.23}$$

The disparity μ is scaled by the micro lens diameter D_M, which defines the size of a pixel in the sub-aperture image. From eq. (4.23) one receives the depth accuracy $\sigma_{z_C 1.0}$ for plenoptic 1.0 rendering as a function of the disparities standard deviation σ_μ, given in eq. (4.24).

$$\sigma_{z_C 1.0} = \left| \frac{\partial z_C}{\partial \mu} \right| \cdot \sigma_\mu = \frac{(z_C - f_L)^2}{f_L^2} \cdot B \cdot \sigma_\mu \tag{4.24}$$

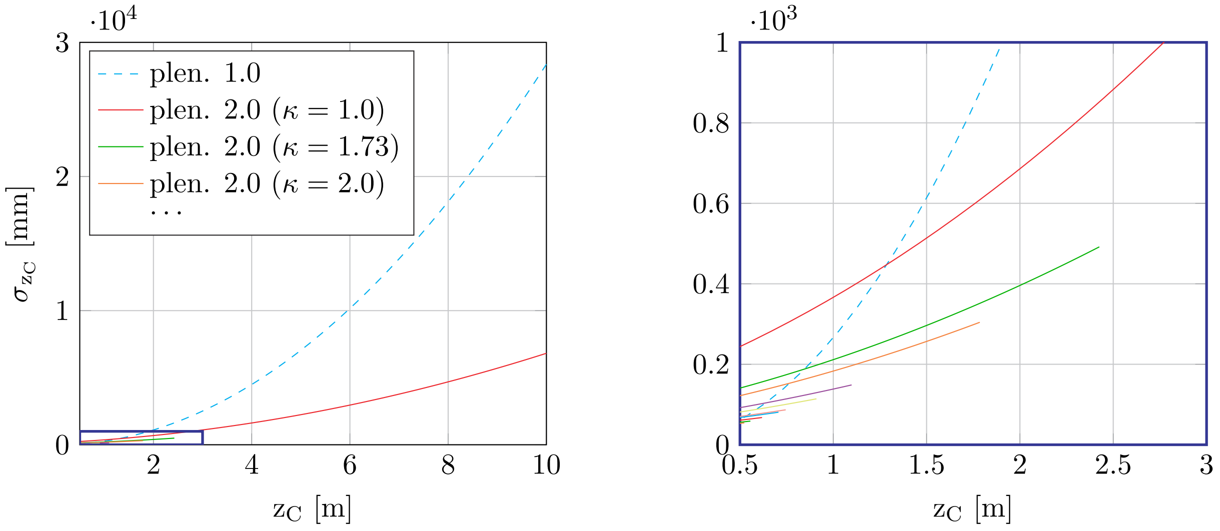
Using the plenoptic camera interpretation of Section 4.1 one can calculate the effective stereo baseline and therefore the theoretical depth accuracy for plenoptic 2.0 (full resolution) rendering, in a similar way. The effective stereo baseline $\varDelta B_{2.0}(\kappa)$ for a pair of projected micro images is obtained as follows:

$$\begin{aligned} \varDelta B_{2.0}(\kappa) &= \|\boldsymbol{p}_{ML}^{(k)} - \boldsymbol{p}_{ML}^{(l)}\| \\ &= \|\boldsymbol{c}_{ML}^{(k)} - \boldsymbol{c}_{ML}^{(l)}\| \frac{f_L}{b_{L0} - f_L} \\ &= \kappa \cdot D_M \cdot \frac{f_L}{b_{L0} - f_L}. \end{aligned} \tag{4.25}$$

The parameter κ defines the multiple of D_M between the two micro lens centers. Therefore $\kappa \geq 1$ holds true.

The object distance z_C is calculated based on the disparity μ_p in the projected micro images as given in eq. (4.26).

$$z_C = \frac{\varDelta B_{2.0}(\kappa)}{\mu_p} - z_{C0} \tag{4.26}$$



(a) depth accuracy over object distance

(b) zoomed subsection of (a)

Figure 4.4: Depth accuracy of plenoptic cameras based on plenoptic 1.0 and 2.0 rendering. Plenoptic 2.0 (full resolution) rendering results in a much larger stereo baseline and therefore better depth accuracy when compared with plenoptic 1.0 approaches.

Again, the depth accuracy $\sigma_{z_C2.0}$ is received from the standard deviation σ_μ of the pixel disparity as follows:

$$\begin{aligned} \sigma_{z_C2.0} &= \left|\frac{\partial z_C}{\partial \mu_p}\right| \cdot \sigma_{\mu_p} = \left|\frac{\partial z_C}{\partial \mu_p}\right| \cdot \left|\frac{\partial \mu_p}{\partial \mu}\right| \cdot \sigma_\mu \\ &= \frac{(z_C + z_{C0})^2 \cdot b_{L0}}{\kappa \cdot D_M \cdot z_{C0}} \cdot \frac{f_L - b_{L0}}{f_L \cdot B} \cdot s_{pixel} \cdot \sigma_\mu \\ &= \frac{(z_C + z_{C0})^2}{\kappa \cdot D_M} \cdot \frac{(f_L - b_{L0})^2}{f_L^2 \cdot B} \cdot s_{pixel} \cdot \sigma_\mu . \end{aligned} \tag{4.27}$$

In eq. (4.27) $\left|\frac{\partial \mu_p}{\partial \mu}\right|$ defines the scaling from the pixel disparity μ to the disparity μ_p in the projected micro images. The parameter s_{pixel} defines the size of a pixel.

Using eq. (4.24) and eq. (4.27), the expected depth accuracies (standard deviations of the object distance z_C) for plenoptic 1.0 and 2.0 rendering are calculated. Figure 4.4 shows the accuracies for both concepts, while using the same camera parameters ($f_L = 35\,\mathrm{mm}$, $B = 0.35\,\mathrm{mm}$, $b_{L0} = 34.3\,\mathrm{mm}$, $D_M = 0.1265\,\mathrm{mm}$, $s_{pixel} = 5.5\,\mu\mathrm{m}$). For both rendering methods a disparity uncertainty $\sigma_\mu = 1$ pixel was chosen.

From eq. (4.24) and eq. (4.27) one can see that both curves are parabolic shaped. However, for the given setup $\sigma_{z_C1.0}$ has a much steeper slope than $\sigma_{z_C2.0}$ (see Fig. 4.4), which is due to a shorter effective stereo baseline.

For the shown setup ($b_{L0} < f_L$), the virtual camera array of the plenoptic 2.0 rendering lies behind the main lens, while the one for plenoptic 1.0 rendering always lies in front of the main lens at a distance of f_L. This leads to different minima in the curves and therefore to an intersection of both curves, as can be seen in Figure 4.4b.

For smaller object distances the plenoptic 2.0 approach can use micro lenses which are spaced farther apart for stereo matching ($\kappa > 1$), which in turn leads to an improved accuracy.

As can be seen from Figure 4.4, the plenoptic 2.0 approach is always superior to the plenoptic 1.0 approach with respect to depth estimation. The κ-s shown in Figure 4.4 are the first 10 received for a hexagonal arrangement of the MLA (1.00, 1.73, 2.00, 2.65, 3.00, 3.46, 3.61, 4.00, 4.36, 4.58).

The functions were not evaluated for object distances closer than 0.5 m, because the images taken in this range can no longer be considered in focus for both of the concepts. Hence, the real accuracy will deviate from the calculated curves. Moreover, in visual odometry (VO) one is generally interested in larger object distances. To resolve smaller object distances the focus of the camera must be adjusted by adjusting the parameter b_{L0}.

The only way to improve the depth accuracy of plenoptic 1.0 with respect to plenoptic 2.0 rendering is to reduce B. However, this seems unfeasible due to the thickness of the MLA and the resulting impractically small F-number of the main lens (see F-number matching in Perwaß and Wietzke [2012]).

5 Probabilistic Light Field based Depth Estimation

At first glance the fact that this chapter covering depth estimation is placed ahead of the topic of camera calibration (Chapter 6) might seem confusing. This can be done because the micro images of a plenoptic cameras are rectified by construction. Therefore, one of the plenoptic camera's convenient properties is that depth estimation can be performed without applying a full camera calibration. The only entities which have to be known for estimating depth are the coordinates of the micro lens centers $\boldsymbol{c}_{ML}$. These can be estimated based on a single white image captured with the plenoptic camera (e.g. Cho et al. [2013]). Since at the current point in time no further knowledge about the intrinsic camera parameters in available, one must consider the estimated centers of the micro images to be equivalent to the micro lens centers $\boldsymbol{c}_{ML}$. In practice this is not the case. The way it influences the projection model of the plenoptic camera, and thereby influences the estimated depth, will be discussed in later chapters (Sections 6.2.3 and 9.2).

Most light field based depth estimation approaches rely on higher order 4D light field abstractions like sub-aperture images or epipolar plane images (EPIs). While these approaches generally are computationally quite complex, they also generate highly accurate and globally consistent disparity maps from the light field data. The goal here, however, is to develop an efficient algorithm which works directly with the raw data (i.e. micro images) recorded by a focused plenoptic camera and therefore avoids any preprocessing steps.

Virtual depth estimation in a focused plenoptic camera can be considered a multiple view stereo problem. Each virtual image point is mapped to multiple micro images. However, the problem simplifies since all micro lenses have the same orientation by construction and thus, the micro images are already rectified.

One approach to solving such a multiple view stereo problem would be to optimize either a photometric or geometric error function over multiple micro images. One would have to establish point correspondences over multiple micro images with sub-pixel accuracy or run a multi-dimensional photometric optimization over the entire virtual depth range. Furthermore, due to the extremely small size of the micro images, traditional corner detection will very likely suffer from an insufficient number of points.

Here, a different approach is pursued, namely one which is based on multiple depth observations received from different micro image pairs. To combine these multiple observations in a single estimate a probabilistic virtual depth model is defined (Section 5.1). Depth observations from the different micro images are obtained in an incremental way using a graph of stereo baselines (Section 5.2). Here one benefits from the structure of the recorded micro images, where for each virtual image point stereo pairs with short and long baselines are present. Correspondences are determined by performing a 1D intensity error minimization along the epipolar line (Section 5.3). For each depth observation an uncertainty is calculated. The uncertainties are used to incorporate the obtained depth observations into the probabilistic model (Section 5.4). This approach is similar to the one of Engel et al. [2013], where the probabilistic model is used to merge depth observations in monocular visual odometry (VO). The depth map estimated in micro images is projected to the virtual image space (Section 5.5) and is refined in a filtering step (Section 5.6.1). Finally, it is described how a totally focused intensity image for the virtual main lens image can be synthesized (Section 5.7).

Throughout this entire chapter coordinates can be considered to have pixel dimensions.

5.1 Probabilistic Virtual Depth

From eq. (2.12) one can see that the inverse virtual depth v^{-1} is proportional to the disparity μ observed from a pair of micro images. Hence, the inverse virtual depth $z = v^{-1}$, given in eq. (5.1), is defined.

$$z = \frac{1}{v} = \frac{\mu}{\Delta c_{ML}} \tag{5.1}$$

Since pixel correspondences are determined by matching pixel intensities, the sensor noise is considered to be the main error source which disturbs the estimated disparity μ and thereby also disturbs the inverse virtual depth z. Furthermore, the effect of differently focused micro images which occur in multi-focus plenoptic cameras (Perwaß and Wietzke [2012]) is modeled as a second error source disturbing the disparity estimation. Any misalignments of the MLA with respect to the image sensor or offsets on the micro lens centers are completely neglected. One can do this because, as one can see from eq. (5.1), the estimate of z relies only on the baseline distance Δc_{ML} and the disparity μ, both of which result as differences of absolute 2D positions in pixel coordinates. Thus, at least within a local neighborhood, the estimate of z is invariant of alignment errors on the MLA.

The sensor noise is usually modeled as additive white Gaussian noise (AWGN). Since pixel correspondences are estimated based on intensity values, the disparity μ, and thus the estimated inverse virtual depth z, can also be considered Gaussian distributed.

In the following, the inverse virtual depth hypothesis of a pixel is given by the random variable $Z \sim \mathcal{N}(z, \sigma_z^2)$, defined by the probability density function $f_Z(x)$ as given in eq. (5.2).

$$f_Z(x) = \frac{1}{\sqrt{2\pi}\sigma_z} e^{-\frac{(x-z)^2}{2\sigma_z^2}} \tag{5.2}$$

5.2 Graph of Stereo Baselines

For stereo matching a graph of stereo baselines is defined. This graph defines which micro images are matched to each other. Each baseline in the graph is defined by its length Δc_{ML} as well as its 2D orientation on the MLA plane $\boldsymbol{e}_p = [e_{px}, e_{py}]^T$. Since the micro images are rectified by construction, the orientation vector of the baseline is equivalent to that of the corresponding epipolar line. Thus, $\boldsymbol{e}_p$ defines the epipolar line for each pixel of the micro lens pair. In the following, it will be always considered that $\boldsymbol{e}_p$ is normed to unity ($||\boldsymbol{e}_p|| = 1\,\text{pixel}$).

In the graph the baselines are sorted in ascending order with respect to their length. This is also the order in which stereo matching will be performed. For initialization of the depth hypotheses the algorithm benefits from short baselines, for which it is more likely to find unique matches, while long baselines improve the accuracy of the estimation but are also more likely to result in ambiguous matches. By performing such an incremental approach, the results for short baselines can be used as prior information for micro image pairs which are connected by a longer

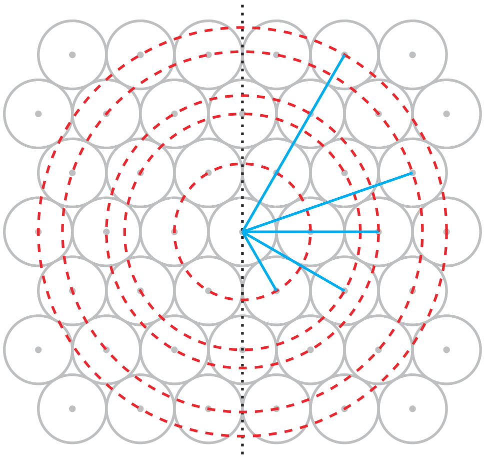
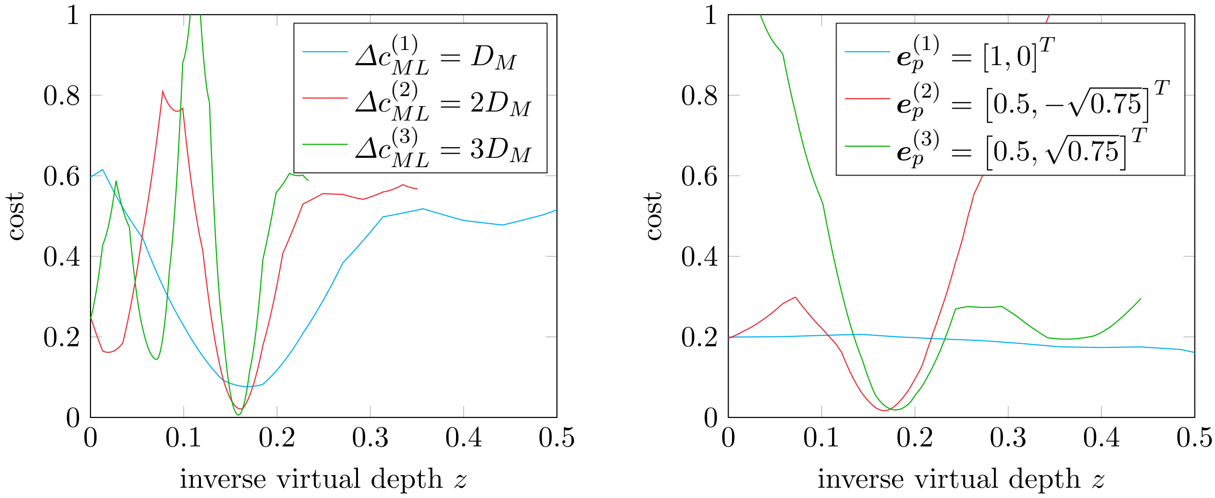


Figure 5.1: Five shortest stereo baseline distances in a hexagonal micro lens grid. For a micro lens, stereo matching is only performed with neighbors for which the baseline angle ϕ is in the range $-90° \leq \phi < 90°$.



(a) Cost function for different baseline distances

(b) Cost function for different epipolar line angles

Figure 5.2: Matching cost plotted over the entire inverse virtual depth range from $z = 0$ ($v \to \infty$) to $z = 0.5$ ($v = 2$). (a) Matching cost for different baseline distances in an image region of strong intensity variation. While the short baseline $\Delta c_{ML}^{(1)}$ results in a unique but broad minimum, the minimum becomes more distinct with longer baselines. However, the cost function also shows ambiguous optimal solutions. (b) Matching cost along an edge with intensity gradient $\boldsymbol{g}_I \approx [0,1]^T$ ($\boldsymbol{g}_I^T \boldsymbol{e}_p^{(1)} \approx 0$). Thus, no accurate depth can be estimated along $\boldsymbol{e}_p^{(1)}$, while $\boldsymbol{e}_p^{(2)}$ and $\boldsymbol{e}_p^{(3)}$ result in unique minima.

baseline. By way of example, Figure 5.2a shows the matching cost (which will be defined below in eq. (5.8)) as a function of the inverse virtual depth z of a certain image point for three different baseline distances. While the shortest baseline $\Delta c_{ML}^{(1)}$ results in a unique but broad minimum, the minimum becomes more distinct with longer baselines. Yet at the same time the cost function shows ambiguous optimal solutions.

Stereo matching is performed for each pixel on the raw sensor image separately. One must assure that corresponding pixels in different micro images are matched to each other only once.

Thus, it suffices to perform matching only with respect to micro images to the right of the reference micro image. All micro images to the left will establish correspondence when they are considered as reference. Thus, only baselines or epipolar lines with an angle $-90° \leq \phi < 90°$ are considered.

Figure 5.1 shows the five shortest baseline distances in a hexagonal MLA grid. Here the red dashed circles represent the respective baseline distance around the micro lens of interest. The solid blue lines show one example baseline for each distance, while only baselines to the right of the dotted line are used for stereo matching. The epipolar line $\boldsymbol{e}_p$ is defined in such a way that it points away from the centered reference micro lens.

5.3 Virtual Depth Estimation

The estimation of the inverse virtual depth z is performed for each pixel $\boldsymbol{x}_R = [x_R, y_R]^T$ in the raw (sensor) image $I_{ML}(\boldsymbol{x}_R)$. As already mentioned, the depth observations are obtained starting from the shortest baseline and working up to the largest possible baseline. Based on each new observation, the inverse depth hypothesis $Z_{ML}(\boldsymbol{x}_R)$ of a raw image pixel $\boldsymbol{x}_R$ is updated, thus becoming more reliable.

$$Z_{ML}(\boldsymbol{x}_R) \sim \mathcal{N}(z(\boldsymbol{x}_R), \sigma_z^2(\boldsymbol{x}_R)) \tag{5.3}$$

To reduce computational effort, the first step is to check for each baseline whether the pixel under consideration $\boldsymbol{x}_R$ has a sufficiently high intensity gradient along the epipolar line, as defined in eq. (5.4).

$$|\boldsymbol{g}_I(\boldsymbol{x}_R)^T \boldsymbol{e}_p| \geq T_H \tag{5.4}$$

Here $\boldsymbol{g}_I(\boldsymbol{x}_R)$ represents the intensity gradient vector at the coordinates $\boldsymbol{x}_R$ (eq. (5.5)) and T_H represents a predefined threshold.

$$\boldsymbol{g}_I(\boldsymbol{x}_R) = \boldsymbol{g}_I(x_R, y_R) = \begin{bmatrix} \frac{\partial I(x_R,y_R)}{\partial x_R} & \frac{\partial I(x_R,y_R)}{\partial y_R} \end{bmatrix}^T \tag{5.5}$$

Figure 5.2b shows an example of the matching cost (cf. eq. (5.8)) dependent on the inverse virtual depth z of a certain image point for three different epipolar line angles. In this specific example the intensity gradient $\boldsymbol{g}_I(\boldsymbol{x}_R)$ is almost orthogonal to $\boldsymbol{e}_p^{(1)} = [1, 0]^T$. Therefore, no minimum is obtained in the cost function and hence, depth estimation does not have to be performed for this baseline. However, for the same point $\boldsymbol{x}_R$ both $\boldsymbol{e}_p^{(2)} = \left[0.5, -\sqrt{0.75}\right]^T$ and $\boldsymbol{e}_p^{(3)} = \left[0.5, \sqrt{0.75}\right]^T$ result in unique minima.

5.3.1 Stereo Matching

To find the pixel in a certain micro image which corresponds to the pixel of interest $\boldsymbol{x}_R$, it is searched for the minimum intensity difference along the epipolar line in the corresponding micro image.

If no inverse virtual depth observation for the pixel of interest $\boldsymbol{x}_R$ has been obtained yet, an exhaustive search along the epipolar line must be performed. For this case, the search range is limited by the micro lens border on the one end and by the coordinates of $\boldsymbol{x}_R$ with respect to the micro lens center on the other end. A pixel on the micro lens border results in the maximum observable disparity μ and thus in the minimum observable virtual depth v, while a pixel at the same coordinates as the pixel of interest in the corresponding micro image equals a disparity $\mu = 0$ and thus also equals a virtual depth $v = \infty$. Furthermore, the virtual depth range is lower bounded by the total covering plane (TCP). The total covering plane (TCP) defines the plane closest to the MLA for which it can be guaranteed that any virtual image point $\boldsymbol{x}_V$ on the plane appears focused in at least one micro image. For a multi-focus plenoptic camera with a hexagonally arranged MLA, consisting of three different types of micro lenses, the TCP is at a virtual depth $v = 2$ (see Perwaß and Wietzke [2012]).

If an inverse virtual depth hypothesis $Z_{ML}(\boldsymbol{x}_R)$ for a point $\boldsymbol{x}_R$ already exists, the search range can be limited to $z(\boldsymbol{x}_R) \pm n\sigma_z(\boldsymbol{x}_R)$, where n is usually chosen to be $n = 2$. In the following we define the search range along the epipolar line as given in eq. (5.6).

$$\boldsymbol{x}_R^s(\mu) = \boldsymbol{x}_{R0}^s - \mu \cdot \boldsymbol{e}_p. \tag{5.6}$$

Here $\boldsymbol{x}_{R0}^s$ is defined as the coordinate of a point on the epipolar line at the disparity $\mu = 0$, as given in eq. (5.7).

$$\boldsymbol{x}_{R0}^s = \boldsymbol{x}_R + \Delta c_{ML} \cdot \boldsymbol{e}_p \tag{5.7}$$

Within the search range, the sum of squared intensity differences e_{ISS} over a 1D pixel patch $(1 \times N)$ along the epipolar line is calculated, as defined in eq. (5.8).

$$e_{ISS}(\mu) = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} \left[I_{ML}(\boldsymbol{x}_R + k\boldsymbol{e}_p) - I_{ML}(\boldsymbol{x}_R^s(\mu) + k\boldsymbol{e}_p)\right]^2 \tag{5.8}$$

The resulting estimate is the disparity μ which minimizes $e_{ISS}(\mu)$. For the implementation we found $N = 5$ to be a good choice for the patch size. It is important to emphasize that μ is estimated with sub-pixel accuracy by interpolating linearly between the samples of the intensity image $I_{ML}(\boldsymbol{x}_R)$ and therefore the disparity is not restricted to any regular grid. In the following it is referred to the estimated disparity by $\hat{\mu}$, which defines the corresponding pixel coordinate $\boldsymbol{x}_R^s(\hat{\mu})$.

Observation Uncertainty

For each inverse virtual depth observation, an observation uncertainty, represented by the inverse virtual depth variance σ_z^2, is calculated. Hence, the variance σ_z^2 defines a measure of certainty with which an estimate represents the corresponding real value. For the determination of the variance σ_z^2, on the one hand, a photometric component is considered, which models the effect of sensor noise, and on the other hand a focus component is considered, which models the effect of differently focused micro images, as they occur in a multi-focus plenoptic camera (Perwaß and Wietzke [2012]).

Photometric Disparity Error The effect of sensor noise ϵ_n on the estimated disparity μ is modeled in a photometric disparity error, defined by the variance $\sigma_{\mu(I)}^2$. This error source is inspired by Engel et al. [2013] and is modeled in a similar way.

The variance of the sensor noise σ_n^2 can be considered to be the same for each pixel $\boldsymbol{x}_R$. In doing so, the effect of vignetting correction is neglected. To compensate for vignetting of the micro lenses pixel intensities at micro image boundaries and, accordingly, the additive noise components are amplified.

In the following it will be derived how σ_n^2 affects the disparity estimation. To do this, we formulate the stereo matching by the minimization problem given in eq. (5.9), where the estimated disparity $\hat{\mu}$ is the one which minimizes the squared intensity difference $e_I(\mu)^2$. In the interest of temporarily simplifying the expressions, the sum over a number of pixels is omitted, as defined for $e_{ISS}(\mu)$ in eq. (5.8).

$$\begin{aligned}\hat{\mu} &= \arg\min_{\mu}\left[e_I(\mu)^2\right]\\ &= \arg\min_{\mu}\left[\left(I_{ML}(\boldsymbol{x}_R) - I_{ML}(\boldsymbol{x}_R^s(\mu))\right)^2\right] \qquad (5.9)\end{aligned}$$

To find the minimum, the first derivative of $e_I(\mu)^2$ with respect to μ is calculated and is set to zero. This results in eq. (5.11) as long as $g_I(\mu) \neq 0$ holds true (which is guaranteed by eq. (5.4)).

$$\begin{aligned}\frac{\partial e_I(\mu)^2}{\partial\mu} &= \frac{\partial\left[I_{ML}(\boldsymbol{x}_R) - I_{ML}(\boldsymbol{x}_R^s(\mu))\right]^2}{\partial\mu}\\ &= 2\left[I_{ML}(\boldsymbol{x}_R) - I_{ML}(\boldsymbol{x}_R^s(\mu))\right]\cdot\left[-g_I(\mu)\right] \qquad (5.10)\\ 0 &\overset{!}{=} I_{ML}(\boldsymbol{x}_R) - I_{ML}(\boldsymbol{x}_R^s(\mu)) \qquad (5.11)\end{aligned}$$

In eq. (5.10) the intensity gradient along the epipolar line $g_I(\mu)$ is defined as follows:

$$g_I(\mu) = g_I\left(\boldsymbol{x}_R^s(\mu)\right) = \frac{\partial I_{ML}(\boldsymbol{x}_{R0}^s - \mu\boldsymbol{e}_p)}{\partial\mu} = \boldsymbol{g}_I(\boldsymbol{x}_R^s(\mu))^T\boldsymbol{e}_p. \qquad (5.12)$$

After approximating eq. (5.11) by its first-order Taylor-series at the disparity $\mu = \mu_0$, which is close to $\hat{\mu}$, it can be solved for $\hat{\mu}$ as given in eq. (5.13).

$$\hat{\mu} = \frac{I_{ML}(\boldsymbol{x}_R) - I_{ML}(\boldsymbol{x}_R^s(\mu_0))}{g_I\left(\boldsymbol{x}_R^s(\mu_0)\right)} + \mu_0 \qquad (5.13)$$

If one now considers $I_{ML}(\boldsymbol{x}_R)$ in eq. (5.13) to be disturbed by AWGN, the variance $\sigma^2_{\mu(I)}$ of the disparity μ can be derived as given in eq. (5.14).

$$\begin{aligned}\sigma^2_{\mu(I)} &= \frac{\mathrm{Var}\{I_{ML}(\boldsymbol{x}_R)\} + \mathrm{Var}\{I_{ML}(\boldsymbol{x}_R^s(\mu_0))\}}{g_I(\boldsymbol{x}_R^s(\mu_0))^2}\\ &= \frac{2\sigma_n^2}{g_I(\boldsymbol{x}_R^s(\mu_0))^2} \qquad (5.14)\end{aligned}$$

Figure 5.3 illustrates how the gradient g_I affects the estimation of μ. The blue line represents the tangent at the disparity μ_0, at which the intensity values are projected onto the disparities.

Focus Disparity Error In contrast to regular cameras, which generally are focused to infinity, the micro images of the plenoptic camera can not be considered to be in focus for the entire operating range, especially not for a multi-focus plenoptic camera (Perwaß and Wietzke [2012]). Hence, the focusing itself also affects the stereo observation. In the following the variance $\sigma^2_{\mu(v,k,j)}$ of the focus disparity error is derived.

Let k be the index of the micro image for which mapping is performed, while j is the index of the stereo reference. To model the focus disparity error, it is considered that the real edge $I_{\mathrm{real}}(x)$, which is observed in the micro images, is a perfect Heaviside-step-function $h(x)$ with amplitude A and offset B along the epipolar line:

$$I_{\mathrm{real}}(x) = A\cdot h(x) + B. \qquad (5.15)$$

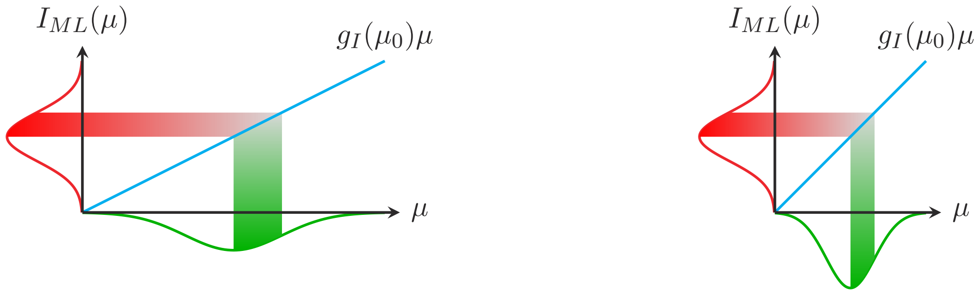


Figure 5.3: Visualization of the photometric disparity error. Sensor noise ϵ_n can be considered additive white Gaussian noise (AWGN) which disturbs the intensity values $I_{ML}(\boldsymbol{x}_R)$ and thus affects the disparity observation as AWGN. As shown on the left, for a low image gradient along the epipolar line, the influence of the sensor noise n_I is stronger than for a high image gradient, as it is shown on the right.

The variable x is the position on the respective epipolar line in relation to the step position μ_{0i} ($i \in \{k, j\}$):

$$x = \mu_k - \mu_{0k} = \mu_j - \mu_{0j}. \tag{5.16}$$

Therefore, the correct disparity is defined by $\mu^* = \mu_{0j} - \mu_{0k}$. During the imaging process the edge $I_{\text{real}}(x)$ is filtered by a Gaussian filter with variance σ_i^2, which in turn depends on the virtual depth v, the micro lens type and the aperture of a pixel[1].

Thus, on the sensor one receives the following intensity functions along the epipolar line:

$$I_{MLi}(x) = B + \frac{A}{2}\left(1 + \text{erf}\left(\frac{x}{\sigma_i\sqrt{2}}\right)\right) \quad i \in \{k, j\}. \tag{5.17}$$

The estimated disparity $\hat{\mu} = \mu^* + \epsilon_f$ is the one for which both intensities have the same value and therefore, the following condition is fulfilled:

$$I_{MLk}(x) \overset{!}{=} I_{MLj}(x - \epsilon_f), \tag{5.18}$$

$$\text{erf}\left(\frac{x}{\sigma_k\sqrt{2}}\right) \overset{!}{=} \text{erf}\left(\frac{x - \epsilon_f}{\sigma_j\sqrt{2}}\right). \tag{5.19}$$

Since the error function ($\text{erf}(\cdot)$) can not be solved analytically, eq. (5.19) is linearized and one obtains, after some rearranging, the following relationship between the focus disparity error ϵ_f and the position on the edge x:

$$\epsilon_f = x \cdot \left(1 - \frac{\sigma_j}{\sigma_k}\right). \tag{5.20}$$

Figure 5.4 visualizes the focus disparity error ϵ_f, by way of example, for two different positions (x_1 and x_2), in relation to the real edge position on the epipolar line. Here, the red and blue curves represent the intensity along the epipolar line in two different micro images with different blur radii. For a certain position x the estimated disparity $\hat{\mu}$ will be the one for which both curves have the same value and therefore will be shifted by ϵ_f with respect to the real disparity μ^*, as

---

[1] Pixel aperture results rather in a sinc response function. However, the Gaussian kernel gives a good approximation as defocus blur will generally be dominate.

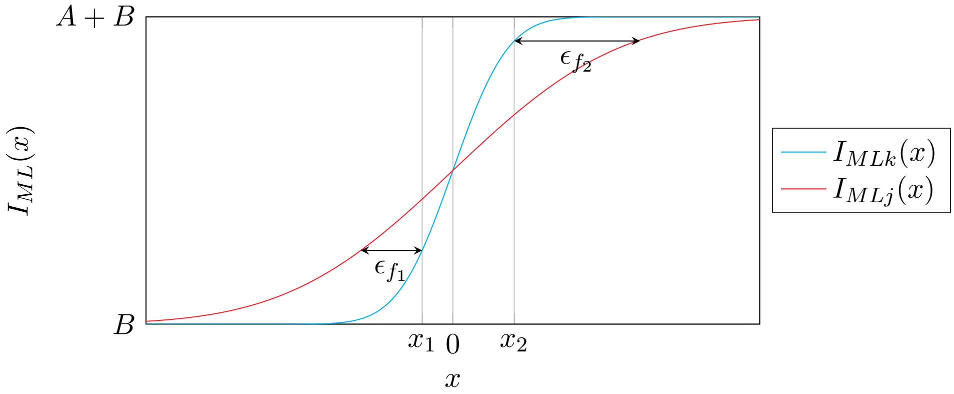


Figure 5.4: Visualization of the focus disparity error ϵ_f for two different positions (x_1 and x_2), in relation to the real edge position, on the epipolar line. The red and blue curves represent the intensity along the epipolar line in two different micro images with different blur radii. For a certain position x on the edge the estimated disparity $\hat{\mu}$ will be the one for which both curves have the same value and therefore will be shifted by ϵ_f with respect to the real disparity μ^*.

given in eq. (5.18). For the case that both images have the same blur radius, both curves are perfectly overlaid and therefore $\epsilon_f = 0$ will hold for any position x.

Considering the position in relation to the real edge x as a random variable with variance σ_x^2, the variance $\sigma^2_{\mu(v,k,j)}$ of the focus disparity error is as follows:

$$\sigma^2_{\mu(v,k,j)} = \sigma_x^2 \cdot \left(1 - \frac{\sigma_j}{\sigma_k}\right)^2 \tag{5.21}$$

The standard deviation σ_i ($i \in \{k, j\}$) depends on the blur diameter s_i of the respective micro image, which is calculated based on the virtual depth v and the micro lens parameters (D_M, f_M, B) using the thin lens equation:

$$s_i = D_M \cdot \left| \frac{1}{v_i} + \frac{B}{f_{Mi}} - 1 \right| \tag{5.22}$$

Since the minimum blur radius is limited by the pixel pitch and the overall optical system, the blur radius has a lower boundary s_0. Thus the following variances σ_k^2 and σ_j^2 result to be:

$$\sigma_k^2 = \beta^2 \cdot \min\{s_k^2, s_0^2\}, \quad \sigma_j^2 = \beta^2 \cdot \min\{s_j^2, s_0^2\} \tag{5.23}$$

The constant parameter β models the scaling from blur diameter to the standard deviation of the Gaussian filter.

Considering the two error sources, photometric disparity error and focus disparity error, as independent random variables, the complete observation uncertainty for the inverse virtual depth z can be defined as given in eq. (5.24)

$$\sigma_z^2 = \mathit{\Delta} c_{ML}^{-2} \cdot \left(\sigma^2_{\mu(I)} + \sigma^2_{\mu(v,k,j)}\right) \tag{5.24}$$

5.4 Updating Virtual Depth Hypothesis

As described in Section 5.2, the observations for the inverse virtual depth z are obtained by starting from the shortest baseline and working up to the longest possible baseline for which a

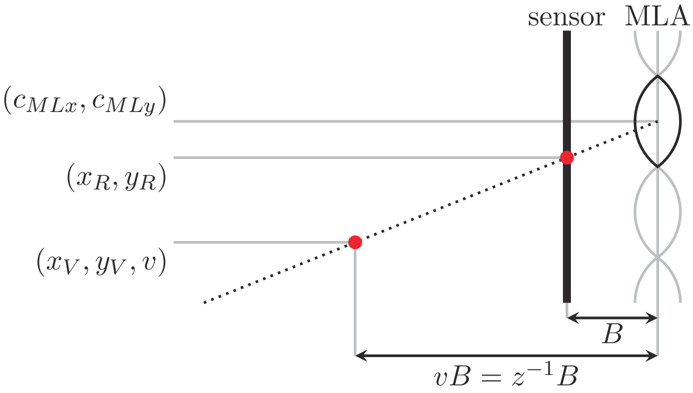


Figure 5.5: Projection of a raw image point $\boldsymbol{x}_R$ to a virtual image point $\boldsymbol{x}_V$ based on an central projection through the micro lens center $\boldsymbol{c}_{ML}$.

virtual image point is still seen in both micro images. An exhaustive stereo matching over all possible micro images[2] is performed for each pixel in this manner. In the algorithm new inverse virtual depth observations are incorporated similar to the update step in a Kalman filter. Thus, the new distribution $\mathcal{N}(z, \sigma_z^2)$ results from the previous distribution $\mathcal{N}(z_p, \sigma_p^2)$ and the new inverse depth observation $\mathcal{N}(z_o, \sigma_o^2)$ as given in eq. (5.25).

$$\mathcal{N}(z, \sigma_z^2) = \mathcal{N}\left(\frac{\sigma_p^2 \cdot z_o + \sigma_o^2 \cdot z_p}{\sigma_p^2 + \sigma_o^2}, \frac{\sigma_p^2 \cdot \sigma_o^2}{\sigma_p^2 + \sigma_o^2}\right) \tag{5.25}$$

The baseline distance $\varDelta c_{ML}$ is more or less proportional to the virtual depth $v = z^{-1}$. From eq. (5.24) one can see that the inverse virtual depth variance σ_z^2 is inversely proportional to $\varDelta c_{ML}^2$. Moreover, the number of observations increases with the virtual depth v, since one point occurs in more micro images. For the case that all depth observations are statistically independent and have the same variance $\sigma_i^2 = \sigma_o^2$ $(i \in 1, 2, \ldots, N)$, the variance of the combined depth estimate σ_z^2 is just N times smaller than the observation variance σ_0^2, as given in eq. (5.26).

$$\frac{1}{\sigma_z^2} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} = \frac{N}{\sigma_o^2} \tag{5.26}$$

In conclusion, one can assume that the inverse virtual depth variance σ_z^2 improves approximately proportionally to v^3.

5.5 Calculating a Virtual Depth Map

Based on the observed inverse virtual depth z, a point in the raw image I_{ML}, defined by the coordinates $\boldsymbol{x}_R$, can be projected in the virtual image space. In the virtual image space, a point is denoted by the coordinates $\boldsymbol{x}_V = \left[x_V, y_V, v = z^{-1}\right]^T$. This projection is defined as follows in eqs. (5.27) and (5.28).

$$x_V = (x_R - c_{MLx})z^{-1} + c_{MLx} \tag{5.27}$$
$$y_V = (y_R - c_{MLy})z^{-1} + c_{MLy} \tag{5.28}$$

Here $\boldsymbol{c}_{ML} = [c_{MLx}, c_{MLy}]^T$ defines the center of the respective micro lens. The projection is visualized in Figure 5.5. Points which are projected to the same virtual image coordinates are

---

[2] Possible micro images are all those which still see the respective virtual image point.

merged based on the updating procedure described in Section 5.4. Hence, building on the depth hypotheses $Z_{ML}(\boldsymbol{x}_R)$ which in turn are based on raw image coordinates, a probabilistic depth map $Z_V(\boldsymbol{x}_V)$ based on virtual image coordinates can be calculated.

5.6 Probabilistic Depth Map Filtering

In the proposed probabilistic depth estimation algorithm, pixel correspondences are found only based on local criteria. Furthermore, each pixel is processed separately without considering a larger neighborhood. Thus, the estimated inverse virtual depth values $z = E\{Z_{ML}(\boldsymbol{x}_R)\}$ in the micro images, which are defined as the expected values of $Z_{ML}(\boldsymbol{x}_R)$, can be considered to be more or less uncorrelated. This is, in fact, not the case for depth maps of real scenes, where neighboring pixels usually are highly correlated, as they very likely belong to the same object. Therefore, a post-processing step is formulated in which the connection between points in a certain neighborhood is modeled and used to refine the depth estimates.

The regularization is done in several steps. Even before the micro image points are projected to the virtual image space as described in Section 5.5, the first step is to perform outlier removal and hole filling for each micro image in the depth map $Z_{ML}(\boldsymbol{x}_R)$ individually (Section 5.6.1). After this, the pixels from the micro images are projected into the virtual image space as described in Section 5.5. After the projection, in the virtual image again outlier removal and hole filling are performed. Finally, the depth map is refined using all inverse virtual depth hypotheses of the pixels within a certain neighborhood (Section 5.6.2).

5.6.1 Removing Outliers and Filling Holes in Micro Images

Due to the very small size of the micro images, the unfiltered depth map received from the recording of a plenoptic camera generally suffers from a large number of outliers. However, the high overlap of the micro images results in multiple depth observations for a certain virtual image point. Therefore, a quite strict outliers removal strategy is carried out, followed by hole filling to enhance the quality of the depth map.

Removing Outliers

Due to ambiguous structures in the micro images, wrong correspondences are occasionally established between pixels in the micro images. In a first step pixels which are outliers with respect to their neighborhood are removed. For each valid depth pixel $\boldsymbol{x}_R^{(i)}$ in the raw image, with the depth hypothesis $\mathcal{N}(z_i, \sigma_{zi}^2)$, an average inverse virtual depth $\overline{z_i}$ and a corresponding variance $\overline{\sigma_{zi}^2}$ of all valid depth pixels $N_{R\text{valid}}$ within a neighborhood $N_R^{(i)}$ are defined:

$$\overline{z_i} = \frac{\sum_{k \in N_x^{(i)}, k \neq i} z_k \cdot \left(\sigma_{zk}^2\right)^{-1}}{\sum_{k \in N_x^{(i)}, k \neq i} \left(\sigma_{zk}^2\right)^{-1}}, \tag{5.29}$$

$$\overline{\sigma_{zi}^2} = \frac{|N_x^{(i)}| - 1}{\sum_{k \in N_x^{(i)}, k \neq i} \left(\sigma_{zk}^2\right)^{-1}}. \tag{5.30}$$

In eq. (5.30) $N_x^{(i)}$ defines the intersection between the set of neighborhood pixels $N_R^{(i)}$ of $\boldsymbol{x}_R^{(i)}$ and the set of valid depth pixels $N_{R\text{valid}}$ ($N_x^{(i)} = N_R^{(i)} \cap N_{R\text{valid}}$). Furthermore, $|N_x^{(i)}|$ is the cardinality of the set $N_x^{(i)}$ and defines the number of elements in the set. Equations (5.29) and (5.30) are actually quite similar to the definition of z and σ_z^2 in eq. (5.25). The only difference is that the sum of the inverse variances is multiplied by the number of valid neighbors ($|N_x^{(i)}| - 1$).

Each pixel $\boldsymbol{x}_R^{(i)}$ that has an inverse virtual depth estimate z_i which does not satisfy the following condition is classified as outlier:

$$(z_i - \overline{z_i})^2 \leq 4 \cdot \overline{\sigma_{zi}^2}. \tag{5.31}$$

For the implementation of the algorithm a squared neighborhood of 5 pixel × 5 pixel is defined.

Filling Holes

After removing the outliers in the micro images, pixels which have an absolute intensity gradient higher than the threshold T_H (same threshold as used for depth estimation, see eq. (5.4)) but no valid depth estimate are filled based on the neighboring pixels. Therefore, again, the average inverse virtual depth $\overline{z_i}$ within a neighborhood region is calculated based on eq. (5.29). The inverse virtual depth $\overline{z_i}$ gives the new depth value for the invalid pixel $\boldsymbol{x}_R^{(i)}$, while the corresponding variance σ_{zi}^2 is initialized to a predefined high value. By setting the initial variance to a high value, these interpolated depth values cannot negatively effect the later regularization by being overweighted.

After removing outliers and filling holes in the micro images, all micro image pixels which have a valid depth hypothesis are projected into the virtual image space, as described in Section 5.5.

5.6.2 Regularization of the Virtual Image

First, outliers in the virtual image space which were not detected in the micro images are removed. Afterwards, the inverse virtual depth values are smoothed, taking into consideration the imaging concept of the plenoptic camera.

For the regularization in the virtual image space, again a neighborhood region $N_V^{(i)}$ is defined for each pixel $\boldsymbol{x}_V^{(i)}$. Each micro lens produces a central perspective projection and thus, objects with a high virtual depth appear smaller on the sensor than objects with a small virtual depth. At the same time virtual image points with a high virtual depth are observed by a higher number of micro lenses. Vice versa back projected virtual image regions with a high virtual depth consist of more points which are spread over a larger region. Hence, for virtual image regularization the size of the neighborhood is defined as a function of the virtual depth v_i of the pixel of interest $\boldsymbol{x}_V^{(i)}$. For each pixel $\boldsymbol{x}_V^{(i)}$ a radius $r(v_i)$ is defined as follows:

$$r(v_i) = \lceil n \cdot v_i \rceil. \tag{5.32}$$

Here n defines a constant parameter. For lower complexity in calculation, the radius $r(v_i)$ defines the maximum allowed Chebyshev distance L_∞ to the pixel $\boldsymbol{x}_V^{(i)}$, rather than the Euclidean distance. For instance, the Chebyshev distance$L_\infty^{(k)}$ between the pixel $\boldsymbol{x}_V^{(k)}$ and the pixel $\boldsymbol{x}_V^{(i)}$ is defined as follows:

$$L_\infty^{(k)} = \max\left(|x_V^{(i)} - x_V^{(k)}|, |y_V^{(i)} - y_V^{(k)}|\right). \tag{5.33}$$

This defines a squared neighborhood $N_V^{(i)}$ around $\boldsymbol{x}_V^{(i)}$.

Removing Outliers

In order to remove outliers in the virtual image, as it is done for the micro images, the mean inverse virtual depth $\overline{z_i}$ and the mean inverse virtual depth variance $\overline{\sigma_{zi}^2}$ of valid depth pixels within the neighborhood region $N_V^{(i)}$ are calculated. Again, these calculations are performed based on eqs. (5.29) and (5.30), while the definition of the set $N_x^{(i)}$ of valid depth pixels in the neighborhood changes as follows:

$$N_x^{(i)} = N_V^{(i)} \cap N_{V\text{valid}}. \tag{5.34}$$

Here, $N_{V\text{valid}}$ is the set of all pixels $\boldsymbol{x}_V$ which have a valid depth estimate. Nevertheless, beside fulfilling the condition defined in eq. (5.31), a pixel $\boldsymbol{x}_V^{(i)}$ has to have a density of valid depth pixels in its neighborhood above a certain threshold T_D.

$$T_D \leq \frac{|N_x^{(i)}|}{|N_V^{(i)}|} \tag{5.35}$$

As can be seen in eqs. (5.27) and (5.28), the calculated virtual image coordinates depend on the estimated depth. Hence, a certain density of points is required for a particular region, since a low number of isolated points very likely results from erroneous depth estimates. This is especially true for regions with a high virtual depth, where points from many micro images are usually projected to the same area. In the implementation the minimum density was set to $T_D = 0.25$.

Filling Holes

At this point there are no intensities available for the virtual image. Furthermore, in order to obtain the intensity for a certain virtual image pixel, its virtual depth value has to be known. Thus, pixel validity is not defined based on the pixel's intensity gradient, as it is done in the raw image. Instead, a valid depth value is assigned to each pixel $\boldsymbol{x}_V$ in the virtual image which has at least one direct neighbor with a valid depth hypothesis. This depth value is calculated as a weighted average of the depths of neighboring pixels, defined in a way similar to eq. (5.29). The corresponding variance again is initialized by a predefined high value.

Refining Depth Estimates

In a final step, the actual correlation between neighboring pixels in the virtual image is established. For this the same neighborhood $N_V^{(i)}$ is employed, as it was used for outliers removal.

Two different states are defined to handle discontinuities in the depth map. A pixel $\boldsymbol{x}_V^{(k)}$ in the neighborhood $N_V^{(i)}$ belongs to either the same object as $\boldsymbol{x}_V^{(i)}$ or it belongs to a different object. These two objects can be considered to be foreground and background. All neighboring pixels $\boldsymbol{x}_V^{(k)}$ ($k \in N_V$) which have estimates similar to the pixel $\boldsymbol{x}_V^{(i)}$ are assigned to the set N_{sim} (same object), while pixels with depth estimates strongly differing from the one of $\boldsymbol{x}_V^{(i)}$ are assigned to the set N_{diff} (different object). This assignment is defined as follows:

$$\begin{array}{ll} k \in N_{\text{sim}} & \text{if } (z_i - z_k)^2 \leq 2(\sigma_{zi}^2 + \sigma_{zk}^2), \\ k \in N_{\text{diff}} & \text{else,} \end{array} \quad \text{with } k \in N_V^{(i)} \cap N_{V\text{valid}}. \tag{5.36}$$

Furthermore, a function $w(d)$ is defined which models the correlation between the depth values in the neighborhood as a function of the distance d between the respective pixels. The function argument d defines the Euclidean distance between the two pixels $\boldsymbol{x}_V^{(i)}$ and $\boldsymbol{x}_V^{(k)}$. In the implementation a Gaussian curve as given in eq. (5.37) was chosen to define the correlation.

$$w(d) = e^{-\frac{d^2}{2\sigma_w^2}} \tag{5.37}$$

The standard deviation σ_w is defined to be proportional to the virtual depth v, as it was already done for the radius of the neighborhood N_V, as defined in eq. (5.32).

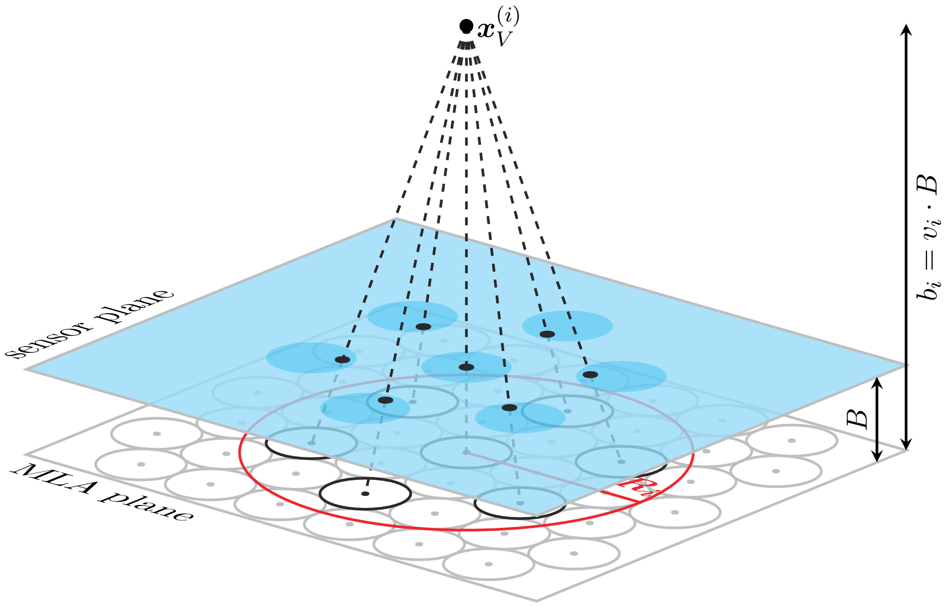


Figure 5.6: Synthesis of the intensity value for a point in the virtual image. Since only points from focused micro images are used for synthesis, an intensity image with a very large DOF can be calculated. This image is called a totally focused image.

The updated depth value of the pixel $\boldsymbol{x}_V^{(i)}$ is then calculated based on the larger one of the two sets N_{sim} and N_{diff} as follows:

$$z_i = \frac{\sum_{k \in N_x} z_k \cdot \frac{w_k}{\sigma_{zk}^2}}{\sum_{k \in N_x} \frac{w_k}{\sigma_{zk}^2}}, \tag{5.38}$$

$$\sigma_{zi}^2 = \frac{\sum_{k \in N_x} w_k}{\sum_{k \in N_x} \frac{w_k}{\sigma_{zk}^2}}. \tag{5.39}$$

Here, N_x is the one of N_{sim} and N_{diff} with the higher cardinality (eq. (5.40)) and w_k defines the respective weight due to the correlation term defined in eq. (5.37).

$$N_x = \begin{cases} N_{\text{sim}} & \text{if } |N_{\text{diff}}| < |N_{\text{sim}}|, \\ N_{\text{diff}} & \text{else.} \end{cases} \tag{5.40}$$

5.7 Intensity Image Synthesis

The regularized virtual depth map can be used to synthesize a totally focused intensity image of the virtual main lens image.

To receive a totally focused image for each virtual image point $\boldsymbol{x}_V^{(i)}$, an intensity value $I_V(\boldsymbol{x}_V^{(i)})$ must be calculated based on the intensities I_{ML} of the respective points in the focused micro images. Prior to any geometric camera calibration there is no knowledge available about the intrinsic camera parameters. Therefore, one has to assume that the micro image centers, which can be detected from a recorded white image, are equivalent to the respective micro lens centers.

The presented depth estimation algorithm obtains depth estimates only for textured regions. However, to synthesize a complete totally focused image, a dense depth map is needed. For image synthesis the depth map is filled with a simple region growing algorithm. Advantage is taken of the fact that regions without depth estimates are homogeneous with respect to the intensity (or else a depth could have been estimated).

To find all micro lenses which see a certain virtual image point $\boldsymbol{x}_V^{(i)}$, a circle with radius R_i, given in eq. (5.41), is defined.

$$R_i = \frac{D_M}{2 \cdot z_i} \tag{5.41}$$

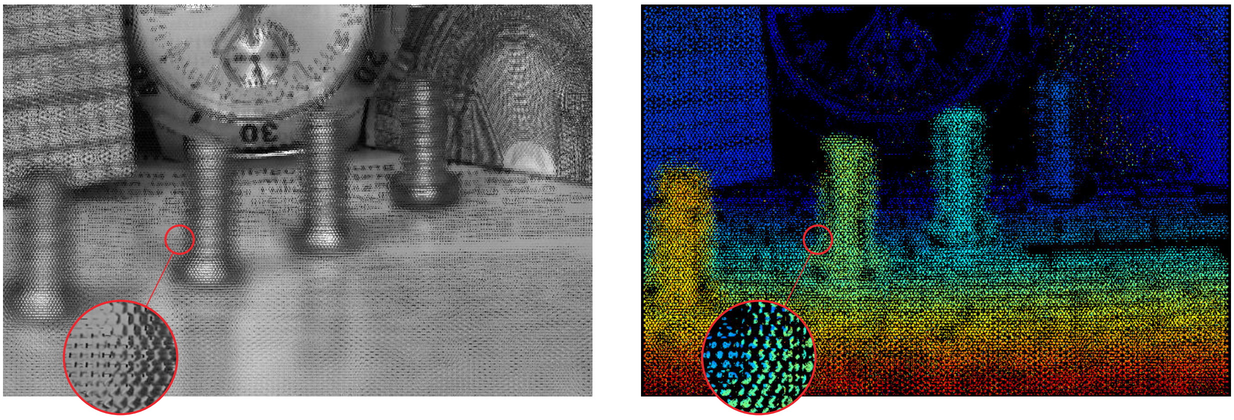
This radius is specified by the diameter of a micro lens D_M and the inverse virtual depth z_i of the virtual image point. For all micro lenses which have their center within the defined circle around the orthogonal projection of the point $\boldsymbol{x}_V^{(i)}$ on the MLA, it is guaranteed that the point is visible in the corresponding micro image. From eq. (5.41) one can see that a point with a large virtual depth results in a large radius, since it is seen by more micro lenses, than a point with a small virtual depth which is close to the MLA. After selecting all micro lenses within the circle, the virtual image point $\boldsymbol{x}_V^{(i)}$ can be projected back to all focused micro images within this region by the definition given in eqs. (5.27) and (5.28). Figure 5.6 visualizes the back projection on the sensor for a single virtual image point $\boldsymbol{x}_V^{(i)}$.

Based on the corresponding intensity in the focused micro images, a weighted mean is calculated:

$$I_V(\boldsymbol{x}_V^{(i)}) = \frac{\sum_k I_{MLk} \cdot h_k^2}{\sum_k h_k^2}. \tag{5.42}$$

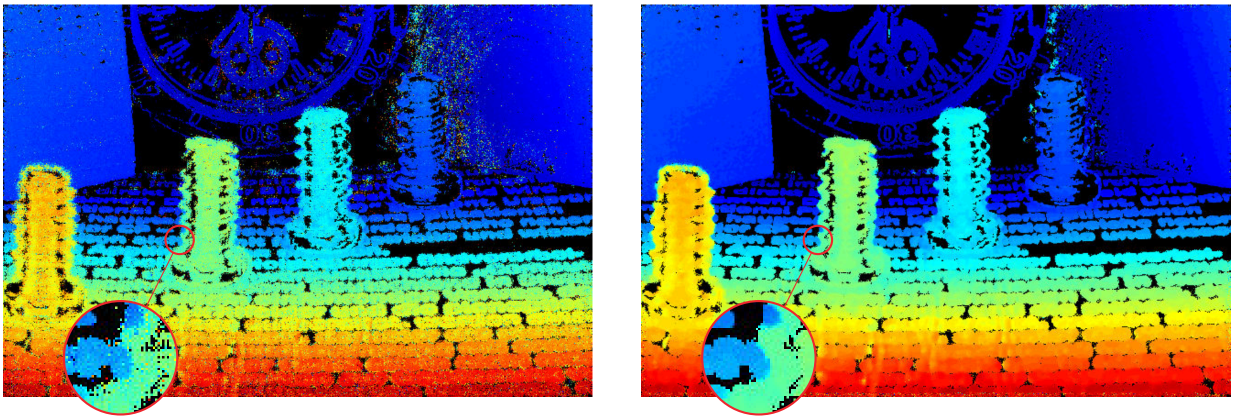
Here, h_k are the respective values from the recorded white image which define the weights for the corresponding intensities I_{MLk}. By calculating a weighted mean, using the information of the white image for each pixel, the optimum signal to noise ratio (SNR) is obtained.

Figure 5.7 shows an example of the complete processing chain of the presented algorithm. Here, Figure 5.7a shows the raw input image, Figure 5.7b shows the estimated inverse virtual depths in the micro images, Figure 5.7c shows depth map in the virtual image space and Figure 5.7d the corresponding regularized version. Figure 5.7e shows the inverse virtual depth variance for each pixel in a logarithmic scale. Finally, Figure 5.7f shows the synthesized totally focused intensity image. Figure 5.7e shows that on average the inverse virtual depth z is estimated with far more confidence for close points than it is for points that are far away. This confirms the assumption which was made above in Section 5.4. According to this assumption σ_z^2 improved with increasing virtual depth v.



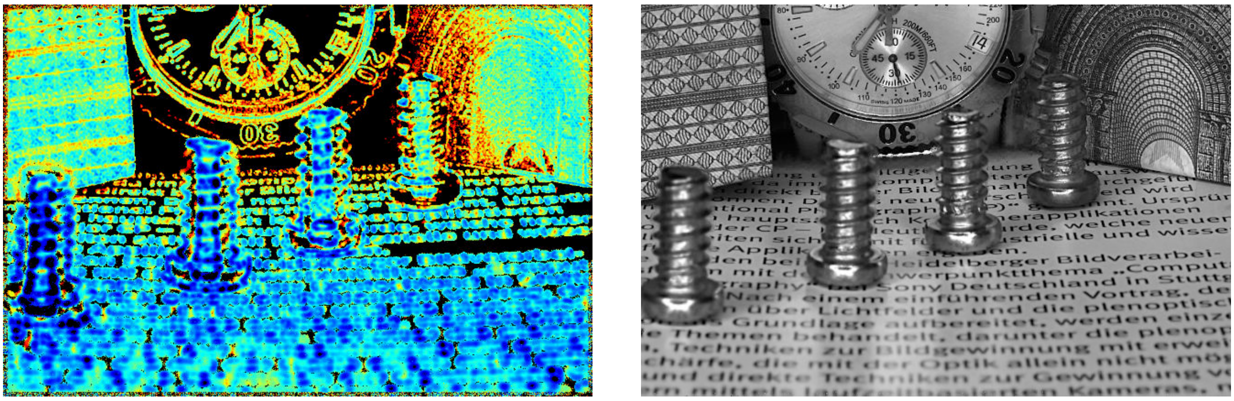
(a) raw image

(b) raw image depth map



(c) virtual image depth map (unfiltered)

(d) virtual image depth map (filtered)



(e) inverse virtual depth variance

(f) totally focused image

Figure 5.7: Exemplary result of the probabilistic depth estimation. Starting from the input raw image in (a) and ending with the synthesized totally focused image in (f). While the filtered depth map (d) contains valid depth information only in structured image regions, homogeneous regions were filled to synthesize a complete intensity image (f). Additionally (e) shows the inverses virtual depth variance corresponding to the filtered depth map in a logarithmic scale. For all color maps red represents large values (virtual depth or variance) and blue represents small values.

6 Plenoptic Camera Calibration

In Chapter 5 a depth estimation approach for focused plenoptic cameras was presented. Depth estimation can be performed without any prior calibration directly on pixel coordinates. Only the micro lens centers have to be known. However, this approach supplies only a map of so-called virtual depths. Without knowing the intrinsic camera model the relationship between the virtual depths and the corresponding real object distances cannot be established. To transform the virtual depths in metric distances, calibration of the camera is necessary. In order to improve the depth estimates obtained from a single image of the plenoptic camera, a mathematical model for the camera which can be used to find multiple view stereo correspondences was derived in Chapter 4. This mathematical model relies on intrinsic camera parameters. For theses parameters either approximate values are given in the datasheet of the camera (e.g. for the focal length), or no values are available at all (e.g. the distance between MLA and sensor), and so, again, calibration is necessary. Furthermore, real lenses deviate from their idealized model. This is a third reason why camera calibration is mandatory to obtain accurate metric 3D reconstructions from the recorded images.

In this chapter plenoptic camera models and calibration approaches at different levels of abstraction are presented. It is started with simple depth conversion functions which allow to transform the virtual depth measurements into metric distances (Section 6.1). Afterwards, it is continued by defining complex camera models which describe the complete projection of a 3D point on the image (Section 6.2). The parameters of these complex models are estimated in a bundle adjustment based calibration approach (Section 6.3).

6.1 Depth Conversion Functions

The simplest way to convert the virtual depth v into a metric object distance z_C is to follow the theoretical relationship between object distance z_C, image distance b_L and virtual depth v, as described in Section 2.4. Figure 2.7 shows that the image distance b_L linearly depends on the virtual depth v, as given in eq. (6.1).

$$b_L = v \cdot B + b_{L0} \tag{6.1}$$

Substituting the image distance b_L in the thin lens equation (eq. (2.7)), by this definition, and rearranging the terms yields the following for the objects distance $z_C(v)$:

$$z_C(v) = \left(\frac{1}{f_L} - \frac{1}{v \cdot B + b_{L0}}\right)^{-1}. \tag{6.2}$$

This function depends on three unknown but constant parameters (f_L, B, and b_{L0}) which have to be estimated. The estimation can be performed from a set of measured calibration points[1] for which the object distance z_C is known.

---

[1] Calibration points are generally checkerboard corners or other interest points detected in the images.

While it will be shown in Section 9.1.1 how one can calculate the object distance z_C based on a set of distance measurements o from an arbitrary but constant position in the line of sight of the plenoptic camera, for the general case one has to assume that z_C cannot be obtained directly. For this case eq. (6.2) has to be extended by the constant distance offset z_0 as follows:

$$o = z_C - z_0 = \left(\frac{1}{f_L} - \frac{1}{v \cdot B + b_{L0}}\right)^{-1} - z_0. \tag{6.3}$$

The constant offset z_0 defines the distance along the optical axis between a reference point from which the distances o are measured (e.g. using a laser rangefinder (LRF)) and the optical main lens center of the camera. Thus, for the general case a forth unknown parameter z_0 has to be added.

Two model-based approaches are presented to estimate this depth conversion function. For comparison also a curve fitting approach is described to approximate the conversion function.

6.1.1 Physical Model

The first approach estimates the unknown parameters of eq. (6.3) explicitly. However, for eq. (6.3) no unique set of the four unknown parameters to define o as a function of v exists. Instead, there are infinite combinations which define the same curve $o(v)$. Thus, to solve this equation, the focal length of the main lens f_L is set to a constant value prior to the estimation. Since f_L is already approximately known from the lens specification, it is set to the respective value. Afterwards, the other three unknown parameters are estimated iteratively.

In the beginning the estimate of the object distance offset $\hat{z}_0$ is set to an initial value. Based on this initial value and the focal length f_L, the corresponding image distance $b_L^{(i)}$ is calculated for each measured object distance $o^{(i)}$. Since the image distance b_L linearly depends on the virtual depth v (see eq. (6.1)), the calculated image distances $b_L^{(i)}$ and the corresponding virtual depths $v^{(i)}$ are used to estimate the Parameters B and b_{L0}. Equations (6.4) to (6.6) show the least squares estimation of the parameters.

$$\begin{bmatrix} \hat{B} \\ \hat{b}_{L0} \end{bmatrix} = \left(\boldsymbol{X}_{Ph}^T \cdot \boldsymbol{X}_{Ph}\right)^{-1} \cdot \boldsymbol{X}_{Ph}^T \cdot \boldsymbol{y}_{Ph} \tag{6.4}$$

$$\boldsymbol{y}_{Ph} = \begin{bmatrix} b_L^{(0)} & b_L^{(1)} & b_L^{(2)} & \cdots & b_L^{(N)} \end{bmatrix}^T \tag{6.5}$$

$$\boldsymbol{X}_{Ph} = \begin{bmatrix} v^{(0)} & v^{(1)} & v^{(2)} & \cdots & v^{(N)} \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}^T \tag{6.6}$$

Based on the estimated parameters $\hat{B}$ and $\hat{b}_{L0}$, for each virtual depth $v^{(i)}$ the corresponding object distance $\hat{z}_C^{(i)}$ is calculated. From the difference between the calculated object distances $\hat{z}_C^{(i)}$ and the measured object distances $o^{(i)}$ the estimated object distance offset $\hat{z}_0$ is updated as given in eq. (6.7).

$$\hat{z}_0 = \frac{1}{N+1} \cdot \sum_{i=0}^{N} \hat{z}_C^{(i)} - o^{(i)} \tag{6.7}$$

By using the updated value $\hat{z}_0$, the image distances $b_L^{(i)}$ are calculated once more and the parameters $\hat{B}$ and $\hat{b}_{L0}$ are updated. The estimation procedure is continued until the variation of $\hat{z}_0$ between two iteration steps is negligibly small.

For this method the focal length of the main lens f_L does not have to be precisely known. There exists an optimum solution for any focal length greater than zero. However, the estimated parameters change when the assumed focal length is changed to a different value. Although, if the specified focal length does not correspond with the real one, the estimated parameters differ from the real physical dimensions. Nevertheless, the estimated set of values is consistent and will not affect the limits of accuracy of the object distance o, when it is calculated on the basis of the estimated virtual depth v.

For the case that z_C can be measured directly, one can skip the iterative optimization. However, it is still recommended to set f_L to a constant value to prevent the estimation from becoming instable due to noise in the measurements of v.

6.1.2 Behavioral Model

The second approach also relies on the function defined in eq. (6.2) or eq. (6.3) respectively. However, this method does not estimate the physical parameters explicitly as done in the first method, but instead estimates the parameters of a function which behaves similarly to the physical model.

In the following the model is formulated based on the function $z_C(v)$ as defined in eq. (6.2). However, exactly the same definition holds for $o(v)$ as given in eq. (6.3). Equation (6.2) can be rearranged to result in eq. (6.8).

$$z_C = z_C \cdot v \cdot \frac{B}{f_L - b_{L0}} + v \cdot \frac{B \cdot f_L}{b_{L0} - f_L} + \frac{b_{L0} \cdot f_L}{b_{L0} - f_L} \tag{6.8}$$

Since the virtual depth v and the object distance z_C both are observable dimensions, a third variable $u = z_C \cdot v$ can be defined. Thus, the term given in eq. (6.9) results from eq. (6.8). Here, the object distance z_C is defined as a linear combination of the measurable variables u and v.

$$z_C = u \cdot c_0 + v \cdot c_1 + c_2 \tag{6.9}$$

The coefficients c_0, c_1, and c_2 are defined as given in eqs. (6.10) to (6.12).

$$c_0 = \frac{B}{f_L - b_{L0}} \tag{6.10}$$

$$c_1 = \frac{B \cdot f_L}{b_{L0} - f_L} \tag{6.11}$$

$$c_2 = \frac{b_{L0} \cdot f_L}{b_{L0} - f_L} \tag{6.12}$$

Since for eq. (6.9) all three variables z_C, v, and u are observable dimensions, the coefficients c_0, c_1, and c_2 can be estimated based on a number of calibration points. The coefficients c_0, c_1, and c_2 can be estimated in a least squares sense as given in eqs. (6.13) to (6.15).

$$\begin{bmatrix} \hat{c}_0 \\ \hat{c}_1 \\ \hat{c}_2 \end{bmatrix} = \left(\boldsymbol{X}_{Be}^T \cdot \boldsymbol{X}_{Be}\right)^{-1} \cdot \boldsymbol{X}_{Be}^T \cdot \boldsymbol{y}_{Be} \tag{6.13}$$

$$\boldsymbol{y}_{Be} = \begin{bmatrix} z_C^{(0)} & z_C^{(1)} & \cdots & z_C^{(N)} \end{bmatrix}^T \tag{6.14}$$

$$\boldsymbol{X}_{Be} = \begin{bmatrix} z_C^{(0)} v^{(0)} & z_C^{(1)} v^{(1)} & \cdots & z_C^{(N)} v^{(N)} \\ v^{(0)} & v^{(1)} & \cdots & v^{(N)} \\ 1 & 1 & \cdots & 1 \end{bmatrix}^T \tag{6.15}$$

After rearranging eq. (6.9) the object distance z_C can be described as a function of the virtual depth v and the estimated parameters c_0, c_1, and c_2 as given in eq. (6.16).

$$z_C(v) = \frac{v \cdot c_1 + c_2}{1 - v \cdot c_0} \tag{6.16}$$

The same formulation can be obtained from the function defined in eq. (6.3). The rearranged function has the same structure as eq. (6.9) with the difference that z_C is substituted by o, resulting in $u = o \cdot v$. Furthermore, the definitions of the constant coefficients c_1, c_2, and c_3 change respectively.

6.1.3 Polynomial based Curve Fitting

The third method is presented for the purpose of comparison and is a common curve fitting approach. It approximates the function between the virtual depth v and the object distance z_C while disregarding the function defined in eq. (6.2).

It is known that any differentiable function can be represented by a Taylor-series, i.e. by a polynomial of infinite order. Hence, in the approach presented here, the function which describes the object distance z_C depending on the virtual depth v will be defined as a polynomial as well. A general definition of this polynomial is given in eq. (6.17).

$$z_C(v) \approx \sum_{k=0}^{K} l_k \cdot (v)^k \tag{6.17}$$

The polynomial coefficients l_0 to l_K are estimated based on a set of measured calibration points in a least squares senses, as given in eqs. (6.18) to (6.20).

$$\begin{bmatrix} \hat{l}_0 \\ \hat{l}_1 \\ \vdots \\ \hat{l}_K \end{bmatrix} = \left(\boldsymbol{X}_{Pol}^T \cdot \boldsymbol{X}_{Pol}\right)^{-1} \cdot \boldsymbol{X}_{Pol}^T \cdot \boldsymbol{y}_{Pol} \tag{6.18}$$

$$\boldsymbol{y}_{Pol} = \begin{bmatrix} z_C^{(0)} & z_C^{(1)} & \cdots & z_C^{(N)} \end{bmatrix}^T \tag{6.19}$$

$$\boldsymbol{X}_{Pol} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ v^{(0)} & v^{(1)} & \cdots & v^{(N)} \\ \left(v^{(0)}\right)^2 & \left(v^{(1)}\right)^2 & \cdots & \left(v^{(N)}\right)^2 \\ \vdots & \vdots & \ddots & \vdots \\ \left(v^{(0)}\right)^K & \left(v^{(1)}\right)^K & \cdots & \left(v^{(N)}\right)^K \end{bmatrix}^T \tag{6.20}$$

The function can also be defined with respect to o instead of z_C, as it can be done for the previous two methods.

For this method a trade-off between the accuracy of the approximated function and the order of the polynomial has to be found. A high order of the polynomial results in a higher effort in calculating the object distance from the virtual depth. Furthermore, for high orders the matrix inversion as defined in eq. (6.18) results in numerical inaccuracies. From eq. (6.20) one can see that with an increase in order, the range of the values in the matrix $\boldsymbol{X}_{Pol}$, and therefore in the approximated autocorrelation matrix $\boldsymbol{X}_{Pol}^T \boldsymbol{X}_{Pol}$, rises exponentially. For such cases a different method for solving the least squares problem must be used (e.g. Cholesky decomposition).

The three methods described here define functions which can be used to transform the estimated virtual depth v into a metric object distance z_C. However, these functions do not define a complete camera model. Depending on the application it might be sufficient to combine such a depth conversion function with a standard pinhole camera model to approximate the projection of a plenoptic camera. This holds true especially for narrow FOV setups where only little perspective distortion is present in the images. Such a model was presented in Zeller et al. [2014d, 2016b]. There is was shown that it is possible to perform odometry estimation on a small scale using such a model (Zeller et al. [2015c, 2016b]).

6.2 Intrinsic Models for Focused Plenoptic Cameras

While Section 6.1 presented different methods for converting the estimated virtual depths into metric distances, in this section full intrinsic models for focused plenoptic cameras are presented. As the plenoptic camera models were gradually refined and adapted for the task of plenoptic odometry estimation during this thesis, ultimately three models at different levels of abstraction are presented. The final model is a highly accurate focused plenoptic camera model, which is used in the Direct Plenoptic Odometry (DPO) algorithm (presented in Chapter 7).

As in all existing plenoptic camera models, the main lens is modeled as a thin lens, while the micro lenses are modeled as pinholes. In reality of course, the main lens does not satisfy the model of a thin lens but is, in fact, a thick lens. Nevertheless, this simplification does not affect the projection from object space to virtual image space, when both spaces are considered to be decoupled from each other. By decoupled it is meant that the virtual image space is defined with respect to the image sided principal plane of the thick main lens, while the object space is defined with respect to the object sided principle plane. For the thin lens model it is considered that both planes are in the same location. Furthermore, lens distortion models are introduced to model imperfections of the main lens. Another effect, which is described for the first model presented here (Section 6.2.1), is a plenoptic sensor (i.e. image sensor + MLA) which is tilted with respect to the main lens, rather than being parallel to it.

6.2.1 Virtual Image Projection Model

In Chapter 5 it was described how a virtual depth map as well as a corresponding totally focused image of the virtual main lens image can be calculated without performing any calibration (only micro lens centers have to be known). Therefore, the first model defines the projection from object space to the virtual image which is calculated prior to the calibration. Hence, unlike to regular cameras, where a 3D object space is projected to a 2D image plane, in a focused plenoptic camera a 3D object space is projected to a 3D image space.

In the previous chapters the absolute dimensions of image and object coordinates were not of interest. However, the absolute dimension of a point is important for metric calibration. While object points have metric dimensions, for the virtual image metric coordinates $\boldsymbol{x}'_V = [x'_V, y'_V, z'_V = b, 1]^T$ as well as the dimensionless coordinates $\boldsymbol{x}_V = [x_V, y_V, v, 1]^T$, which are given in pixels and virtual depths respectively, are defined. The origin for the metric virtual image coordinates $\boldsymbol{x}'_V$ is located at the intersection of the optical axis of the main lens with the MLA plane. For $\boldsymbol{x}_V$ the origin is shifted in x and y direction to the upper left pixel of the virtual image.

Using the ideal thin lens model for the main lens, a virtual image point $\boldsymbol{x}'_V$ can be calculated based on the corresponding object point $\boldsymbol{x}_C$ as follows:

$$x'_V = \frac{b_L}{z_C} \cdot x_C \qquad y'_V = \frac{b_L}{z_C} \cdot y_C \qquad z'_V = b = b_L - b_{L0}, \tag{6.21}$$

or in a compact matrix notation based on homogeneous coordinates, as given in eq. (6.22).

$$\begin{aligned} \eta \cdot \boldsymbol{x}'_V &= \boldsymbol{K} \cdot \boldsymbol{x}_C \\ \eta \cdot \begin{bmatrix} x'_V \\ y'_V \\ z'_V \\ 1 \end{bmatrix} &= \begin{bmatrix} b_L & 0 & 0 & 0 \\ 0 & b_L & 0 & 0 \\ 0 & 0 & b & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_C \\ y_C \\ z_C \\ 1 \end{bmatrix} \end{aligned} \tag{6.22}$$

Unlike the projection in a standard pinhole camera model, the image distance b_L (equivalent to the focal length f in a pinhole camera) is not constant but depends on the corresponding object distance z_C. The image distance results from the thin lens (eq. (2.7)) as follows:

$$b_L = \left(\frac{1}{f_L} - \frac{1}{z_C} \right)^{-1} = b + b_{L0} = B \cdot v + b_{L0}. \tag{6.23}$$

Here, the parameters b, b_L, and b_{L0} conform to the ones shown in Figure 2.7. The introduced scaling factor η in the matrix notation is simply the object distance $\eta = z_C$, as it is obtained from the fourth row of eq. (6.22).

To receive the virtual image coordinates in the dimensions in which they are measured $\boldsymbol{x}_V = [x_V, y_V, v, 1]^T$, a scaling of the axis as well as a translation along the x- and y-axis has to be performed, as defined in eq. (6.24).

$$\begin{aligned} \boldsymbol{x}_V &= \boldsymbol{K}_S \cdot \boldsymbol{x}'_V \\ \begin{bmatrix} x_V \\ y_V \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} s_x^{-1} & 0 & 0 & c_{Lx} \\ 0 & s_y^{-1} & 0 & c_{Ly} \\ 0 & 0 & B^{-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x'_V \\ y'_V \\ z'_V \\ 1 \end{bmatrix} \end{aligned} \tag{6.24}$$

Here s_x and s_y define the size of a pixel, while B is the distance between MLA and sensor.

By combining the matrices $\boldsymbol{K}$ and $\boldsymbol{K}_S$ one can define the complete transformation from camera coordinates $\boldsymbol{x}_C$ to virtual image coordinates $\boldsymbol{x}_V$, as given in eq. (6.25).

$$\eta \cdot \boldsymbol{x}_V = \boldsymbol{K}_S \cdot \boldsymbol{K} \cdot \boldsymbol{x}_C \tag{6.25}$$

Since the pixel size (s_x, s_y) for a certain sensor is generally known, there are five parameters left to be estimated. These parameters are the principal point $\boldsymbol{c}_L = [c_{Lx}, c_{Ly}]^T$ expressed in pixels, the distance B between MLA and sensor, the distance b_{L0} between the main lens' principal plane and the MLA, and the focal length f_L of the main lens.

Distortion Models

Until now the projection of an object point $\boldsymbol{x}_C$ to the corresponding virtual image point $\boldsymbol{x}_V$ is defined without considering any imperfection in the lens or the setup. To account for these imperfections, a lens distortion model which corrects the position of a virtual image point $\boldsymbol{x}_V$ in x, y and z direction is defined.

To implement the distortion model, another coordinate system with the homogeneous coordinates $\boldsymbol{x}_I = (x_I, y_I, z_I, 1)$ is defined. The coordinates $\boldsymbol{x}_I$ actually define the pinhole projection which would be performed in a conventional camera. Here, an image point $\boldsymbol{x}_I$ is defined as the intersection of a central ray through the main lens with a common image plane which is parallel to the main lens. In the presented model the image plane is chosen to be the same as the MLA plane. Thus, by splitting up the matrix $\boldsymbol{K}$ into $\boldsymbol{K}_I$ and $\boldsymbol{K}_V$ (eq. (6.26)) the image coordinates $\boldsymbol{x}_I$ are obtained as defined in eq. (6.27).

$$\begin{aligned}\boldsymbol{K} &= \boldsymbol{K}_V \cdot \boldsymbol{K}_I \\ &= \begin{bmatrix} \frac{b_L}{b_{L0}} & 0 & 0 & 0 \\ 0 & \frac{b_L}{b_{L0}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} b_{L0} & 0 & 0 & 0 \\ 0 & b_{L0} & 0 & 0 \\ 0 & 0 & b & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\end{aligned} \tag{6.26}$$

$$\eta \cdot \boldsymbol{x}_I = \boldsymbol{K}_I \cdot \boldsymbol{x}_C \tag{6.27}$$

By introducing the image coordinates $\boldsymbol{x}_I$ one is able to define a lateral distortion model on this image plane, similar to the one for a regular camera. From eqs. (6.26) and (6.27) one can see that the z-coordinates in the spaces defined by $\boldsymbol{x}_I$ and $\boldsymbol{x}'_V$ are equivalent ($z_I = z'_V = b$).

By applying the distortion at the intersection of all central rays with a common plane, a constant distortion is defined for each central ray through the main lens respectively. This differs from previous models (Johannsen et al. [2013]; Heinze et al. [2015, 2016]), where distortion is defined by performing an orthogonal projection on a plane, instead of a central perspective projection.

In the following the distortion model is defined. This model is split into lateral distortion (similar to a regular camera) and depth distortion. A depth distortion is needed since, as described in Chapter 5, virtual depth estimation is performed prior to any calibration and therefore based on distorted image coordinates. Hence, the depth estimates will also be affected by the distortion and must be corrected.

Lateral Distortion Lateral distortion is defined based on the well known model of Brown [1966]. This model considers radial symmetric as well as tangential distortion.

The radial symmetric distortion is defined by a polynomial along the radius r in polar coordinates, as defined in eq. (6.28).

$$\varDelta r_{rad} = A_0 r^3 + A_1 r^5 + A_2 r^7 \tag{6.28}$$

Here, the radius r is the Euclidean distance of a point $\boldsymbol{x}_I$ on the image plane to the main lens' principal point:

$$r = \sqrt{x_I^2 + y_I^2}. \tag{6.29}$$

This results in the Cartesian correction terms $\varDelta x_{rad}$ and $\varDelta y_{rad}$ as given in eqs. (6.30) and (6.31).

$$\varDelta x_{rad} = x_I \frac{\varDelta r_{rad}}{r} = x_I \cdot \left(A_0 r^2 + A_1 r^4 + A_2 r^6\right) \tag{6.30}$$

$$\varDelta y_{rad} = y_I \frac{\varDelta r_{rad}}{r} = y_I \cdot \left(A_0 r^2 + A_1 r^4 + A_2 r^6\right) \tag{6.31}$$

For tangential distortion the terms given in eqs. (6.32) and (6.33) are defined.

$$\varDelta x_{tan} = B_0 \cdot \left(r^2 + 2x_I^2\right) + 2B_1 x_I y_I \tag{6.32}$$

$$\varDelta y_{tan} = B_1 \cdot \left(r^2 + 2y_I^2\right) + 2B_0 x_I y_I \tag{6.33}$$

Based on the correction terms the distorted image coordinates x_{Id} and y_{Id} are calculated from the ideal projection as follows:

$$x_{Id} = x_I + \varDelta x_{rad} + \varDelta x_{tan}, \tag{6.34}$$

$$y_{Id} = y_I + \varDelta y_{rad} + \varDelta y_{tan}. \tag{6.35}$$

Depth Distortion Models to compensate for the distortion which occurs in virtual depth maps have already been presented. Johannsen et al. [2013], for instance, present a five parameter polynomial based model, while Heinze et al. [2016] reduced the number of parameters to three. For the camera model presented in this section, the depth distortion model of Heinze et al. [2016] was chosen. This depth distortion is defined as follows:

$$\varDelta v = (1 + D_0 \cdot v) \cdot \left(D_1 \cdot r^2 + D_2 \cdot r^4\right). \tag{6.36}$$

Here, $\varDelta v$ defines the virtual depth correction term which depends on the virtual depth v, the radius r and three distortion parameters D_0, D_1, and D_2. The distorted depth is calculated as follows:

$$v_d = v + \varDelta v \qquad \text{or} \tag{6.37}$$

$$z_{Id} = b_d = b + B\varDelta v. \tag{6.38}$$

In contrast to the publications of Johannsen et al. [2013] and Heinze et al. [2016], lateral as well as depth distortion are defined based on the undistorted coordinates. This is the conventional definition as described by Brown [1966].

Since the distortion is exclusively defined by additive terms, they can be represented by a translation matrix $\boldsymbol{K}_D$ as defined in eq. (6.39).

$$\boldsymbol{K}_D = \begin{bmatrix} 1 & 0 & 0 & \varDelta x_{rad} + \varDelta x_{tan} \\ 0 & 1 & 0 & \varDelta y_{rad} + \varDelta y_{tan} \\ 0 & 0 & 1 & B\varDelta v \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.39}$$

This matrix consists of up to eight distortion parameters (A_0, A_1, A_2, B_0, B_1, D_0, D_1, and D_2), which have to be estimated in the calibration process.

Sensor Tilting

A sensor tilted with respect to the main lens plane can be modeled by a rotation around the x- and y-axis of the virtual image space. Therefore, the matrix $\boldsymbol{T}$ has to be defined as follows:

$$\begin{aligned} \boldsymbol{T} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ \sin(\alpha)\sin(\beta) & \cos(\alpha) & -\sin(\alpha)\cos(\beta) & 0 \\ -\cos(\alpha)\sin(\beta) & \sin(\alpha) & \cos(\alpha)\cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{6.40}$$

Complete Projection Model

Taking into consideration both the ideal projection and the distortion, the complete projection of an object point, in camera coordinates $\boldsymbol{x}_C$ to the corresponding virtual image point in distorted coordinates $\boldsymbol{x}_{Vd}$, which are actually measured, is defined as follows:

$$\eta \boldsymbol{x}_{Vd} = \boldsymbol{K}_S \cdot \boldsymbol{T} \cdot \boldsymbol{K}_V \cdot \boldsymbol{K}_D \cdot \boldsymbol{K}_I \cdot \boldsymbol{x}_C. \tag{6.41}$$

6.2.2 Micro Images Projection Model

In Section 6.2.1 the plenoptic camera was considered to be, more or less, a RGB-D sensor, which supplies a virtual depth map and a corresponding totally focused image from the same perspective. However, the camera model presented in Section 4.1 defines a projection from object space directly to the micro images on the sensor. Furthermore, as the intensities in the totally focused image are not directly measured but calculated based on noisy virtual depth estimates, correspondences established on the basis of the totally focused image will be affected by that additional noise.

Here a model is presented which defines the projection from object space directly to the micro images on the sensor. Therefore, this model is basically wholly defined in Section 4.1 and will only be extended by a distortion model.

Intrinsic Parameters

The plenoptic camera model consists of the following intrinsic parameters:

- f_L – focal length of the main lens
- b_{L0} – distance between MLA and main lens
- B – distance between MLA and image sensor
- $\boldsymbol{c}_L$ – principal point (intersection of the optical axis of the main lens with the image sensor)
- $\boldsymbol{c}_{ML}^{(k)}$ – micro lens centers ($k \in \{1, 2, 3, \ldots\}$)

Thus, the intrinsic parameters without distortion are the same as for the previous model (Section 6.2.1), where the micro lens centers are also needed to perform virtual depth estimation and to synthesize the totally focused image.

Distortion Model

In contrast to the previous model (Section 6.2.1), for this model, an object point is projected directly onto a 2D micro image or rather multiple micro images. Hence, the distortion model can also be a applied to a 2D plane, and more importantly, before depth estimation is performed.

Due to the fact that depth estimation is performed based on corrected image and micro lens coordinates, the virtual depth estimates will not be affected by distortion. Therefore, it is sufficient to define only a lateral distortion model.

Again, the same lateral distortion terms as for the previous model (Section 6.2.1) are defined. A practical aspect of this model is that it consists only of additive distortion terms which depend on the undistorted coordinates. Due to the small size of a micro image, it suffices to consider the distortion to be constant within a single micro image. Therefore only the micro lens centers $\boldsymbol{c}_{ML}$ are corrected and not the point coordinates $\boldsymbol{x}_{ML}$ in the respective micro image.

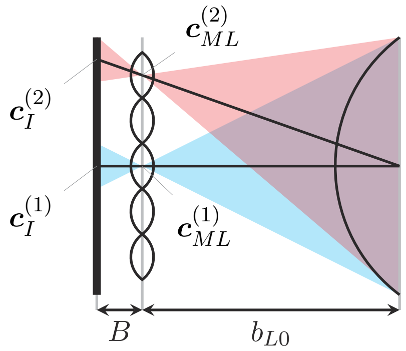


Figure 6.1: Relationship between micro image and micro lens centers. While the micro image centers $\boldsymbol{c}_I$ can be estimated from a recorded white image, the micro lens centers $\boldsymbol{c}_{ML}$ can only be calculated if the intrinsic model of the camera is known.

Furthermore, the effect of a tilted plenoptic sensor is not considered explicitly in order to maintain the structure of the model as described in Section 4.1. In this model the MLA is considered to be parallel to the main lens and therefore the projected MLA (virtual camera array) lies on a common plane at distance z_{C0} to the main lens (see Figure 4.2).

6.2.3 Extended Micro Images Projection Model

For all previous models as well as for the depth estimation presented in Chapter 5 it was considered that the micro lens centers $\boldsymbol{c}_{ML}$ are known. There are various methods for estimating the centers of the micro images formed on the sensor when placing a white balance filter in front of the camera (e.g. Dansereau et al. [2013]; Cho et al. [2013]; Hog et al. [2017]). In the previous chapters we assumed the center of a micro image to also be the center of the corresponding micro lens without further examination of this assumption.

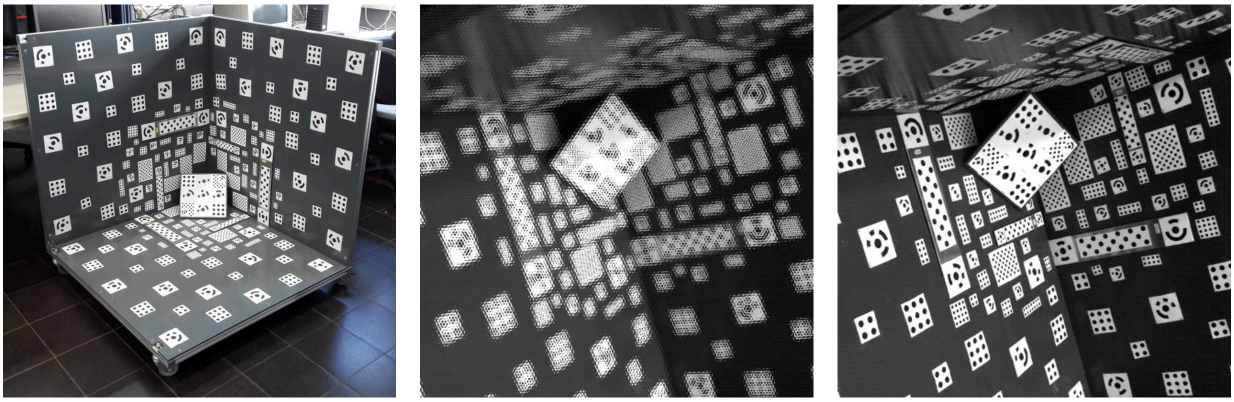
This claim is, in fact, not true, as can easily be seen in Figure 6.1. However, as will be shown later in the evaluation, this assumption can be sustained as long as one is not interested in the exact camera position. The error made in the position of a micro lens center is directly projected to an error in the camera position and therefore results in a constant bias in the estimated camera position.

To account for this offset between micro lens and micro image center the model presented in the previous Section 6.2.2 is further extended. In addition to the micro lens centers $\boldsymbol{c}_{ML}$ the micro image centers are by $\boldsymbol{c}_I$.

In Section 4.1 the micro lens centers were defined by 3D coordinates in the image space with its origin at the intersection of the optical axis of the main lens with the main lens plane. Hence, the micro image centers are defined in the same coordinate system. Thus, the following relationship between micro image and micro lens centers results:

$$\boldsymbol{c}_I = \begin{bmatrix} c_{Ix} \\ c_{Iy} \\ B + b_{L0} \end{bmatrix} = \boldsymbol{c}_{ML} \cdot \frac{b_{L0} + B}{b_{L0}} = \begin{bmatrix} c_{MLx} \\ c_{MLy} \\ b_{L0} \end{bmatrix} \cdot \frac{b_{L0} + B}{b_{L0}}. \tag{6.42}$$

Again the same distortion model as defined in Section 6.2.2 is applied. However, instead of defining lateral distortion on the MLA plane, it is defined directly on the sensor plane. While in Section 6.2.2 the distortion was applied only to the micro lens centers, here, distortion is applied to each raw image point on the sensor.



(a) 3D calibration target (b) recorded raw image (c) totally focused image

Figure 6.2: 3D target used for plenoptic camera calibration. (a) Entire calibration target (b) Sample raw image recorded by the plenoptic camera. (c) Totally focused image synthesized from the sample raw image.

The used radial symmetric distortion model (eqs. (6.28) to (6.30)) is defined such that for a radius $r = 0$ the correction term Δr_{rad} is zero. Most real lenses generally show a barrel distortion behavior, which means that the correction term Δr_{rad} increases with an increasing radius r. This means that in the undistorted image pixel sizes will be enlarged with increasing radius r. Such an enlargement of the pixel size will also affect the estimate of B, which therefore will differ from its real physical value.

This could be compensated by adapting the distortion model in such a way that the integration over all correction terms will be zero and thus the average pixel size in the undistorted image will be equal to the real pixel size. This can be realized by introducing a radius offset r_0 as described by Luhmann et al. [2014] (Chapter 3.3.3). However, the overall projection accuracy does not change by the introduction of r_0. Only the FOV of the rectified image might be affected.

Compared to the model presented in Section 6.2.2, the number of parameters to be estimated stays exactly the same. However, by refining the micro lens centers $\boldsymbol{c}_{ML}$ with respect to the micro image centers $\boldsymbol{c}_I$, one is able to compensate a position bias in the estimation, as will be shown by the results in Section 9.2.2. Furthermore, while previously it was assumed that the micro lenses look straight ahead, now one can see that outer micro lenses actually squint.

6.3 Plenoptic Camera based Bundle Adjustment

In contrast to all existing calibration methods for plenoptic cameras, here the calibration is performed in a bundle adjustment based on a set of unordered 3D points. The 3D structure of the calibration target results in a better conditioning of the optimization problem when compared to a standard 2D target. Furthermore, the calibration does not rely on any model constraints (e.g. all points lying on a plane). Figure 6.2a shows the 3D target which is used for calibration.

The camera calibration consists of several steps. First, a set of images is recorded, showing the calibration target from as many different perspectives as possible (see example in Figure 6.2). In an initial stage, the marker points of the calibration target are detected are assigned to each other. In addition, initial solutions for all model parameters (intrinsic and extrinsic camera parameters, and 3D coordinates of the calibration markers) are established. This initial stage is the same for all three presented camera models. Next, the energy function is defined depending on the

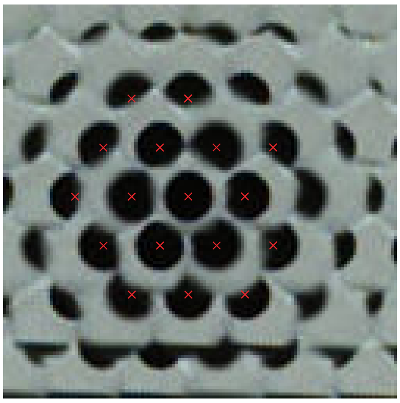


Figure 6.3: Single calibration marker mapped to multiple micro images. Red crosses visualize the detected center of the marker in the various micro images.

respective camera models and is optimized with respect to the intrinsic and extrinsic camera parameters as well as the 3D coordinates of the calibration markers.

In the following, separately the definitions of the energy functions, both for the virtual image based model as well as the two models which are based on micro images, are presented.

For the entire calibration approach the micro image centers are assumed to be already estimated based on a recorded white image.

6.3.1 Initialization of Intrinsic and Extrinsic Parameters

In the initial stage, virtual depth maps are estimated and the corresponding totally focused images are synthesized based on the methods described in Chapter 5 for the set of recorded images. In the synthesized totally focused images one are able to detect the marker points of the calibration target. Furthermore, initial point correspondences can be found for a set of uniquely coded marker points (see Figure 6.2c).

Even though the marker point positions in the micro images are needed for the two later models, the points are still detected in the totally focused images and then are project back onto the micro images. Algorithms which are able to detect the round coded and uncoded marker points reliably in the recorded images already exist. The detection in the micro images, however, would be difficult since the markers of the calibration target are for the most part only partially mapped to the micro images. Figure 6.3, by way of example, shows a single calibration marker mapped to multiple micro images. The coordinates of the marker detected in the totally focused image and projected to the micro images are shown as red crosses.

After the marker detection, intrinsic and extrinsic camera parameters as well as the 3D coordinates of the calibration markers are initialized on the basis of the points detected in the set of totally focused images. While one is able to build a closed form solution for such a structure from motion (SfM) problem based on an uncalibrated monocular camera, this is not the case for the model of a plenoptic camera, where the camera matrix $\boldsymbol{K}$ depends on the object distance z_C. Therefore, the fundamental matrix for a pair of virtual images will also depend on the object distance z_C and thus cannot be solved without knowing that distance. This makes the initialization of the optimization problem more difficult. To initialize the bundle adjustment, it is first considered that the totally focused image of the plenoptic camera results from a regular pinhole camera. This way, the SfM problem is solved by using a standard photogrammetric software. For initialization this approximation is sufficient, as only a rough estimate of all parameters is needed.

In addition, initial parameters for B and b_{L0} are obtained by solving the physical model defined in Section 6.1.1. In this way, initial values are received for all intrinsic and extrinsic camera parameters as well as the 3D object points.

Along with the initial parameters, the photogrammetric software already provides the correct correspondences for all marker points across all recorded images.

In the following the validity of using the pinhole camera model for initialization, at least for $f_L \ll z_C$, is proved based on a specific example. The camera setup is as follows:

- main lens focal length $f_L = 16\,\mathrm{mm}$
- size of the totally focused image: $1024\,\mathrm{pixel} \times 1024\,\mathrm{pixel}$
- principal point in the image center

The calibration target covers a depth range from $z_{C\min} = 500\,\mathrm{mm}$ up to $z_{C\max} = 1500\,\mathrm{mm}$. Based on the thin lens equation the following maximum, minimum and average image distances can be calculated:

- $b_{L\max} = 16.53\,\mathrm{mm}$
- $b_{L\min} = 16.17\,\mathrm{mm}$
- $\bar{b}_L = 16.35\,\mathrm{mm}$

Thus, one receives a maximum projection error between the plenoptic and pinhole camera model as follows:

$$\begin{aligned} \mathit{\Delta} x_{\max} &= x_{\max} \cdot \left(\frac{b_{L\max}}{\bar{b}_L} - 1 \right) \\ &= x_{\max} \cdot \left(1 - \frac{b_{L\min}}{\bar{b}_L} \right) = 5.63\,\mathrm{pixel}. \end{aligned} \tag{6.43}$$

Here, the image plane of the pinhole camera model is considered to be at a distance $\bar{b}_L$ from the main lens. The maximum image coordinate $x_{\max}$ is considered to be at the image boundary and therefore at $x_{\max} = 512\,\mathrm{pixel}$. As can be seen from eq. (6.43), the error made for the initialization is in the range of only a few pixels.

For larger object distances the error decreases further (i.e. for $z_{C\min} = 1500\,\mathrm{mm}$ and $z_{C\max} = 2500\,\mathrm{mm}$ it follows $\mathit{\Delta} x_{\max} = 1.11\,\mathrm{pixel}$). The variation in the average image distance $\bar{b}_L$ results in a small scaling error of the recorded image and therefore in a bias regarding the estimated object distance. For $f_L \ll z_C$ it follows $b_L \approx f_L$ and therefore the estimated principal distance of the pinhole camera model provides a good initialization for the main lens focal length f_L of the plenoptic camera.

6.3.2 Bundle Adjustment on Virtual Image Space

To estimate the exact camera parameters for the virtual image based model (Section 6.2.1), we formulate the following energy function $E(\mathit{\Pi}, \mathit{\Xi})$ over all recorded frames:

$$E(\mathit{\Pi}, \mathit{\Xi}) = \sum_{i=1}^{N} \sum_{j=1}^{M} \left\| \boldsymbol{r}_{(i,j)} \right\|^2 \cdot \theta_{(i,j)}. \tag{6.44}$$

Here, $\varPi$ is the set of all camera parameters (intrinsic and distortion parameters) and $\varXi$ is the set of all camera poses ($\varXi := \{\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_M\}$) with respect to a world frame. Each calibration point in the totally focused image results in a 3D residual vector $\boldsymbol{r}_{(i,j)}$ in the virtual image space. Furthermore, $\theta_{(i,j)}$ is a masking function which is 1 if the i-th object point is visible in the j-th view of the plenoptic camera, or else is zero. To be robust against outliers, $\theta_{(i,j)}$ can also be extended by any robust loss function.

The projection of the i-th object point in world coordinates $\boldsymbol{x}_W^{(i)}$ to the corresponding camera coordinates of the j-th view $\boldsymbol{x}_C^{(i,j)}$ is defined by a rigid body transformation $\boldsymbol{G}(\boldsymbol{\xi}_j) \in \mathrm{SE}(3)$, as follows:

$$\boldsymbol{x}_C^{(i,j)} = \boldsymbol{G}(\boldsymbol{\xi}_j) \cdot \boldsymbol{x}_W^{(i)}. \tag{6.45}$$

Furthermore, the projection on the virtual image is defined in eq. (6.41). The complete projection is combined in the warping function $\pi_V(\cdot)$:

$$\boldsymbol{x}_{Vd}^{(i,j)} = \pi_V\left(\boldsymbol{G}(\boldsymbol{\xi}_j)\boldsymbol{x}_W^{(i)}, \varPi\right). \tag{6.46}$$

In the following a virtual image point $\boldsymbol{x}_{Vd}$, which was calculated based on the projection given in eq. (6.46) is denoted by the coordinates x_{proj}, y_{proj}, and v_{proj}, while the corresponding reference point measured from the recorded image is denoted by the coordinates x_{meas}, y_{meas}, and v_{meas}.

Instead of defining the residual vector $\boldsymbol{r}_{(i,j)}$ as just the difference between the projected and the measured virtual image point, it is defined as follows:

$$\boldsymbol{r}_{(ij)} = \begin{bmatrix} r_x^{(i,j)} & r_y^{(i,j)} & r_v^{(i,j)} \end{bmatrix}^T, \qquad \text{with} \tag{6.47}$$

$$r_x^{(i,j)} = x_{\text{proj}}^{(i,j)} - x_{\text{meas}}^{(i,j)}, \tag{6.48}$$

$$r_y^{(i,j)} = y_{\text{proj}}^{(i,j)} - y_{\text{meas}}^{(i,j)}, \tag{6.49}$$

$$r_v^{(i,j)} = \left(\frac{1}{v_{\text{proj}}^{(i,j)}} - \frac{1}{v_{\text{meas}}^{(i,j)}}\right) \cdot v_{\text{proj}}^{(i,j)} \cdot \omega. \tag{6.50}$$

The residual of the virtual depth $r_v^{(i,j)}$ is defined as given in eq. (6.50), since the virtual depth v is inversely proportional to the disparity μ estimated from the micro images (see eq. (2.12)). Thus, the inverse virtual depth can be considered to be Gaussian distributed (see eq. (5.2)). As already stated in Section 6.2.1, the baseline distance $\varDelta c_{ML}$ is, on average, proportional to the estimated virtual depth v. Thus, the difference in the inverse virtual depth is scaled by the virtual depth v itself. The parameter ω is simply a constant factor which defines the weight between $r_x^{(i,j)}$, $r_y^{(i,j)}$, and $r_v^{(i,j)}$. In general, the virtual depth residual $r_v^{(i,j)}$ must be down weighted with respect to the lateral residuals $r_x^{(i,j)}$ and $r_y^{(i,j)}$, because compared to the noise in the coordinates of the markers detected in the totally focused images, the noise in the virtual depth is higher by orders of magnitude.

The optimal parameters ($\hat{\varPi}$, $\hat{\varXi}$) are the ones which minimize the cost function $E(\varPi, \varXi)$:

$$\{\hat{\varPi}, \hat{\varXi}\} = \arg\min E(\varPi, \varXi). \tag{6.51}$$

This highly nonlinear optimization problem is solved using the Levenberg-Marquardt algorithm with respect to $\varPi$ and $\varXi$. The initialization of $\varPi$ and $\varXi$ has to guarantee that the optimization starts in the convex region around the optimum solution $E(\hat{\varPi}, \hat{\varXi})$.

For this first model the coordinates of the 3D points $\boldsymbol{x}_W^{(i)}$ are not optimized in the final bundle adjustment, since this way the optimization becomes more robust. The results presented in Section 9.2.2 will show that the coordinates $\boldsymbol{x}_W^{(i)}$ are already estimated with high accuracy during the initialization.

6.3.3 Full Bundle Adjustment on Micro Images

To estimate the exact camera parameters for the two later models (Sections 6.2.2 and 6.2.3), we define the following energy function, quite similar to eq. (6.44):

$$E(\mathit{\Pi}, \mathit{\Xi}, P) = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{L} \left\| \boldsymbol{r}_{(i,j,k)} \right\|^2 \cdot \theta_{(i,j,k)}. \tag{6.52}$$

Here, $\mathit{\Pi}$ is again the set of all camera parameters (intrinsic and distortion parameters), $\mathit{\Xi}$ is the set of all camera poses ($\mathit{\Xi} := \{\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_M\}$). In comparison to Section 6.3.2 the model parameters are extended by the set P. This set contains all marker point coordinates ($P := \{\boldsymbol{x}_W^{(1)}, \cdots, \boldsymbol{x}_W^{(N)}\}$). Each calibration point in a micro image results in a 2D residual vector $\boldsymbol{r}_{(i,j,k)}$. The function $\theta_{(i,j,k)}$ is a masking function which is 1 if the i-th object point is visible in the k-th micro image of the j-th view, or else is zero. Again, $\theta_{(i,j,k)}$ can be extended by any robust loss function. In contrast to the previous approach, here, the object point coordinates $\boldsymbol{x}_W \in P$ are also defined as model parameters which are refined in the optimization.

The residual vector for the calibration point with measured coordinates $\boldsymbol{x}_{MLd}$, corresponding to the i-th object point which is seen in the k-th micro image of the j-th camera view, is defined as follows:

$$\boldsymbol{r}_{(i,j,k)} = \pi_{ML}\left(\boldsymbol{G}(\boldsymbol{\xi}_j)\boldsymbol{x}_W^{(i)}, \boldsymbol{c}_{ML}^{(k)}, \mathit{\Pi}\right) - \boldsymbol{x}_{MLd}. \tag{6.53}$$

The function $\pi_{ML}(\cdot)$ defines the projection from camera to micro image coordinates of the k-th micro image. Hence, $\pi_{ML}(\cdot)$ defines either of the two camera models (Section 6.2.2 or Section 6.2.3), both of which rely on the same number of parameters. In eq. (6.52) the sum over i is the sum over all object points, the sum over j is the sum over all camera poses and the sum over k is the sum over all micro images. The vector $\boldsymbol{\xi}_j \in \mathfrak{se}(3)$ defines the rigid body transformation $\boldsymbol{G}(\boldsymbol{\xi}_j) \in \mathrm{SE}(3)$ from world coordinates to camera coordinates of the j-th view.

The optimal parameters ($\hat{\mathit{\Pi}}$, $\hat{\mathit{\Xi}}$, $\hat{P}$) are the ones which minimize the cost function $E(\mathit{\Pi}, \mathit{\Xi}, P)$:

$$\{\hat{\mathit{\Pi}}, \hat{\mathit{\Xi}}, \hat{P}\} = \arg\min E(\mathit{\Pi}, \mathit{\Xi}, P). \tag{6.54}$$

Similarly as for the previous approach, this highly nonlinear optimization problem is solved using the Levenberg-Marquardt algorithm. The initialization of $\mathit{\Pi}$, $\mathit{\Xi}$ and P must guarantee that the optimization starts in the convex region around the optimum solution $E(\hat{\mathit{\Pi}}, \hat{\mathit{\Xi}}, \hat{P})$.

The scaling of the calibration object is obtained based on known distances between certain object points. Those distances are used as additional constraints in the optimization problem.

In Chapter 9 the various methods and models, which were developed to calibrate a focused plenoptic camera, and which were presented in this chapter, will be evaluated and assessed.

7 Full-Resolution Direct Plenoptic Odometry

Our Direct Plenoptic Odometry (DPO) algorithm performs direct image alignment and semi-dense mapping directly on the micro images recorded by a plenoptic camera. It makes use of the geometric camera interpretation defined in Chapter 4. The geometric camera interpretation relies on known intrinsic camera parameters as well as distortion corrected images. Therefore, prior to running DPO, the plenoptic camera must be calibrated as described in Chapter 6.

7.1 Overview

Figure 7.1 shows the complete workflow of DPO. Furthermore, Algorithm 1 in Appendix B.1 shows a more detailed pseudo code of the approach. A brief overview of the algorithm is given here and it will be focused on the individual steps of the workflow in later sections (Section 7.2 to 7.6). A new recorded light field frame is tracked based on a keyframe. Keyframes are selected light field frames in respect to which tracking and mapping of all recorded frames takes place. In addition to the light field (raw image), two depth maps (a micro image depth map and a virtual image depth map) are established for each keyframe and a totally focused intensity image is synthesized. After a successful tracking of a light field frame, the current keyframe is updated, i.e. the depth maps are refined based on the tracked frame and the totally focused image is recalculated.

While the DPO algorithm is in progress, the pose of a newly recorded frame differs more and more from the one of the initially established keyframe. Based on a score, for each new frame we check whether the old keyframe is still valid or whether the new frame has to be set as a new keyframe. For a new keyframe, in-frame depth estimation is performed based on its recorded light field. The in-frame depth map is merged with the one of last keyframe which is propagated into the new perspective. Hence, only the depth maps of the current keyframe will be kept. To adjust scale drifts, the absolute scale of the last keyframe is estimated before it is deleted. Using the new scale estimate, the absolute scale along the entire trajectory is optimized and the poses of all past keyframes are updated.

7.2 Depth Map Representation

For a plenoptic camera, the observed inverse virtual depth v^{-1} can be considered to result from a Gaussian process, as was shown in Chapter 5. Therefore, due to eq. (4.21), the observed inverse effective depth $\lambda^{-1} = z_C'^{-1}$, which was introduced in Section 4.1, has to result from a Gaussian process, too. This holds true, at least as long as the micro images, which are used to obtain the observation, point in the same direction. Similar to the proposed probabilistic depth estimation approach (Chapter 5), and similar to Engel et al. [2013], a probabilistic depth hypothesis is defined

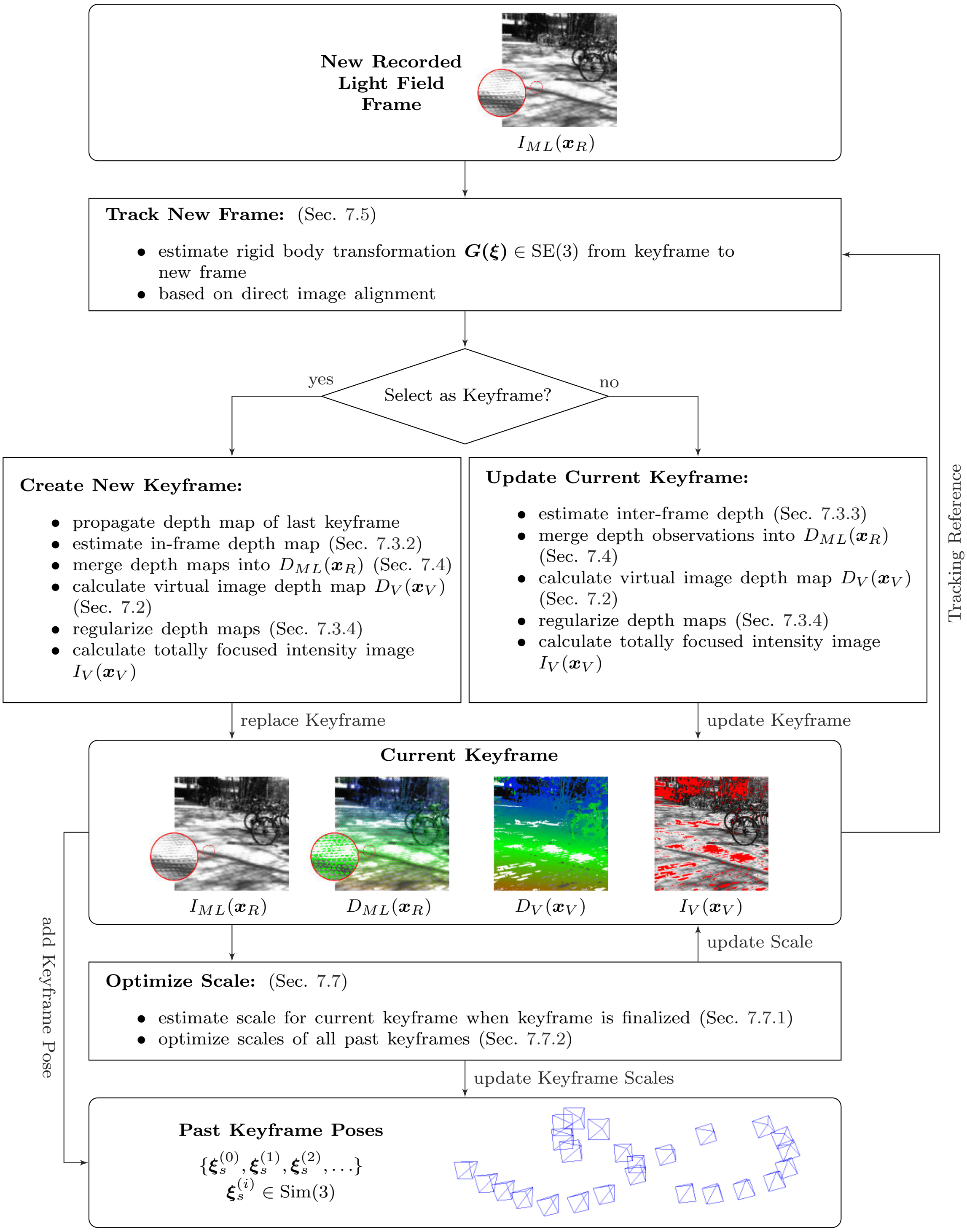

New Recorded Light Field Frame
$I_{ML}(\boldsymbol{x}_R)$
Track New Frame: (Sec. 7.5)
estimate rigid body transformation $\boldsymbol{G}(\boldsymbol{\xi}) \in \mathrm{SE}(3)$ from keyframe to new frame
based on direct image alignment
Select as Keyframe?
yes
no
Create New Keyframe:
propagate depth map of last keyframe
estimate in-frame depth map (Sec. 7.3.2)
merge depth maps into $D_{ML}(\boldsymbol{x}_R)$ (Sec. 7.4)
calculate virtual image depth map $D_V(\boldsymbol{x}_V)$ (Sec. 7.2)
regularize depth maps (Sec. 7.3.4)
calculate totally focused intensity image $I_V(\boldsymbol{x}_V)$
Update Current Keyframe:
estimate inter-frame depth (Sec. 7.3.3)
merge depth observations into $D_{ML}(\boldsymbol{x}_R)$ (Sec. 7.4)
calculate virtual image depth map $D_V(\boldsymbol{x}_V)$ (Sec. 7.2)
regularize depth maps (Sec. 7.3.4)
calculate totally focused intensity image $I_V(\boldsymbol{x}_V)$
Tracking Reference
replace Keyframe
update Keyframe
Current Keyframe
$I_{ML}(\boldsymbol{x}_R)$
$D_{ML}(\boldsymbol{x}_R)$
$D_V(\boldsymbol{x}_V)$
$I_V(\boldsymbol{x}_V)$
add Keyframe Pose
update Scale
Optimize Scale: (Sec. 7.7)
estimate scale for current keyframe when keyframe is finalized (Sec. 7.7.1)
optimize scales of all past keyframes (Sec. 7.7.2)
update Keyframe Scales
Past Keyframe Poses
$\{\boldsymbol{\xi}_s^{(0)}, \boldsymbol{\xi}_s^{(1)}, \boldsymbol{\xi}_s^{(2)}, \ldots\}$
$\boldsymbol{\xi}_s^{(i)} \in \mathrm{Sim}(3)$

Figure 7.1: Workflow of the Direct Plenoptic Odometry (DPO) algorithm.

for each image point with an absolute gradient above a certain threshold. For this the inverse effective depth is modeled as a Gaussian process:

$$\mathcal{N}\left(d, \sigma_d^2\right). \tag{7.1}$$

Here, $d = z_C'^{-1}$ defines the mean, while σ_d^2 is the corresponding variance of the inverse effective depth.

7.2.1 Micro Image Depth Map and Virtual Image Depth Map

Using the probabilistic depth model (eq. (7.1)) two different depth maps ($D_{ML}(\boldsymbol{x}_R)$ and $D_V(\boldsymbol{x}_V)$) are defined for each keyframe (Figures 7.2b and 7.2c).

$D_{ML}(\boldsymbol{x}_R)$ is defined with respect to raw image coordinates $\boldsymbol{x}_R$. Here, a point in the i-th micro image, with micro image coordinates $\boldsymbol{x}_{ML}$, has the raw image coordinates $\boldsymbol{x}_R = \boldsymbol{x}_{ML} + \boldsymbol{c}_{ML}^{(i)}$. $D_V(\boldsymbol{x}_V)$ as well as a totally focused intensity image $I_V(\boldsymbol{x}_V)$ (Figure 7.2d) are defined on virtual image coordinates $\boldsymbol{x}_V$ (see Section 5.5), where micro image points observing the same object point are merged (see Section 7.4).

These two distinct representations are considered because in the raw image one object point is projected to multiple image points, while there is a one-to-one mapping from object space to the virtual image. In the DPO algorithm, depth estimation is performed on the micro image representation ($D_{ML}(\boldsymbol{x}_R)$ and $I_{ML}(\boldsymbol{x}_R)$), while the virtual image representation ($D_V(\boldsymbol{x}_V)$ and $I_V(\boldsymbol{x}_V)$) of a keyframe is only used for tracking new frames. Since DPO makes use of the focused high resolution micro images (raw image) instead of, for instance, low resolution sub-aperture images, it is called a full-resolution algorithm.

7.3 Estimating Depth Hypotheses

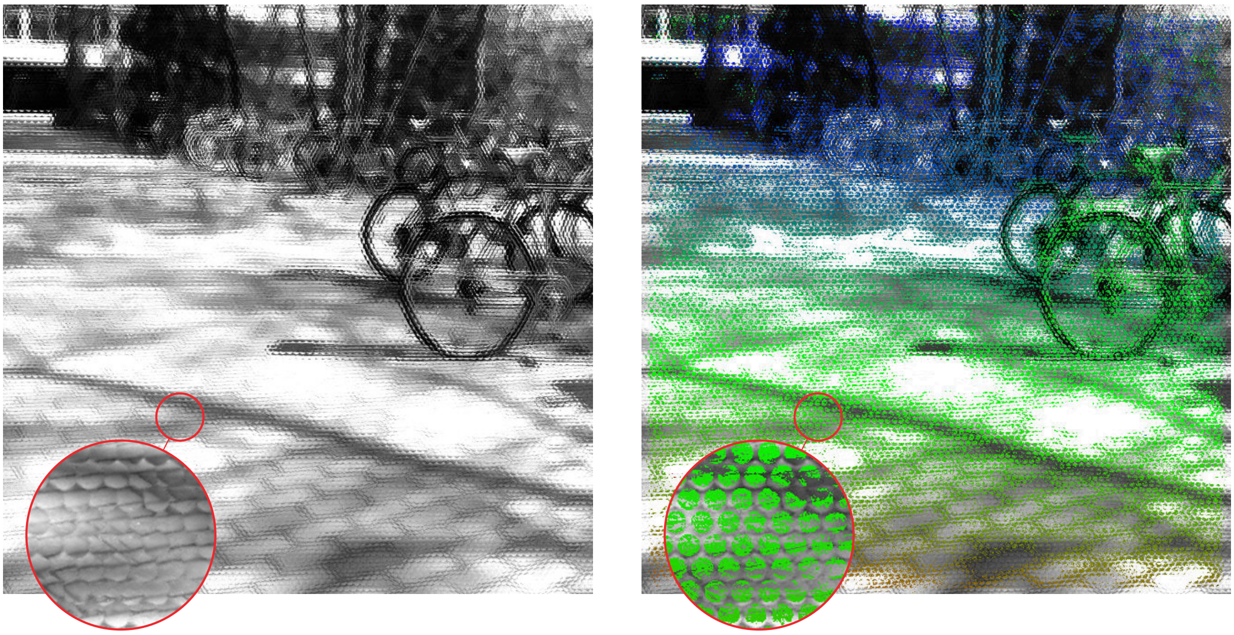
For each keyframe, in-frame depth estimation is performed based on stereo matching in its own micro images as well as inter-frame depth estimation based on micro images of subsequent frames using the epipolar gemometry defined in Section 4.2. For both in-frame and inter-frame depth estimation, stereo matches are found by optimizing the sum of squared intensity differences (SSID) over a one-dimensional pixel patch along the epipolar line, similar to Section 5.3. Here, prior estimates are used to narrow the search range to $d \pm 2\sigma_d$.

7.3.1 Observation Uncertainty

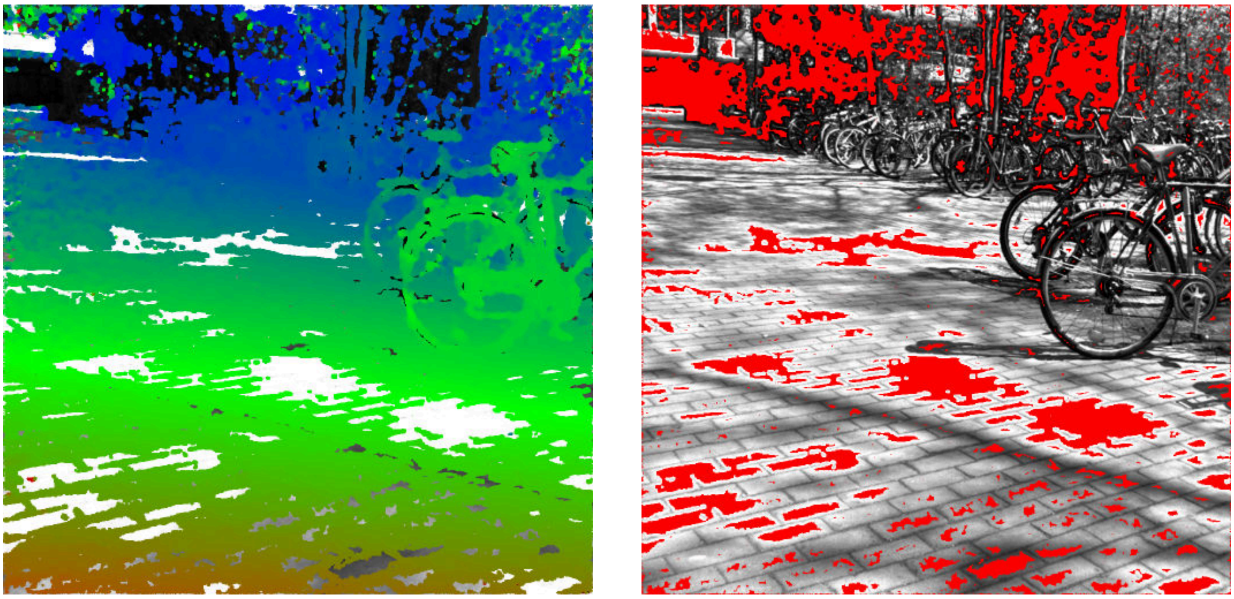
Similar to Engel et al. [2013], the observation uncertainty, which results in the variance σ_d^2 of the inverse effective depth d, is derived based on uncertainty propagation. In addition to a geometric and a photometric disparity error, a focus disparity error, which regards unfocused micro images, similar to Chapter 5, is defined.

Geometric Disparity Error

Because of inaccuracies in the camera model and uncertainties in the frame pose, one can expect a misalignment of the epipolar line. Due to the very short search range, it is considered, similar to Engel et al. [2013], that only the absolute epipolar line position $\boldsymbol{l}_0$ to be affected by isotropic Gaussian noise ϵ_l and any rotational error is neglected. Since here the search range is always limited by the micro lens dimension, the assumption of a short search range holds even better than it does for the monocular case. Thus, the variance $\sigma_{\mu(\xi,\pi)}^2$ of the geometric disparity error



(a) raw intensity image (b) raw image depth map



(c) virtual image depth map (d) totally focused intensity image

Figure 7.2: Keyframe depth maps and intensity images. (a) Micro images $I_{ML}(\boldsymbol{x}_R)$ (raw image recorded by the plenoptic camera). (b) Micro image depth map $D_{ML}(\boldsymbol{x}_R)$. (c) Virtual image depth map $D_V(\boldsymbol{x}_V)$. (d) Totally focused intensity image $I_V(\boldsymbol{x}_V)$. For the pixels marked in red no depth value, and therefore no intensity, was calculated.

can be defined as follows (Engel et al. [2013]):

$$\sigma^2_{\mu(\xi,\pi)} = \frac{\sigma_l^2}{\langle \boldsymbol{g}, \boldsymbol{l} \rangle^2}. \tag{7.2}$$

Here, $\boldsymbol{g}$ is the normalized image gradient, $\boldsymbol{l}$ is the normalized epipolar line direction, and is σ_l^2 the variance of ϵ_l.

Photometric Disparity Error

The photometric disparity error describes the effect of sensor noise ϵ_n on the estimated disparity. It results in an error on the estimated disparity with variance $\sigma^2_{\mu(I)}$, as derived in Section 5.3.1:

$$\sigma^2_{\mu(I)} = \frac{2 \cdot \sigma_n^2}{g_I^2}. \tag{7.3}$$

Here σ_n^2 is the variance of ϵ_n and g_I the intensity gradient along the epipolar line.

Focus Disparity Error

The focus disparity error models the effect of differently focused micro images used for stereo matching. The variance $\sigma^2_{\mu(v,k,j)}$ of the focus disparity error is defined as follows, where eq. (5.21) is restated:

$$\sigma^2_{\mu(v,k,j)} = \sigma_x^2 \cdot \left(1 - \frac{\sigma_j}{\sigma_k}\right)^2. \tag{7.4}$$

Here, σ_x^2 defines the distribution of the position, in relation to an edge, on the epipolar line. Furthermore, σ_j and σ_k define the blur radii in the respective micro images used for disparity estimation. A detailed derivation of the focus disparity error is given in Section 5.3.1.

Though, as described in Chapter 5, the two micro images used for disparity estimation are rectified in relation to each other for the case of in-frame depth estimation, this is not the case for inter-frame depth estimation. Hence, the definition of the position x given in eq. (5.16) must be modified as follows:

$$x = \mu_k - \mu_{0k} = \mu_j \cdot \gamma - \mu_{0j}. \tag{7.5}$$

The parameter $\gamma = \frac{z_C'^{(j)}}{z_C'^{(k)}}$ defines the scaling factor between the two micro images of interest.

When considering all three error sources to be independent random variables, the overall observation uncertainty is received as follows:

$$\sigma_d^2 = \alpha^2 \cdot \left(\sigma^2_{\mu(\xi,\pi)} + \sigma^2_{\mu(I)} + \sigma^2_{\mu(v,k,j)}\right). \tag{7.6}$$

The parameter α defines the derivative of d with respect to the disparity μ.

7.3.2 Estimating In-Frame Depth

In-frame depth estimation is performed similarly to the process that has been described previously in Chapter 5. The only difference is that here directly the inverse effective depth $z_C'^{-1}$ is calculated instead of the inverse virtual depth v^{-1}. Nevertheless, since there is a linear connection, the overall procedure remains the same.

7.3.3 Estimating Inter-Frame Depth

While for in-frame depth estimation the stereo baseline is limited by the camera dimensions, we are able to improve the depth accuracy based on inter-frame depth observations. Inter-frame depth estimation is performed based on the current keyframe and the newly tracked frames. Therefore, the relative orientation between the frames $\boldsymbol{\xi} \in \mathfrak{se}(3)$ is considered to be known.

For each micro image point in the keyframe, stereo observations are obtained from all possible micro images in the new frame[1].

---

[1] Possible micro images are those which very likely see the respective object points.

Defining Epipolar Lines

One is able to define the epipolar geometry between the micro images of two different frames as given in Section 4.2. Due to the linear relation between projected micro image coordinates $\boldsymbol{x}_p$ and real micro image coordinates onto the sensor $\boldsymbol{x}_{ML}$ (or $\boldsymbol{x}_R$), the epipolar lines defined in the projected micro images can be directly mapped onto the sensor. Therefore, stereo matching for inter-frame depth estimation can be performed directly on the recorded raw images, in a way similar to in-frame depth estimation.

7.3.4 Regularizing Depth Maps

Each time the depth maps are updated, a regularization step on $D_{ML}(\boldsymbol{x}_R)$ and $D_V(\boldsymbol{x}_V)$ is performed. Outliers are removed and estimates are smoothed based on probabilistic metrics, as stated in Section 5.6. In the virtual image, multiple micro image points are merged to a single image point. Furthermore, the neighborhood of a point $\boldsymbol{x}_V$ is not bounded by the micro image borders as it is for a raw image point $\boldsymbol{x}_R$. Therefore, it is far easier to detect outliers in $D_V(\boldsymbol{x}_V)$ and to mark the corresponding points in $D_{ML}(\boldsymbol{x}_R)$ as outliers, too. The depth hypotheses in $D_{ML}(\boldsymbol{x}_R)$ themselves are not updated based on $D_V(\boldsymbol{x}_V)$ in order to avoid loops in the filtering process. After updating $D_V(\boldsymbol{x}_R)$, the totally focused intensity image $I_V(\boldsymbol{x}_R)$ is recalculated, too.

7.4 Merging Depth Hypotheses

Probabilistic depth values are updated in a Kalman-like fashion, in a similar as done by Engel et al. [2013]. However, in contrast to the update step in Section 5.4 and to Engel et al. [2013], here the variance of the merged depth hypothesis is limited to the variance of the best observation. This has the effect that multiple depth observations of the same point are not considered to be uncorrelated, which in the limit case would lead to zero variance. Thus, the following merging routine is defined:

$$\mathcal{N}\left(\frac{\sum_{i\in O} d_i \cdot \left(\sigma_{di}^2\right)^{-1}}{\sum_{i\in O}\left(\sigma_{di}^2\right)^{-1}}, \min\left(\sigma_{di}^2\right)\right), \tag{7.7}$$

where O is the set of depth observations. This algorithm prevents points in the background which are observed in numerous frames from receiving low variances.

7.5 Tracking

Newly recorded frames are tracked based on direct image alignment. Here, the current keyframe ($D_V(\boldsymbol{x}_V)$ and $I_V(\boldsymbol{x}_V)$) is used as tracking reference. In the following, the relative pose, which has to be estimated is denoted by $\boldsymbol{\xi}_{kj}$, where k defines the keyframe index and i the index of the current frame.

In the publication Zeller et al. [2017] a tracking approach is described which is based on an initial tracking stage using a synthesized totally focused image of the new frame. To synthesize this image, the depth map of the new frame has to be known. This depth map can either be calculated or, as described by Zeller et al. [2017], an approximation of it can be obtained by projecting the depth map from the current keyframe into the new frame. For this projection an initial estimate of the relative pose $\boldsymbol{\xi}_{kj}$ must be used. However, this procedure is quite time consuming and furthermore, the projected depth map might be incomplete and error-prone.

A new tracking approach is presented which, again, works on pyramid levels. However, this new approach does not rely on a depth map of the new frame. Furthermore, strategies are presented to make the tracking more robust.

7.5.1 Tracking on Pyramid Levels in the Micro Images

In direct tracking approaches there exists no fixed point correspondences and therefore the region of convergence around the optimal solution is quite limited. Hence, a good initialization of the model parameters which have to be optimized, in this case $\boldsymbol{\xi}_{kj}$, is needed. Kerl et al. [2013b], for instance, showed that the region of convergence can be increased by performing tracking in a coarse-to-fine approach.

In one of our previous publications (Zeller et al. [2017]), initial tracking was performed on totally focused intensity images. Here it was shown that a course-to-fine approach can, in fact, also be performed directly on the micro images of the new frame.

To do this, pyramid levels of the complete recorded raw image are built by simple pixel binning. As long as the size of a pixel on a certain pyramid level is smaller than a micro lens, the image of reduced size still represents a light field image. At coarse levels, where the pixel size exceeds the size of a micro lens, the image turns into a standard central perspective image. In a first assumption this can be viewed as integrating along the u and v domain in the two plane parametrization (2PP) of the light field[2].

For all pyramid levels, the following energy function is defined, which then is optimized with respect to $\boldsymbol{\xi}_{kj}$:

$$E(\boldsymbol{\xi}_{kj}) = \sum_i \sum_l \left\| \left(\frac{r_{ML}^{(i,l)}}{\sigma_r^{(i,l)}} \right)^2 \right\|_\delta , \tag{7.8}$$

$$r_{ML}^{(i,l)} := I_{Vk}\left(\boldsymbol{x}_V^{(i)}\right) - I_{MLj}\left(\pi_{ML}\left(\boldsymbol{G}(\boldsymbol{\xi}_{kj})\pi_V^{-1}(\boldsymbol{x}_V^{(i)}), \boldsymbol{c}_{ML}^{(l)}\right)\right). \tag{7.9}$$

Here $\pi_{ML}(\cdot,\cdot)$ defines the projection from camera coordinates $\boldsymbol{x}_C$ to micro image coordinates $\boldsymbol{x}_R$, through a certain micro lens $\boldsymbol{c}_{ML}$. Furthermore, $\pi_V^{-1}(\cdot)$ defines the projection from virtual image coordinates $\boldsymbol{x}_V$ to camera coordinates $\boldsymbol{x}_C$. The operation $\|\cdot\|_\delta$ denotes the robust Huber norm (Huber [1964]).

Initial Tracking

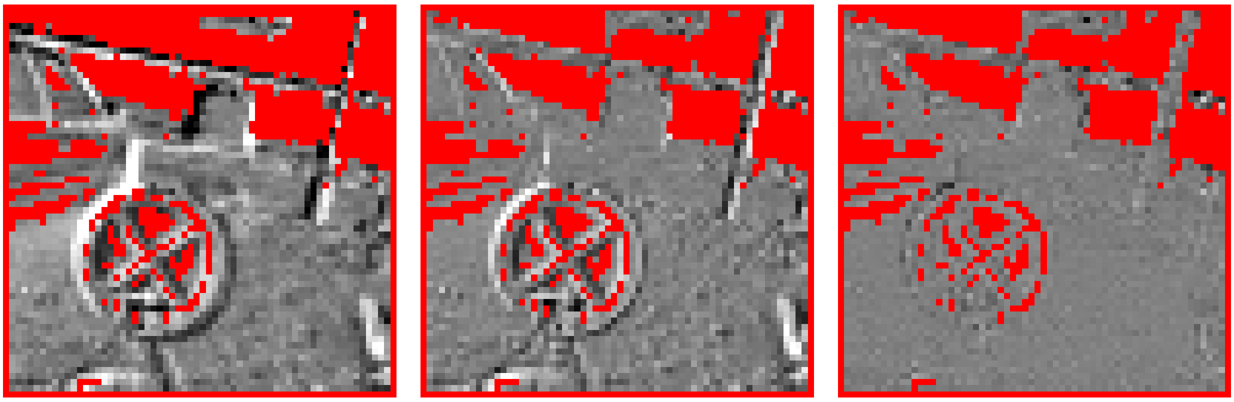
In the initial stage, each point in the tracking reference (frame k) $\boldsymbol{x}_V^{(i)} \mapsto \boldsymbol{x}_C^{(i)}$ is projected to only a single micro image of the new frame (frame j). Hence, in eq. (7.8) the sum over l is dropped for initialization.

For the sake of simplicity, the same energy function is also kept for the central perspective images at coarse pyramid levels. Here, a micro lens consists only of one or even of only a fraction of a pixel and therefore the projection becomes equivalent to a central perspective projection.

Starting from the coarsest pyramid level, the function $E(\boldsymbol{\xi}_{kj})$ is optimized with respect to $\boldsymbol{\xi}_{kj}$ for each pyramid level. This is done by the Levenberg-Marquardt as introduced in Section 2.5.3.

---

[2] This is true only on first assumption, since a 2PP of the light field can not be directly extracted from the raw image of a focused plenoptic camera.



(a) first iteration (b) 4th iteration (c) 9th iteration

Figure 7.3: Tracking residual after various numbers of iterations. The figure shows residuals in virtual image coordinates of the tracking reference. The grey value represents the value of the tracking residual. Black: negative residuals with high absolute values. White: positive residuals with high absolute values. Red: regions without valid depth information and therefore without tracking residuals.

Final Tracking

On the final (finest) pyramid level, a single point in the tracking reference is projected to all micro images in the new frame which see this point. This is modeled by the sum over l in eq. (7.8). By projecting the reference points to multiple micro images in the new frame, one is able to implicitly incorporate the in-frame disparities of the new frame into the optimization problem. In this way, the accumulated scale drifts, as they generally occur for monocular tracking, are kept small.

Figure 7.3, by way of example, shows the tracking residual for different iterations of the Levenberg-Marquardt optimization.

7.5.2 Variance of Tracking Residual

The variance σ_r^2 of the photometric residual r_{ML} is obtained by uncertainty propagation. Therefore, σ_r^2 consists of a photometric component, which results from noise on the intensities and a geometric component, which results from noise on the depth estimate as follows:

$$\sigma_r^2 := \sigma_n^2 \left(\frac{1}{N_k} + 1 \right) + \left| \frac{\partial r(\boldsymbol{x}_V, \boldsymbol{\xi}_{kj})}{\partial d(\boldsymbol{x}_V)} \right|^2 \sigma_d^2(\boldsymbol{x}_V). \tag{7.10}$$

The noise on the intensities in the different micro images is considered to be uncorrelated. In the totally focused image an intensity value is calculated as the average value over multiple micro image points. Therefore, the variance of an intensity value in the totally focused image results from the variance of the sensor noise σ_n^2 divided by the number of micro image points used to calculate the intensity value:

$$\left(\sigma_I^{(k)} \right)^2 = \frac{\sigma_n^2}{N_k}. \tag{7.11}$$

Here, N_k is the number of micro image points used for calculation. Since micro image points are measured directly on the sensor, their noise is equivalent to the sensor noise $\sigma_I^{(j)} = \sigma_n$.

The geometric term is received based on uncertainty propagation from the variance σ_d^2 of the corresponding depth hypothesis.

7.5.3 Compensation of Lighting Changes

To compensate for changing lighting conditions in the recorded scene and for different exposure times between the current keyframe and the frame to be tracked, the residual term defined in eq. (7.9) is extended. Therefore, the residual term $r_{ML}^{(i,l)}$ is modified as follows:

$$r_{ML}^{(i,l)} := I_{Vk}\left(\boldsymbol{x}_V^{(i)}\right) \cdot a + b - I_{MLj}\left(\pi_{ML}\left(\boldsymbol{G}(\boldsymbol{\xi}_{kj})\pi_V^{-1}(\boldsymbol{x}_V^{(i)}), \boldsymbol{c}_{ML}^{(l)}\right)\right). \tag{7.12}$$

Here, a and b are scalars which compensate for lighting changes by an affine transformation of the intensities in the totally focused image of the tracking reference. The coefficient a and b also must be estimated in the optimization process.

For most cases, $a = 1$ and $b = 0$ is a good initialization for the lighting compensation. However, especially for strong lighting changes between the reference and the new frame, this might be a bad initialization and, in the worst case, leads to a tracking failure.

Hence, an initialization is formulated based on first- and second-order statistics calculated from the intensity images I_{Vk} and I_{MLj}. Obviously, the initial value of b results as follows:

$$b_{\text{init}} = \overline{I}_{MLj} - \overline{I}_{Vk}. \tag{7.13}$$

In eq. (7.13) $\overline{I}_{MLj}$ and $\overline{I}_{Vk}$ are the average intensity values over the complete images respectively. The initialization of a can be defined as follows:

$$a_{\text{init}} = \frac{\sigma_{I_{MLj}}}{\sigma_{I_{Vk}}}. \tag{7.14}$$

Here, $\sigma_{I_{MLj}}$ and $\sigma_{I_{Vk}}$ are the empirical standard deviations over the image intensities respectively.

7.5.4 Motion Prior for Tracking Robustness

For a focused plenoptic camera, the depth accuracy is more or less proportional to the squared focal length f_L^2. Therefore, one chooses a large focal length and simultaneously tries to keep a sufficiently wide FOV. However, the chosen FOV will be much narrower than the one of a monocular camera used in VO. Because of the narrow FOV, the region of convergence in the tracking approach will shrink.

It is taken advantage of the fact that sequences in general are recored with high frame rates compared to the acceleration of the camera. Therefore, the motion vectors of subsequent frames will be highly correlated. A motion prior is formulated based on a linear motion prediction to constrain the optimization. In this way, one does not increase the region of convergence but shifts it to an area where the optimal solution is more likely to be located.

A prediction $\widetilde{\boldsymbol{\xi}}_{kj} \in \mathfrak{se}(3)$ of the relative pose, between the tracking reference (k-th frame) and the new frame (j-th frame) which is tracked, is obtained as follows:

$$\widetilde{\boldsymbol{\xi}}_{kj} = \dot{\boldsymbol{\xi}}_{j-1} \circ \boldsymbol{\xi}_{k(j-1)}. \tag{7.15}$$

The vector $\boldsymbol{\xi}_{k(j-1)} \in \mathfrak{se}(3)$ is the relative pose estimate of the last frame and $\dot{\boldsymbol{\xi}}_{j-1}$ the respective motion vector. Using the new pose prediction $\widetilde{\boldsymbol{\xi}}_{kj}$, the energy function $E(\boldsymbol{\xi}_{kj})$ is extended as follows:

$$E(\boldsymbol{\xi}_{kj}) = \sum_i \sum_l \left\| \left(\frac{r_{ML}^{(i,l)}}{\sigma_r^{(i,l)}} \right)^2 \right\|_\delta + \tau(\delta\boldsymbol{\xi})^T \delta\boldsymbol{\xi}, \qquad \text{with}$$

$$\delta\boldsymbol{\xi} = \log_{\mathrm{SE}(3)}\left(\exp_{\mathfrak{se}(3)}(\boldsymbol{\xi}_{kj}) \cdot \left(\exp_{\mathfrak{se}(3)}(\widetilde{\boldsymbol{\xi}}_{kj}) \right)^{-1} \right) = \boldsymbol{\xi}_{kj} \circ \left(-\widetilde{\boldsymbol{\xi}}_{kj} \right). \tag{7.16}$$

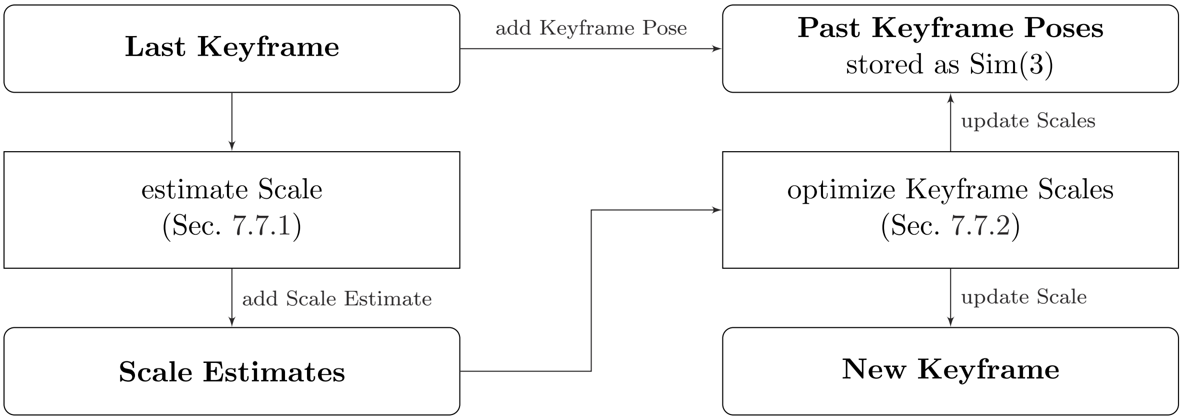


Figure 7.4: Workflow of the scale optimization framework.

In eq. (7.16) τ weights the motion prior with respect to the photometric term. On coarse pyramid levels the initial frame pose is very uncertain and can be far from the correct frame pose. Therefore a hight weight τ is chosen. This weight is decreased as the optimization moves down in the pyramid. On the finest pyramid level, the weight is set to $\tau = 0$ such that an error in the predicted pose does not influence the final estimate.

7.6 Selecting Keyframes

A new keyframe must be selected when the perspective of a new frame differs too much from the one of the current keyframe. Therefore, a score is defined based on the translation between the two frames and the overlap of the two views.

When a frame is selected to be the new keyframe, in-frame depth estimation is performed in the new frame. Afterwards, the micro image depth map of the last keyframe is propagated to the new one and the depth hypotheses are merged. Projected depths which are divergent from the in-frame depth are rejected and not merged.

7.7 Optimizing Global Scale

From the micro images created by the MLA, one is able to estimate depth maps from a single image of the camera. Due to the small stereo baseline between the micro images in a single frame, the estimates have a high uncertainty, especially for large object distances. This high uncertainty in the in-frame depth estimates will negatively affect the absolute scale of the trajectory and the 3D reconstruction estimated by the VO algorithm.

Here, a framework is proposed which estimates and optimizes the absolute scale. For each keyframe which is finalized, a scale estimate is obtained (Section 7.7.1). These scale estimates are fed to a probabilistic filter which optimizes the global scale of the complete trajectory (Section 7.7.2). All past keyframe poses are stored in a list, to be able to update the scale of previous keyframes based on the information gained from new scale estimates. This scale optimization framework is visualized in Figure 7.4.

7.7.1 Keyframe based Scale Estimation

Scale estimation can be performed in a way similar to how tracking is performed. It can be viewed as tracking a light field frame based on its own virtual image depth map $D(\boldsymbol{x}_V)$. However, instead of optimizing all pose parameters, the logarithmized scale (log-scale) ρ is optimized. The parameter ρ is equivalent to the tangent space scale parameter for the Lie group of similarity transformations Sim(3), which was introduced in Section 2.5.2.

The scale estimation works on the log-scale ρ to transform the scale $s = e^{\rho}$ into a Euclidean space. The scale operation actually forms a 1D Lie group with the tangent space ρ. Better yet, it forms a commutative Lie group which means that any concatenation of scale results in a linear operation on the tangent space, without any approximation.

Similar to the tracking approach described in Section 7.5, an energy function $E(\rho)$ is defined as follows:

$$E(\rho) = \sum_{i} \sum_{l \neq 0} \left\| \left(\frac{r_{ML}^{(i,l)}}{\sigma_r^{(i,l)}} \right)^2 \right\|_{\delta}, \tag{7.17}$$

$$r_{ML}^{(i,l)} := I_{MLk}\left(\pi_{ML}\left(\pi_V^{-1}(\boldsymbol{x}_V^{(i)}, \rho), \boldsymbol{c}_{ML}^{(0)}\right)\right) - I_{MLk}\left(\pi_{ML}\left(\pi_V^{-1}(\boldsymbol{x}_V^{(i)}, \rho), \boldsymbol{c}_{ML}^{(l)}\right)\right). \tag{7.18}$$

$$\left(\sigma_r^{(i,l)}\right)^2 := 2\sigma_n^2 + \left| \frac{\partial r_{ML}^{(i,l)}(\boldsymbol{x}_V^{(i)}, \rho)}{\partial \sigma_d(\boldsymbol{x}_V^{(i)})} \right|^2 \sigma_d^2(\boldsymbol{x}_V^{(i)}). \tag{7.19}$$

Instead of defining the photometric residual r_{ML} with respect to the intensities of the totally focused image, residuals are defined as the differences between the intensity of the centered micro image and those of all surrounding micro images which still see the same virtual image point $\boldsymbol{x}_V$. The residual is defined this way, since a wrong scale also slightly affects the intensity in the totally focused image, while the micro image coordinates and therefore the respective intensities are refined in each iteration of the optimization.

A central perspective projection, as it is performed in a pinhole camera, is completely scale invariant. This is not the case for the plenoptic camera, where the totally focused image is not formed on a plane. Changing the scale in the z_C domain by a scale factor $s = e^{\rho}$, while at the same time keeping the virtual image coordinates x_V and y_V unchanged, will result in the following scaled 3D camera coordinates $\boldsymbol{x}_{Cs} = [x_{Cs}, y_{Cs}, z_{Cs}]^T$:

$$z_{Cs} = e^{\rho} \cdot z_C, \qquad x_{Cs} = \frac{z_{Cs} - f_L}{z_C - f_L} \cdot x_C, \qquad y_{Cs} = \frac{z_{Cs} - f_L}{z_C - f_L} \cdot y_C. \tag{7.20}$$

One can see that for $z_C \gg f_L$ or for $|\rho| \ll 1$, eq. (7.20) simplifies to the scale invariant case:

$$\boldsymbol{x}_{Cs} \approx e^{\rho} \cdot \boldsymbol{x}_C \qquad \text{for } z_C \gg f_L \text{ or } |\rho| \ll 1. \tag{7.21}$$

In eq. (7.18), the term $\pi_V^{-1}(\boldsymbol{x}_V^{(i)}, \rho)$ defines a warping function which firstly performs the inverse projection from the virtual image to the object space and then does the scaling as defined in eq. (7.20).

In conjunction to the log-scale estimate ρ, its variance σ_ρ^2 is calculated. This variance is defined as follows:

$$\sigma_\rho^2 = \frac{N}{\sum_{i=0}^{N-1} \sigma_{\rho i}^{-2}} \qquad \text{with} \qquad \sigma_{\rho i}^2 = \left| \frac{\partial \rho}{\partial d} \right|^2 \cdot \sigma_{di}^2. \tag{7.22}$$

Here, σ_{di}^2 is the inverse virtual depth variance of a certain virtual image point $\boldsymbol{x}_V^{(i)}$. Far points do not contribute to a reliable scale estimate, since the ration between the stereo baseline Δc_{ML} and the effective object distance z'_C of those points becomes negligibly small. Hence, the N points used to define the scale variance are only the closest N points or, in other words, the points with the largest inverse effective depth d.

7.7.2 Scale Optimization

In the DPO algorithm the depth map of a new keyframe is initialized by the depth map of the last keyframe. Thus, the scale of a new keyframe is also indirectly initialized by the scale of the last keyframe. Hence, one can expect that the scales of subsequent keyframes are highly correlated and scale drifts between them are marginal. Therefore the absolute log-scale of the complete trajectory is modeled as a random process over time with an autocorrelation function $L_{\rho\rho}(m)$ as follows:

$$L_{\rho\rho}(m) = c^{|m|}. \tag{7.23}$$

In eq. (7.23), the variable m is the discrete time index in keyframes and the parameter c ($0 \leq c \leq 1$) defines the correlation between subsequent keyframes. As a high correlation is considered, c will be close to one.

On the basis of the stochastic process with the autocorrelation function $L_{\rho\rho}(m)$, the maximum likelihood estimator for the expected value of the logarithmized scale (log-scale) $\hat{\rho}^{(l)}$ at a certain keyframe with time index l is formulated as follows:

$$\hat{\rho}^{(l)} = \left(\sum_{m=-N}^{N} \rho^{(m+l)} \cdot \frac{c^{|m|}}{\left(\sigma_\rho^{(m+l)}\right)^2} \right) \cdot \left(\sum_{m=-N}^{N} \frac{c^{|m|}}{\left(\sigma_\rho^{(m+l)}\right)^2} \right)^{-1}. \tag{7.24}$$

While each log-scale estimate $\rho^{(i)}$ ($i \in \{0, 1, \ldots, k\}$) is weighted by its inverse variance, estimates of keyframes which are farther from the keyframe of interest (index l) are down weighted by the respective power of c. In general, the influence length $N \to \infty$ can be chosen, although in the implementation N is truncated.

The proposed scale optimization realizes a non-causal finite impulse response (FIR) filter. Thus, the final estimate $\hat{\rho}^{(l)}$ is not available until the index k of the current keyframe $k \geq l + N$ holds. In Appendix B.2 a recursive formulation of the estimator defined in eq. (7.24) is presented. Instead of solving the complete sum for each new keyframe multiple times, the existing estimates $\hat{\rho}^{(l)}$ merely have to be updated.

7.8 Finding Micro Lenses of Interest

For several tasks in the DPO algorithm, one has to find the set of micro images in which an object point (or virtual image point respectively) is visible. In Section 5.7, it was previously described how these micro images are found for the case that they are looking straight ahead in the virtual image space. Corresponding micro images lie within a cone, spanned from the virtual image point, as shown in Figure 5.6.

However, this criteria changes for the case of squinting micro lenses, as is the case in a plenoptic camera. In this case, the cone is distorted. Similar to Section 5.7, a radius $R = 0.5 D_M \cdot v$ can be defined, which gives a circular area of micro lenses which see the respective virtual image point $\boldsymbol{x}_V$. Instead of having a micro lens center $\boldsymbol{c}_{ML}$ within the radius R around the orthogonal projection of the virtual image point $\boldsymbol{x}_V$ onto the MLA, the point has to be projected along the central ray through the main lens. Hence, the following condition has to be fulfilled:

$$\|(\boldsymbol{c}_I - \boldsymbol{c}_{ML}) \cdot v + \boldsymbol{c}_{ML} - \boldsymbol{x}_V\| \leq R. \tag{7.25}$$

Here, all coordinates $\boldsymbol{c}_I$, $\boldsymbol{c}_{ML}$, and $\boldsymbol{x}_V$ are considered to be 2D coordinates in the x-y-domain with the same origin.

For all micro lenses with center coordinates $\boldsymbol{c}_{ML}$ which fulfill the condition of eq. (7.25), it is guaranteed that the virtual image point $\boldsymbol{x}_V$ is projected onto the respective micro image. This condition is needed for inter-frame depth estimation, virtual intensity image synthesis, tracking and scale estimation, since in all of these tasks an object point $\boldsymbol{x}_C$, or respective virtual image point $\boldsymbol{x}_V$, has to be projected onto all micro images which actually see that point.

8 Experiments – Probabilistic Light Field based Depth Estimation

8.1 Data

Capturing ground truth depth data for images of real scenes is quite complex and requires the registration of the images with highly accurate 3D data, e.g. the registration of images and 3D data obtained from a laser scanner (e.g. Schöps et al. [2017]). This makes it challenging to asses the quality of depth information gained from images, be they stereo image pairs or, as in this case, plenoptic images. However, while for stereo data there exist several synthetic datasets for which a pixel-wise ground truth is available, there are no such datasets available for focused plenoptic cameras. There are some synthetic light field datasets (Wanner et al. [2013]; Honauer et al. [2017]) available. However, the algorithm proposed in Chapter 5, which is designed to perform depth estimation directly from the raw data of a plenoptic camera, is not fitted to used these synthetic 4D light field representations.

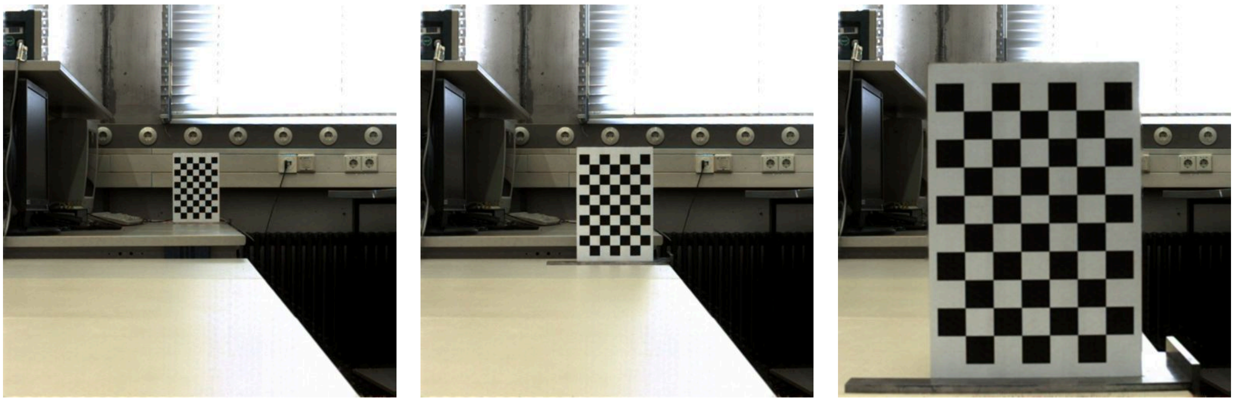
The proposed plenoptic camera based probabilistic depth estimation algorithm is evaluated on a quantitative basis by calculating statistics of the estimated virtual depth for a planar checkerboard pattern which is recorded in various object distances. In addition to this, the algorithm is evaluated qualitatively using available public datasets as well as own recordings for which no ground truth is available.

8.1.1 Checkerboard Recorded by the Plenoptic Camera Raytrix R5

To compare the proposed depth estimation algorithm quantitatively to already existing methods, different light field images were recorded by a Raytrix R5 camera with main lens focal length $f_L = 35\,\mathrm{mm}$, while placing a planar checkerboard in various distances to the camera. Figure 8.1 shows three images with the checkerboard placed in various object distances: $z_{C1} = 1.2\,\mathrm{m}$, $z_{C2} = 3.1\,\mathrm{m}$, and $z_{C3} = 5.1\,\mathrm{m}$. In the evaluation section the different algorithms are compared based on statistics calculated from the depth estimates across the checkerboard.

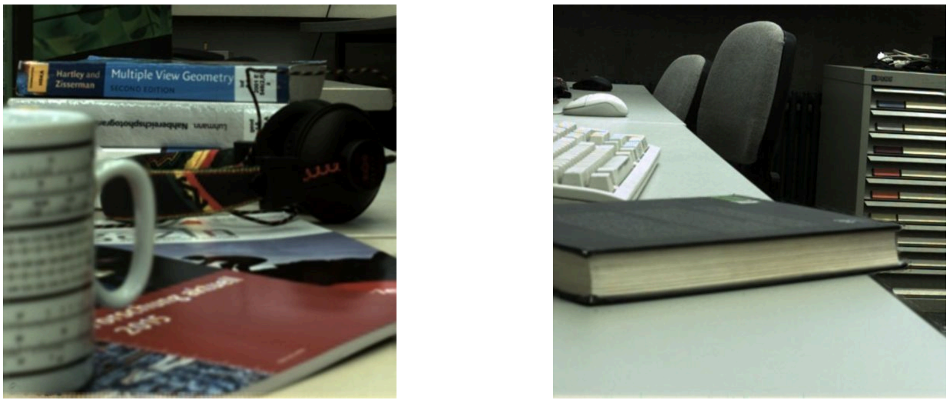
8.1.2 Plenoptic Images of Real Scenes

As mentioned before, most light field based depth estimation algorithms do not work on the raw data recorded by a plenoptic camera, but do work on some kind of 4D light field representation which can be resampled from this raw data. However, there is only a small number of public datasets available which were recorded by focused plenoptic cameras. These few public dataset as well as self-recorded images are used to evaluate the proposed algorithm and to compare it to the state-of-the-art on a qualitative basis. In the following, totally focused images is shown to visualize all recorded datasets.



(a) $z_{C1} = 1.2\,\mathrm{m}$ (b) $z_{C2} = 3.1\,\mathrm{m}$ (c) $z_{C3} = 5.1\,\mathrm{m}$

Figure 8.1: Checkerboard dataset recorded by a Raytrix R5 camera with main lens focal length $f_L = 35\,\mathrm{mm}$. A planar checkerboard pattern was placed in front of the plenoptic camera and recorded at various object distances. The quality of the estimated depth map can be evaluated by analyzing the scattering of the depth estimated across the checkerboard.



(a) Headphones (b) Desk

Figure 8.2: Own focused plenoptic camera dataset. Both images were recorded by a Raytrix R5 camera with main lens focal length $f_L = 35\,\mathrm{mm}$. The figures show the totally focused images synthesized from the recorded light fields.

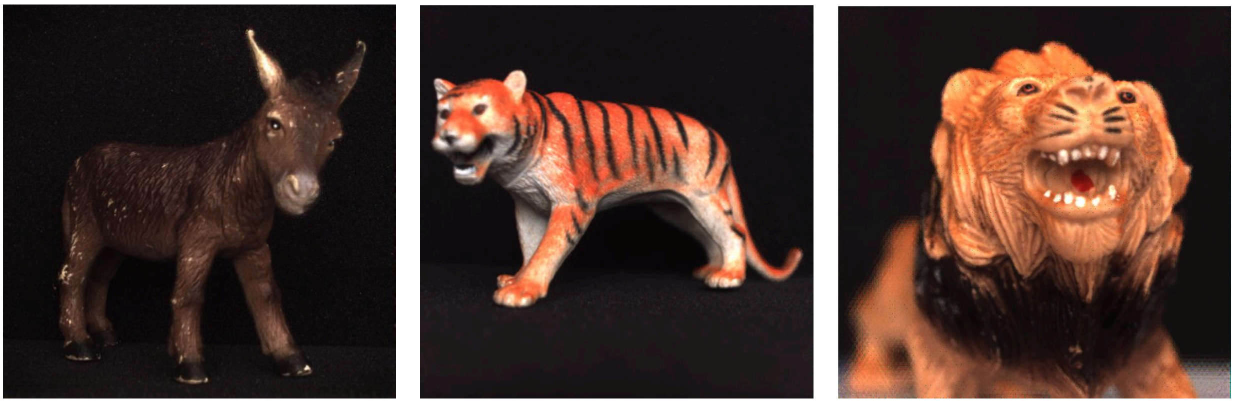
Figure 8.2 shows two real scenes recorded by us. Both images were captured with the camera Raytrix R5 using a main lens focal length of $f_L = 35\,\mathrm{mm}$. Here, real scenarios were captured which show a large depth range, to see how the algorithms behave for different object distances.

Furthermore, Hog et al. [2017] have also recorded a dataset based on a Raytrix R5 camera and presented the results obtained by their algorithm in the paper[1]. The three scenes recorded by Hog et al. [2017] show artificial scenarios as shown in Figure 8.3. Another dataset is made publicly available by the manufacturer Raytrix itself, consisting of four different scenes shown in Figure 8.4. The scenes have been recorded with a Raytrix R11 camera.

While the Raytrix R5 camera has a sensor resolution of 2048 pixel × 2048 pixel, the Raytrix R11 has a sensor resolution of 4016 pixel × 2688 pixel.

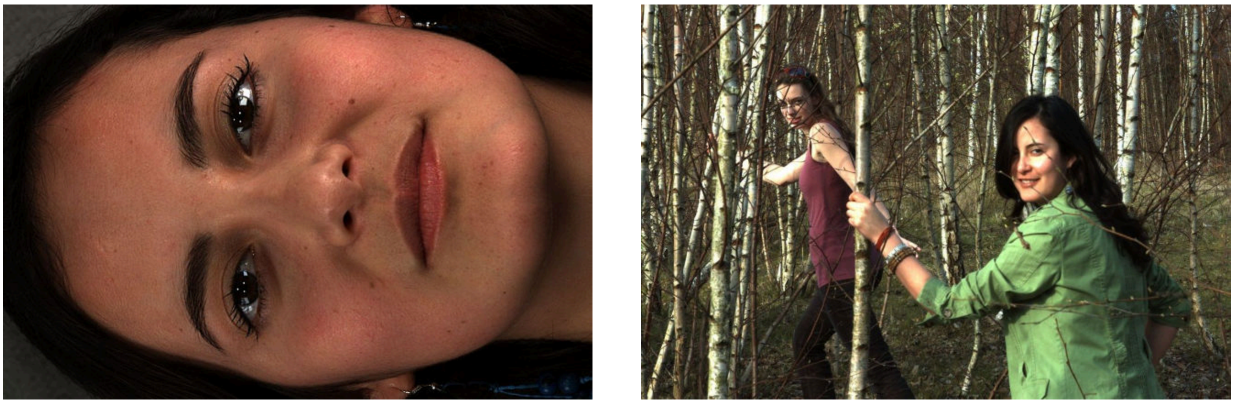
---

[1] At this point we want to thank Matthieu Hog who kindly supplied us with their dataset as well as the results of their algorithm (Hog et al. [2017]).

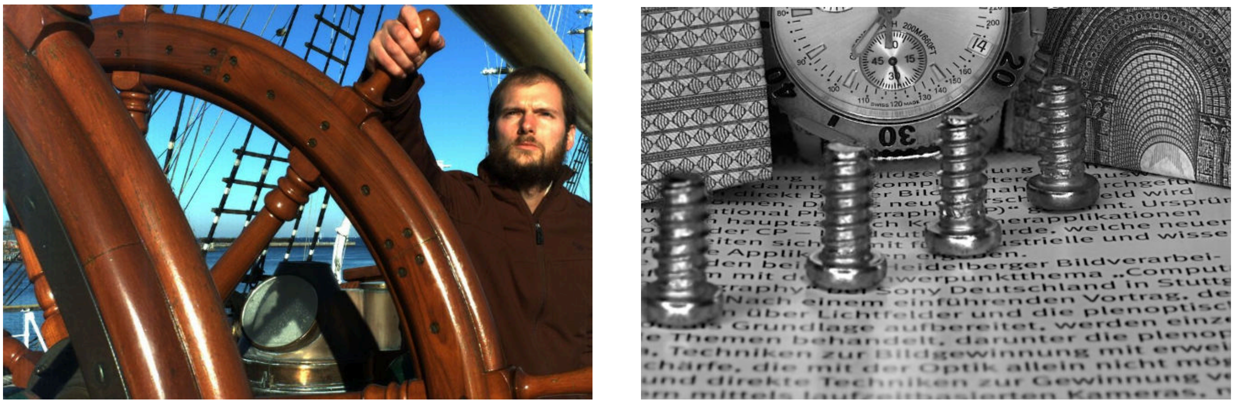


(a) Donkey (b) Tiger (c) Lion

Figure 8.3: Focused plenoptic camera dataset published by Hog et al. [2017]. All three images were recorded with a Raytrix R5 camera. The figures show the totally focused images synthesized from the recorded light fields.



(a) Andrea (b) Forest



(c) Pilot (d) Watch

Figure 8.4: Raytrix dataset. All four images were recorded by a Raytrix R11 camera. The figures show the totally focused images synthesized from the recorded light fields.

Method	depth pixel density			std. dev.			mean		
	z_{C1}	z_{C2}	z_{C3}	z_{C1}	z_{C2}	z_{C3}	z_{C1}	z_{C2}	z_{C3}
Ours (unfiltered)	0.23	0.37	0.44	0.027	0.044	0.033	0.182	0.276	0.321
Ours ($\sigma_z^2 < T(z)$)	0.14	0.31	0.39	0.009	0.014	0.015	0.186	0.287	0.326
Ours (filtered)	0.35	0.53	0.69	0.003	0.005	0.007	0.185	0.289	0.326
RxLive$_{(0.1)}$	0.06	0.08	0.10	0.084	0.086	0.098	0.234	0.260	0.286
RxLive$_{(0.25)}$	0.16	0.30	0.42	0.077	0.064	0.068	0.216	0.267	0.306
RxLive$_{(0.5)}$	0.14	0.26	0.48	0.066	0.059	0.053	0.204	0.279	0.317

Table 8.1: Inverse virtual depth statistics obtained from the checkerboard dataset. The table shows the depth pixel density, the standard deviation and the mean of the inverse virtual depth $z = v^{-1}$ calculated on the basis of all depth pixels across the checkerboard pattern.

Based on the presented data, the proposed plenoptic camera based depth estimation algorithm will be evaluated and will be compared to the method implemented in the RxLive software (Raytrix GmbH [2013]) and the one recently published by Hog et al. [2017].

8.2 Results

This section states the results which were obtained based on the dataset presented in the previous section. The proposed method (presented in Chapter 5) is compared to two other algorithms:

- the algorithm implemented in the RxLive software from the company Raytrix GmbH [2013]
- the algorithm of Hog et al. [2017], which was recently published

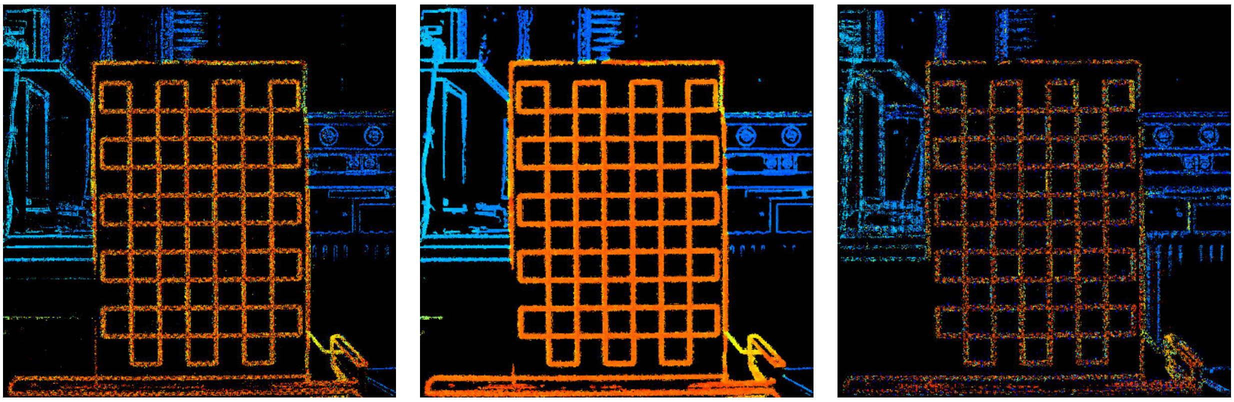
Both of these algorithms work, as the proposed method does, directly on the raw images recorded by a focused plenoptic camera and do not rely on any resampled light field representation such as sub-aperture images or EPIs. For the RxLive algorithm no details are available. It seems that the method performs traditional block matching on the micro images followed by strong filtering and hole filling. Hog et al. [2017] extract a, what they call stereo focal stack of image pairs. Using this stack of images, Hog et al. [2017] apply a standard stereo matching algorithm (Drazic and Sabater [2012]) to find point correspondences and estimate a focus map based on these correspondences.

The results of Hog et al. [2017] are available for their own dataset as well as for the Raytrix R11 dataset. Unfortunately, the dataset supplied by Hog et al. [2017] is not compatible with the RxLive software. Hence, results are presented only for the respective combination of data and algorithm. However, results for the RxLive algorithm based on the dataset of Hog et al. [2017] can be found in their publication. Due to the same reason, quantitative results based on the checkerboard dataset are only obtained for the method proposed by us and the RxLive algorithm.

The RxLive algorithm calculates, as the proposed method does, a map of virtual depth values v. It is not obvious how the focus parameter g_f of Hog et al. [2017] is connected to the virtual depth v. However, in Appendix A.1 a proof is given for the fact that $g_f = v$ actually holds true by definition.

8.2.1 Quantitative Results

The proposed algorithm as well as the RxLive algorithm, using three different settings, are applied to the checkerboard dataset. The RxLive algorithm works on a regular sub-pixel grid. Hence, three different sub-pixel resolutions were used for this algorithm. These are: RxLive$_{(0.1)}$ = 0.1 pixel,



(a) our unfiltered depth map (b) our filtered depth map (c) depth map RxLive$_{(0.25)}$

Figure 8.5: Virtual depth maps calculated for the checkerboard dataset exemplary for the object distance $z_{C1} = 1.2\,\mathrm{m}$. (a) Unfiltered depth map calculated by our algorithm. (b) Filtered depth map calculated by our algorithm. (c) Unfiltered depth map calculated by RxLive at a disparity resolution of 0.25 pixel (RxLive$_{(0.25)}$). The color range was set to $2 \leq v \leq 6.5$. Due to the Galilean configuration far objects result in a low virtual depth v (blue) and close objects in a high virtual depth v (orange). Pixels without depth and outside the color range are marked in black.

RxLive$_{(0.25)}$ = 0.25 pixel, and RxLive$_{(0.5)}$ = 0.5 pixel. No post-processing is applied to the RxLive algorithm.

In addition to the virtual depth $v = z^{-1}$, our algorithm offers an inverse virtual depth variance σ_z^2. Thus, for this algorithm, in addition to the completely unfiltered depth map, a second depth map is evaluated in which only those depth pixels are considered which have a variance σ_z^2 lower than a certain threshold $T(z)$, as defined in eq. (8.1).

$$\sigma_z^2(\boldsymbol{x}_V) < T(z) = \beta \cdot z(\boldsymbol{x}_V)^3 \tag{8.1}$$

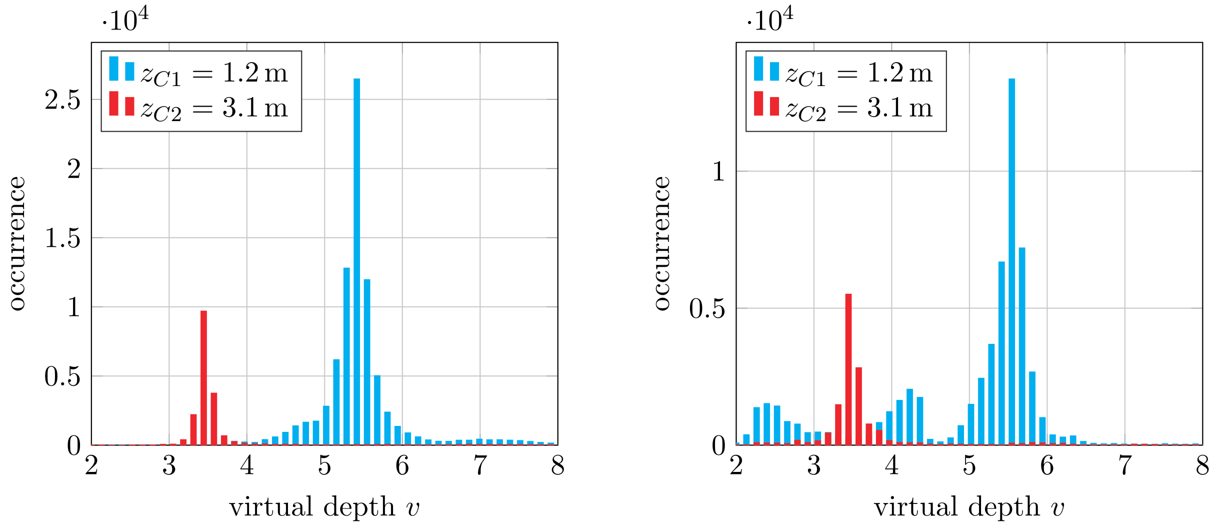
The threshold $T(z)$ is chosen as a third order function of z, corresponding to the assumptions laid out in Section 5.4. In eq. (8.1), β is just a scaling factor, which defines the point density of the resulting depth map. In the experiments, a scaling factor $\beta = 0.1$ was chosen. Furthermore, the depth map results after applying the filtering approach (presented in Section 5.6) are evaluated.

Figure 8.5, by way of example, shows the depth maps calculated for the object distance $z_{C1} = 1.2\,\mathrm{m}$. Figure 8.5a and 8.5b show the unfiltered and filtered depth maps calculated by the proposed algorithm while Figure 8.5c shows the depth map received from RxLive$_{(0.25)}$.

From Figure 8.5 one can already see that the outliers in our method are drastically reduced in the filtered depth map, while most of the details are retained. Furthermore, one can see that the depth map of RxLive$_{(0.25)}$ is significantly more sparse than the raw depth map resulting from our algorithm. It seems that the outliers of RxLive$_{(0.25)}$, especially on the checkerboard plane, are not statistically independent but occur in clusters.

Table 8.1 presents a complete list of statistics calculated for all algorithms based on the checkerboard dataset. This table shows the depth pixel density, the empirical standard deviation of the inverse virtual depth z, and the mean of the inverse virtual depth z, across the checkerboard pattern respectively. The depth pixel density is defined as the ratio between the number of valid depth pixels and the total number of pixels within the region of interest.

One can see that without removing any outliers our method has a higher depth pixel density than the RxLive algorithm for all object distances. At the same time, the standard deviation



(a) our algorithm (unfiltered)

(b) RxLive (unfiltered)

Figure 8.6: Virtual depth histograms of unfiltered virtual depth maps. (a) Histograms of the unfiltered virtual depth maps calculated by our algorithm for $z_{C1} = 1.2\,\text{m}$ and $z_{C2} = 3.1\,\text{m}$. (b) Histograms of the unfiltered virtual depth maps calculated by RxLive for z_{C1} and z_{C2}.

of our algorithm is lower for all object distances. Introducing the threshold on the variance σ_z^2 of course slightly reduces the pixel density. However, at the same time, the standard deviation is significantly reduced. Applying the probabilistic filtering approach improves the standard deviation by one order of magnitude with respect to the unfiltered one. Of course filtering on a planar object does not offer any insight into the quality of the resulting depth map. However, Figure 8.5 shows that the filter retains most of the details and seems to preserve edges quite well. Furthermore, the following Section 8.2.2 will give more details about the performance of the filtering process.

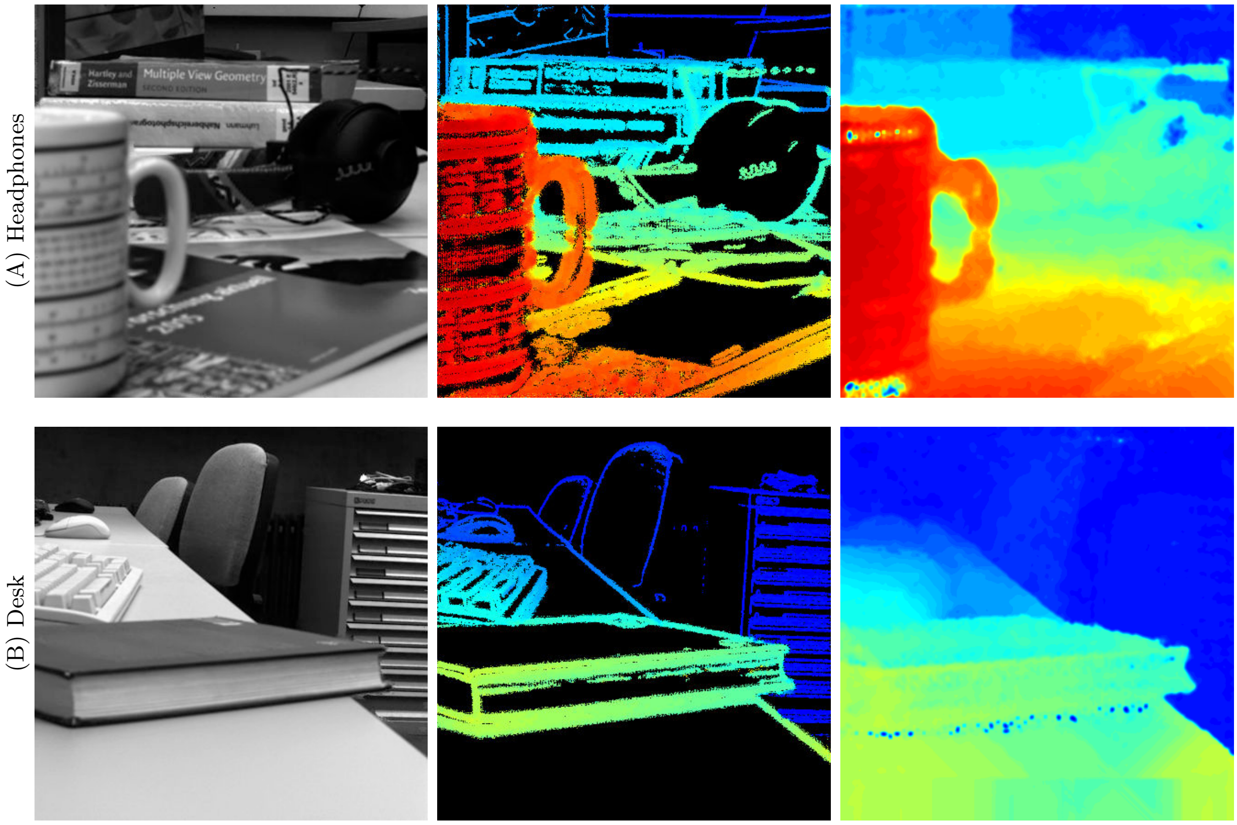
Figure 8.6 shows an example of the virtual depth histograms across the checkerboard target, for our algorithm and RxLive$_{(0.25)}$, for the object distances $z_{C1} = 1.2\,\text{m}$ and $z_{C2} = 3.1\,\text{m}$. Especially for z_{C1} one can see that the outliers of the RxLive$_{(0.25)}$ have some systematic characteristics as it was already visible in Figure 8.5c.

8.2.2 Qualitative Results

The algorithm of Hog et al. [2017] performs pixel-wise depth estimation for all pixels in the virtual image. Hence, even for points which do not carry any information at all, an estimate is obtained. This often results in implausible estimates, e.g. negative values. Thus, the estimates are limited to a plausible range. In the following figures showing depth maps, black pixels represent pixels which are either outside the virtual depth range or do not carry any information at all. For the RxLive algorithm, all settings were set to “high” to obtain reproducible results. For some cases, this might result in over-smoothed depth maps.

Figure 8.7 shows the results for our own recorded dataset. The figure shows the totally focused images synthesized by our algorithm and the filtered depth maps of our method as well as the RxLive algorithm. While the “Headphone” scene contains large textured regions, the “Desk” scene has many homogeneous regions for which no stereo matching can be performed.

Figure 8.8 shows the results for the dataset of Hog et al. [2017]. All three scenes show a small figure in front of a dark background. Hog et al. [2017] supply the raw data for all three

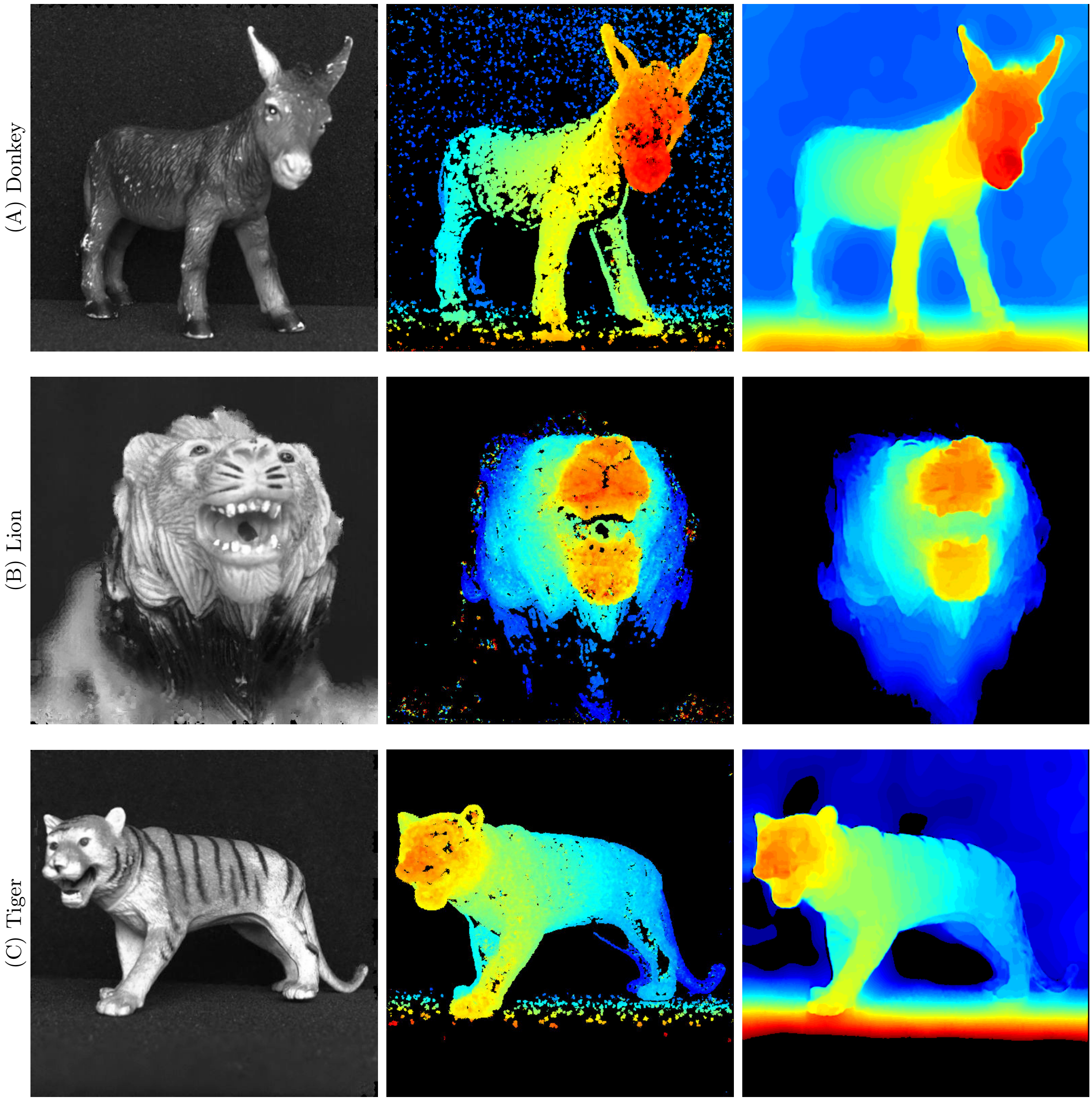


(a) our totally focused image (b) our filtered depth map (c) filtered depth map RxLive

Figure 8.7: Results for our own Raytrix R5 dataset. The figure shows, from left to right, the totally focused image synthesized by our method, the filtered virtual depth map of our algorithm and the refined virtual depth map of the RxLive algorithm. The color range (from blue to red) was set to $2 \leq v \leq 12$. Pixels without depth and outside the color range are marked in black.

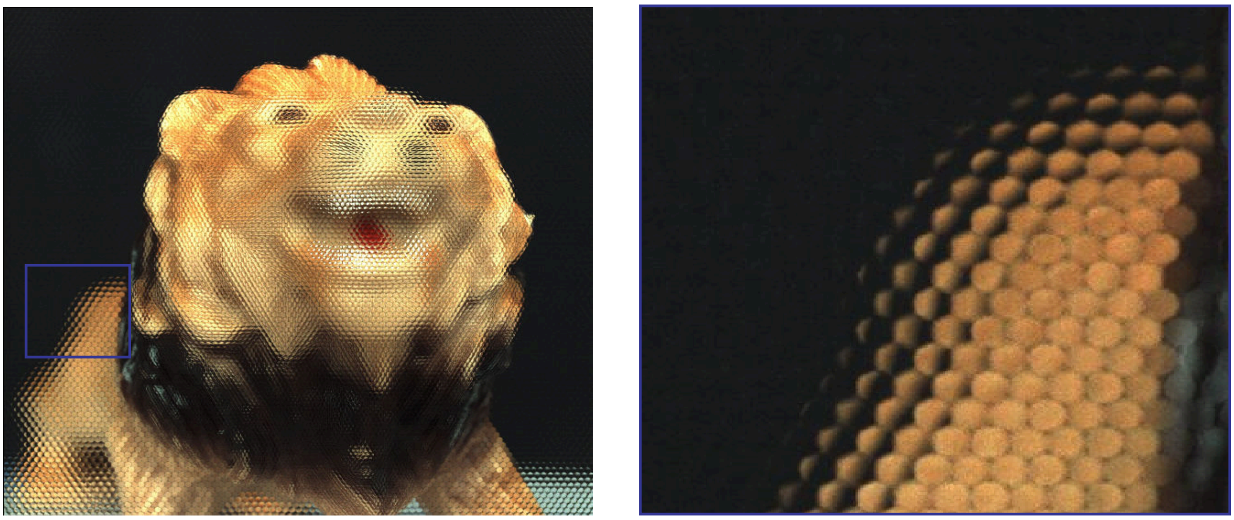
scenes as well as a recorded white image. This white image which was recorded by mounting a white balance filter in front the camera's main lens. However, the recorded white image is quite underexposed and therefore the resulting vignetting-corrected and demosaiced images are quite noisy. For the "Lion" one can see in Figure 8.8Bb that there are many outlying estimates around the head of the lion. Furthermore, by our algorithm no depth estimates were obtained at the back of the lion, even though there are textured image regions. The reason for this can be seen in Figure 8.9, which shows the recoded raw image for this dataset. From the close-up (Fig. 8.9b), one can see that the micro lenses actually produce micro images which are mirrored with respect to the real scene. While in the real scene the edge produced by the lion's back goes from left (background) to right (foreground), the edge in the micro images runs in the opposite direction. This indicates that for this region, the plenoptic camera operates in the Keplerian mode, rather than in the Galilean mode. This means that the main lens image is formed in front of the MLA and is captured as a real image by the micro lenses. This case, which results in negative virtual depths, is not covered by our algorithm. Furthermore, Raytrix cameras are not designed to perform in the Keplerian mode and therefore will produce out-of-focus micro images. The algorithm of Hog et al. [2017] seems to be able to deal with this case. Though, as can be seen from the results in their publication, it also produces obviously erroneous estimates.

Finally, Figure 8.10 and Figure 8.11 show the results for the R11 dataset supplied by the company Raytrix. While Figure 8.10 shows the synthesized totally focused images of our method



(a) our totally focused image (b) our filtered depth map (c) depth map Hog et al. [2017]

Figure 8.8: Results for the Raytrix R5 dataset of Hog et al. [2017]. The figure shows, from left to right, the totally focused image synthesized by our method, the refined virtual depth map of our algorithm and the virtual depth map (or focus map) of Hog et al. [2017]. The color range (from blue to red) was set to $2 \leq v \leq 6$. Pixels without depth and outside the color range are marked in black.



(a) processed raw image (b) marked subsection of (a)

Figure 8.9: Processed raw image of the “Lion” dataset of Hog et al. [2017]. (a) Complete image. (b) Zoomed subsection of (a). At the back of the lion the micro lenses produces mirrored micro images, as one can see at the transition from the lion to the background. This indicates that here the micro images were formed by the Keplerian mode, which means that the main lens image was formed in front of the MLA.

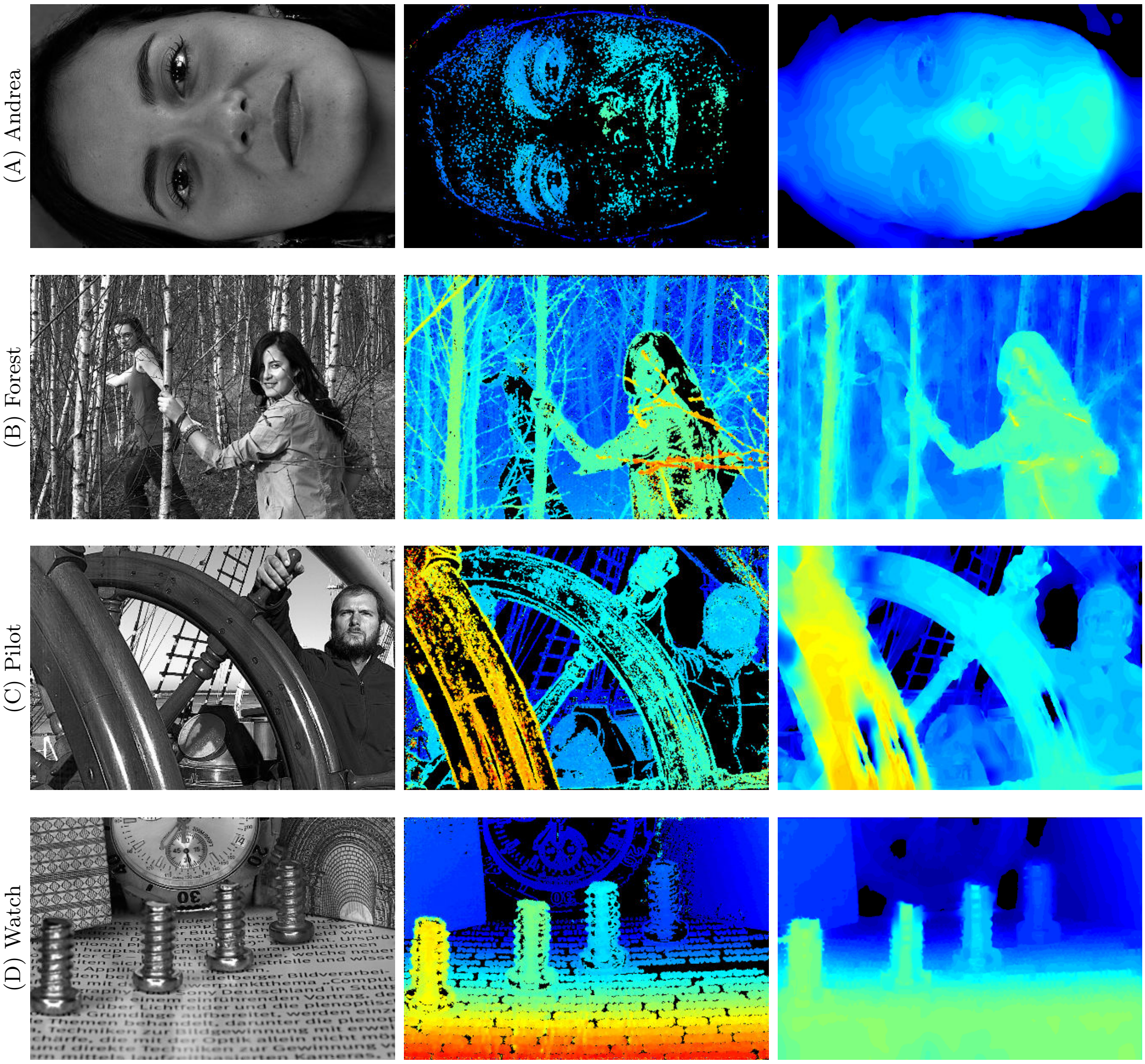
along with our refined depth maps as well as the depth maps of Hog et al. [2017], Figure 8.11 shows the unfiltered depth maps of our method and the RxLive algorithm, as well as the refined depth maps of the RxLive approach.

Figure 8.12 shows once more the results of our algorithm for the “Tiger” dataset of Hog et al. [2017]. While Figure 8.12b shows the refined virtual depth map, Figure 8.12c shows the corresponding inverse virtual depth variance σ_z^2 in a logarithmic scale. The blue regions represent a low variance, while red regions correspond to high variances. The figure shows quite clearly that our algorithm is very certain about its estimates in regions of high intensity gradients. This is the case, for instance, around the nose and the eyes of the tiger, where dark spots are on a bright ground, or at the stripes on the tiger’s back. Instead, in regions of less texture such as the floor, the algorithm is quite uncertain about its estimates. One also can see that the average variance increases with increasing distance (decreasing virtual depth). This is due to smaller stereo baselines as well as fewer stereo observations.

In the Figures 8.7, 8.8, and 8.10 the left column shows the totally focused images synthesized by our method for the respective dataset. This implementation produces grayscale images but can be easily extended to the calculation of color images.

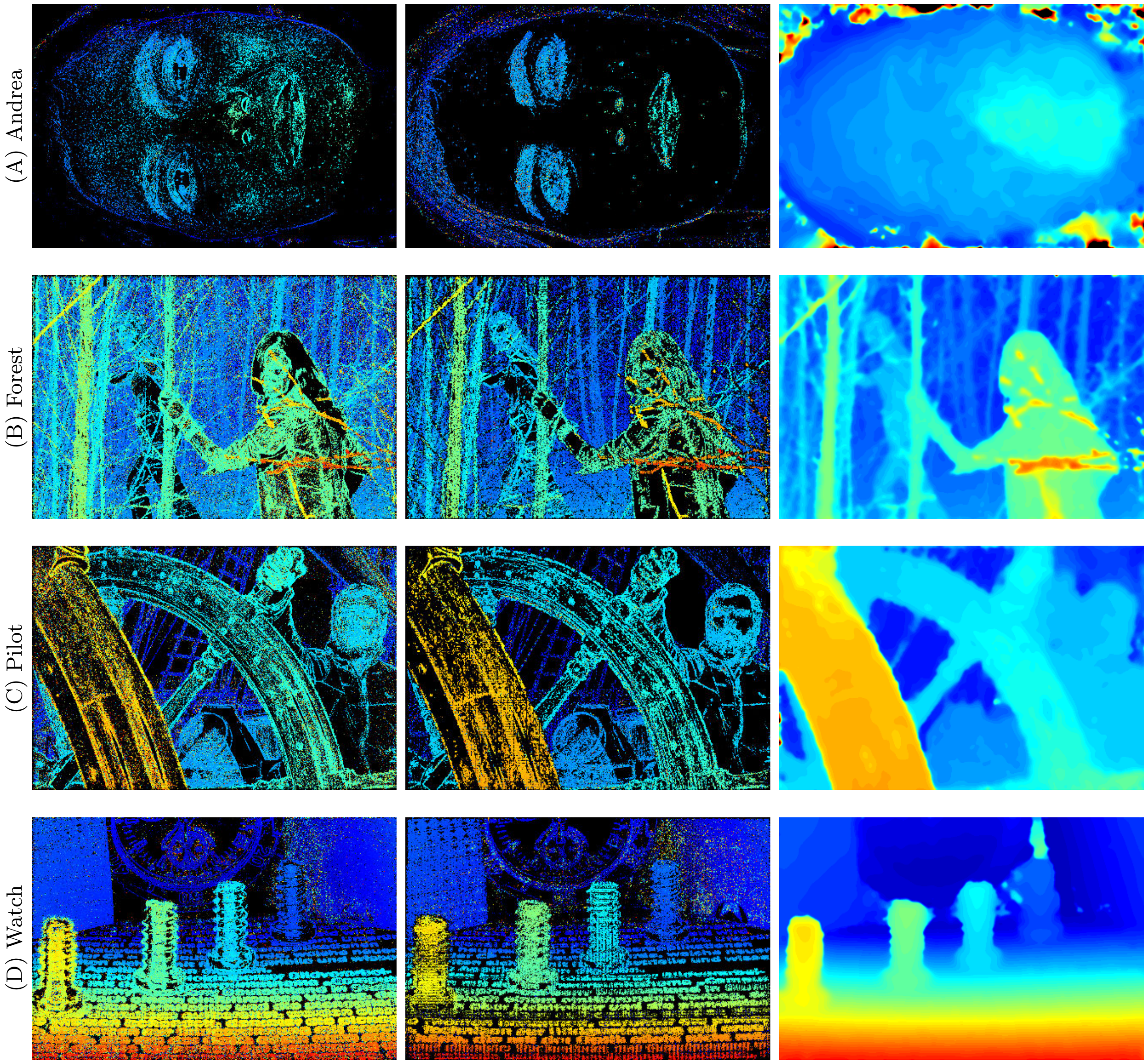
8.3 Conclusion

Regarding the quantitative numbers which are obtained from the checkerboard dataset, our method significantly outperforms the RxLive algorithms. However, as these numbers represent only depth estimates from a planar surface, no statement can be made about the overall quality of the depth map. Nevertheless, from the histograms in Figure 8.6 one can see that the unfiltered depth estimates of our algorithm are more symmetrically distributed than the results from the RxLive algorithm and have a much lower variance.



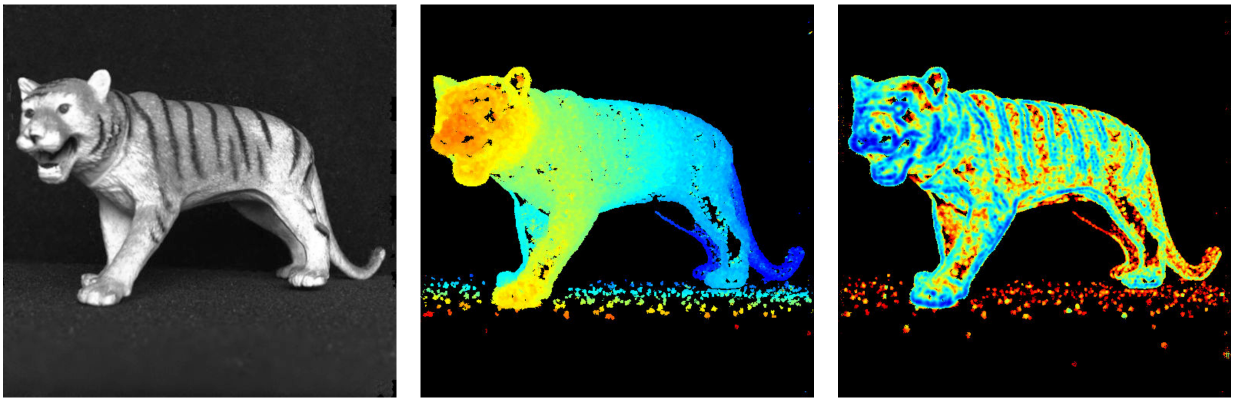
(a) our totally focused image (b) our filtered depth map (c) depth map Hog et al. [2017]

Figure 8.10: Results for the Raytrix R11 dataset. The figure shows, from left to right, the totally focused image synthesized by our method, the refined virtual depth map of our algorithm and the virtual depth map (or focus map) of Hog et al. [2017]. The color range (from blue to red) was set to $2 \leq v \leq 6$ for the first three datasets and to $2 \leq v \leq 14$ for the "Watch" dataset. Pixels without depth and outside the color range are marked in black.



(a) our unfiltered depth map (b) unfiltered depth map RxLive (c) filtered depth map RxLive

Figure 8.11: Further results for the Raytrix R11 dataset. The figure shows, from left to right, the unfiltered virtual depth map of our algorithm, the unfiltered virtual depth map of the RxLive algorithm and the refined virtual depth map of the RxLive algorithm. The color range (from blue to red) was set to $2 \leq v \leq 6$ for the first three datasets and to $2 \leq v \leq 14$ for the "Watch" dataset. Pixels without depth and outside the color range are marked in black.



(a) totally focused image (b) virtual depth map (c) inverse virtual depth variance

Figure 8.12: Estimation uncertainty obtained by our depth estimation algorithm. (a) Totally focused image, (b) virtual depth map, (c) logarithmic inverse virtual depth variance $\ln(\sigma_z^2)$. In (c) blue represents low variances while red shows high variances.

Perhaps the first thing one recognizes when comparing the depth maps is that our algorithm does not supply a complete depth map at any stage of refinement. The algorithm by Hog et al. [2017] instead calculates a dense, pixel-wise depth map, while the RxLive algorithm aims to fill holes entirely in a refinement stage. Even though the dense depth maps might look more appealing, they are not beneficial for the purpose of camera tracking. A plenoptic camera is a passive sensor, which means that no depth can be obtained for homogeneous regions. For these regions depth values are only obtained when the depth map is constrained to be smooth. However, there is no guarantee that this smoothness is also fulfilled by the recorded scene. For the purpose of camera tracking, it is, in fact, preferable to have undefined regions rather than wrong depth estimates.

In some regions, the filled and refined depth maps calculated by the RxLive software show large clusters of false estimates (see Figures 8.7 and 8.11). This seems to be especially true for regions with discontinuities in depth and in regions of low texture. The algorithm proposed in this thesis seems to perform better with regard to those regions. Based on the foreground to background segmentation in our filtering method (Section 5.6.2) discontinuities in the depth map are preserved very well. Furthermore, noisy estimates in low textured regions are reliably detected as outliers.

Comparing the raw depth maps of the method proposed in this thesis with the RxLive approach (Figure 8.11) shows that the proposed method performs significantly better at object boundaries. This is probably due to the 1D pixel patch which is used in our algorithm compared to the 2D patch used in the RxLive approach. Our algorithm obtains depth estimates for the faces of persons, which are quite low textured regions. However, it is interesting to notice that our algorithm does not obtain any depth for the black hair in the “Andrea” and “Forest” dataset. This might due to the fact that the raw data is stored as 8 bit grayscale images after performing vignetting correcting and debayering. Hence, the corresponding intensity differences might fall below the quantization threshold.

The algorithm of Hog et al. [2017] produces much smoother and, as previously mentioned, dense depth maps. However, it produces large clusters of outliers in certain regions. Wrong estimates are obtained for specular reflecting surfaces, as can be seen in the case of the eyes in the “Andrea” dataset and the steering wheel in the “Pilot” dataset. False estimates are also visible along the stripes in the “Tiger” dataset as well as on the whisker in the “Lion” dataset. The method of Hog et al. [2017] seems to also produce false estimates for very close regions, as can be seen in the "Watch" dataset. On explanation for this effect could be that here, due to the very high virtual depth ($v \approx 13$), all micro images are out of focus which in turn creates problems for the algorithm.

However, in the end one must state that both the method by Hog et al. [2017] as well as our method have their advantages. The depth maps of Hog et al. [2017] have less stochastic noise, while our method produces fewer outliers. With respect to the task of camera tracking, an algorithm can deal much better with stochastic noise as long as it is symmetrically distributed and uncorrelated.

Another advantage of our algorithm compared to the others is that an uncertainty measure is obtained for each estimate (see Figure 8.12) which allows for weighting estimates in a probabilistically correct manner.

9 Experiments – Plenoptic Camera Calibration

9.1 Data

In Chapter 6 plenoptic camera calibration algorithms at different stages of complexity were presented. Depth conversion functions were defined (Section 6.1), which can be calculated based on a set of range measurements using a planar target. Furthermore, complete plenoptic camera models at different levels of abstraction were presented (Section 6.2) which can be solved in a bundle adjustment (Section 6.3).

A set of images at different object distances was recorded to estimate the depth conversion functions. Furthermore, recordings from a 3D calibration target were made to perform bundle adjustment based calibrations.

Since basically all state-of-the-art algorithms rely on the recordings of a planar calibration target, in addition a set of images from a planar checkerboard pattern was recorded to compare our approach to the one presented by Bok et al. [2014, 2017].

All experiments were performed with a Raytrix R5 camera (image resolution: 2048 pixel × 2048 pixel, sensor size: 11.264 mm × 11.264 mm, diameter of micro-lenses: ~23 pixel, aperture: f/2.8). For some experiments datasets were recorded using various main lens focal lengths for the plenoptic camera: $f_L = 12\,\text{mm}$, $f_L = 16\,\text{mm}$, and $f_L = 35\,\text{mm}$.

9.1.1 Series of Range Measurements

A series of range measurements was recorded based on the setup shown in Figure 9.1a. As shown in the figure, a checkerboard pattern is placed perpendicularly to the optical axis of the plenoptic camera's main lens at various distances. The distance to the pattern is measured from a reference point behind the camera using a laser rangefinder (LRF). For the plenoptic camera, a main lens focal length $f_L = 35\,\text{mm}$ was used.

For this series, the checkerboard pattern shown in the figure was recorded at 48 different object distances in the range from $z_C = 0.85\,\text{m}$ to $z_C = 5.02\,\text{m}$. All object distances are more or less uniformly distributed across the entire range, while the spacing between two distances ranges from 5 cm to 10 cm. Since the pattern on the calibration target has 54 reference points, 54 measurement points are received for each recorded target.

Even though the checkerboard pattern was precisely aligned to the plenoptic camera, one cannot guarantee that the optical axis of the plenoptic camera's main lens stand precisely perpendicularly on the checkerboard. Hence, the range measurements obtained by the LRF are refined as described below.

One is able to detect the checkerboard corners in the totally focused image of the plenoptic camera and one knows their position on the target, with the exception of a scaling factor s.

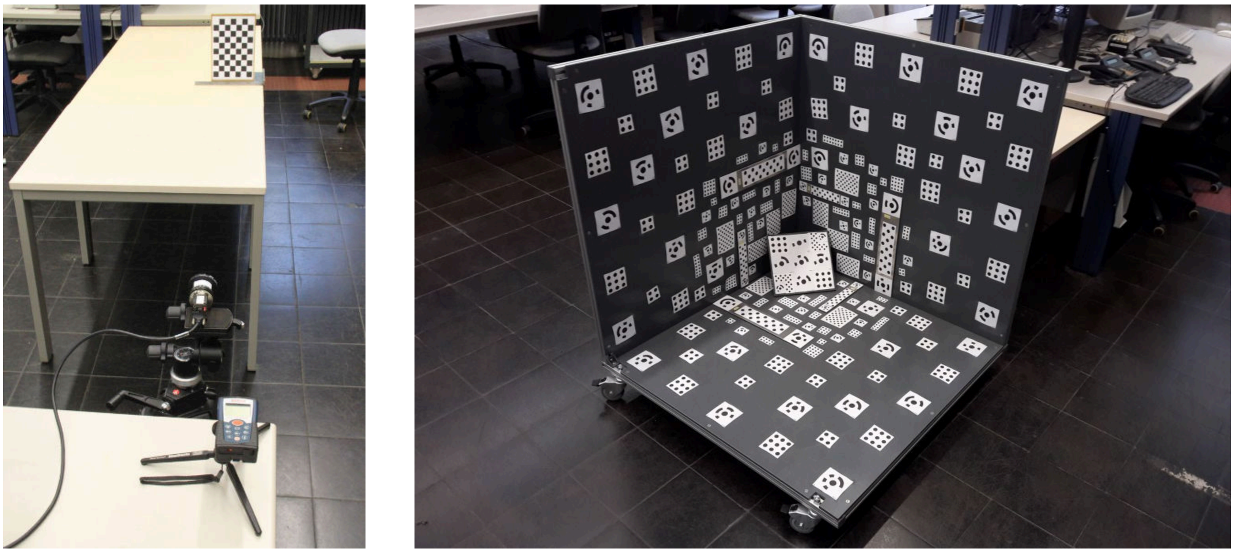

(a) range series setup (b) 3D calibration target

Figure 9.1: Setups for plenoptic camera calibration. (a) Setup to obtain a series of range measurements. A checkerboard target is moved along the optical axis of the plenoptic camera. The target distance is measured by a laser rangefinder (LRF). (b) 3D calibration target used to perform camera calibration based on a bundle adjustment. The target consists of unordered, uniquely coded and uncoded calibration markers.

Thus, the relation between coordinates on the checkerboard $\boldsymbol{x}_{CB} = [x_{CB}, y_{CB}, z_{CB}]^T$ and the undistorted virtual image coordinates $\boldsymbol{x}_V = [x_V, y_V]^T$ can be defined as given in eq. (9.1). For the sake of simplification, the totally focused images of the plenoptic camera are considered to conform to those of a pinhole camera.

$$\eta \begin{bmatrix} x_V \\ y_V \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s\boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \cdot \begin{bmatrix} x_{CB} \\ y_{CB} \\ z_{CB} \\ 1 \end{bmatrix} \tag{9.1}$$

The matrix $\boldsymbol{R} \in \mathrm{SO}(3)$ defines the 3D rotation and $\boldsymbol{t} \in \mathbb{R}^3$ the 3D translation from checkerboard coordinates to camera coordinates. Since the intrinsic camera parameters (f, c_x, and c_y) are known, based on the undistorted recorded image point $\boldsymbol{x}_V$, the matrix $\boldsymbol{R}$ and the vector with respect to the scaling factor $\tilde{\boldsymbol{t}} = \boldsymbol{t} \cdot s^{-1}$ can be estimated. For the case that the checkerboard coordinates $\boldsymbol{x}_{CB}$ are defined with the x-y-plane being on the checkerboard plane ($z_{CB} = 0$), the third coefficient of the translation vector equals the object distance $[\boldsymbol{t}]_3 = z_C$. Between the scaled object distance $\tilde{z}_C = z_C \cdot s^{-1}$ and the target distances measured by the LRF d_{LRF}, the following linear relation can be defined:

$$d_{LRF} = s \cdot \tilde{z}_C + z_0. \tag{9.2}$$

Both the scaling factor s and the offset z_0 (between LRF and camera) are constant and thus can be estimated based on all measured target distances.

9.1.2 Recordings of 3D Calibration Target

Figure 9.1b shows the 3D target which is used for our bundle adjustment based calibration approaches. A 3D target is used, since it results in a better conditioned optimization in comparison

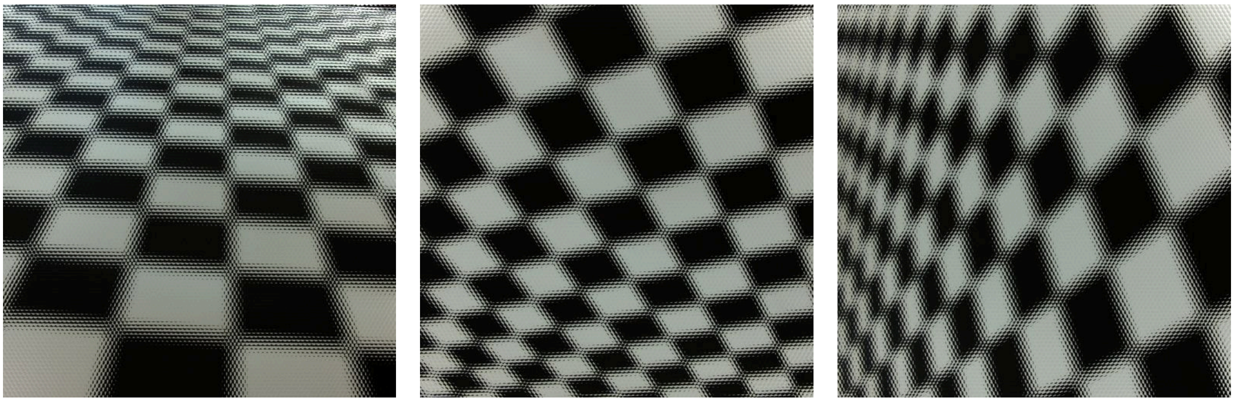


Figure 9.2: Raw image samples of a checkerboard pattern used for the calibration method of Bok et al. [2017].

to a 2D target, for instance. Our calibration target consists of an unordered set of markers. For all of these markers, the positions in the 3D space are estimated during the calibration and therefore the calibration does not rely on any model assumptions. The only model assumption which is made on the calibration target is that six distances between certain marker points are known[1]. In this way, the scale of the calibration target can be retrieved. Since multiple of these reference distances are used, possible errors in the distances will level each other out.

Images of the calibration target were recorded using three different main lenses with different focal lengths ($f_L = 12.5\,\mathrm{mm}$, $f_L = 16\,\mathrm{mm}$, $f_L = 35\,\mathrm{mm}$). For each focal length between 70 and 94 images were recorded from views that are as different as possible from each other. The images were recorded in a distance of approximately 0.3 m up to 4.5 m in front of the calibration target.

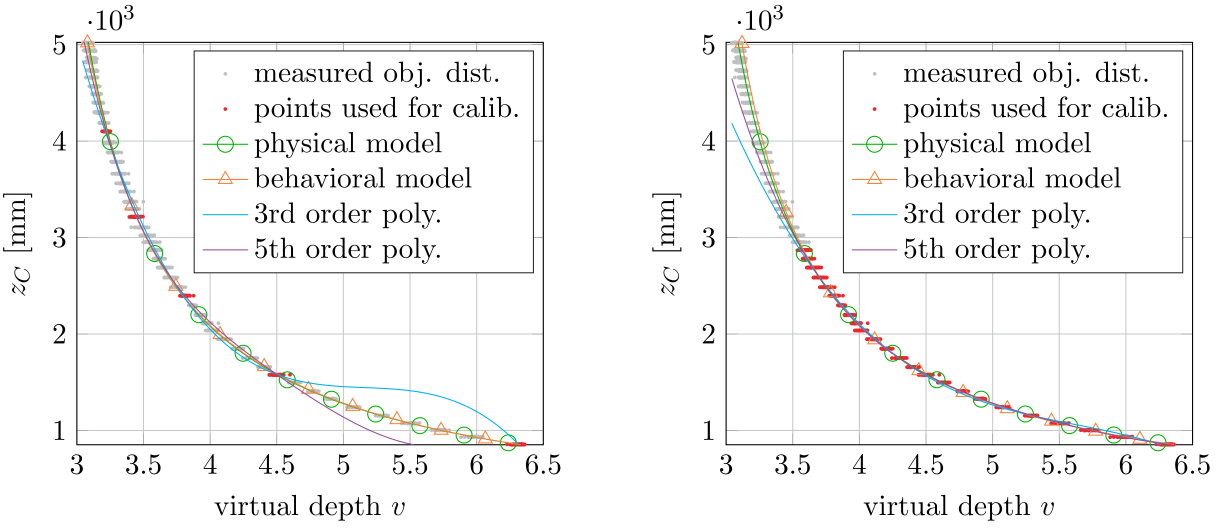
Our calibration approaches are also supposed to be compared to other algorithms. For this purpose, two further sets of images were recorded based on the $f_L = 16\,\mathrm{mm}$ main lens focal length. On the one hand, another set based on our calibration target (Figure 9.1b) and on the other hand a set of a planar checkerboard pattern were recorded. These images of the checkerboard pattern conform to the type of data used by Bok et al. [2017]. Figure 9.2 shows sample images of the checkerboard pattern. For both targets a set of 64 images was recorded which will be used to perform a statistical evaluation of our calibration approaches in comparison to the one by Bok et al. [2017].

9.2 Results

Based on the recorded datasets different experiments were performed. Section 9.2.1 presents the results of the depth conversion functions which were introduced in Section 6.1. In Section 9.2.2 the bundle adjustment based approaches are investigated. The results show how the calibration behaves on one side for the different camera models and on the other side for different camera parameters (i.e. different main lenses). The presented results also show the depth accuracy of the camera by using different models and camera parameters. Furthermore, the results of the bundle adjustment based calibration approaches presented in this thesis are compared to the results obtained by the method of Bok et al. [2017]. We explicitly show how the consideration of squinting micro lenses in the camera model affect the entire projection model.

---

[1] Distances are obtained from accurate scale standards mounted on the calibration target.



(a) (b)

Figure 9.3: Estimated depth conversion functions. Based on a series of range measurements, the physical model, the behavioral model and two polynomials are fitted to the data. (a) estimated functions using only five distinct object distances. (b) estimated function using only the lower half of the range measurements.

9.2.1 Depth Conversion Functions

Based on the recorded series of range measurements different experiments were performed to evaluate the depth conversion functions presented in Section 6.1.

In a first experiment, only the measured object distances of five recorded targets were used for calibration. This experiment was performed to evaluate whether a low number of calibration points is sufficient to estimate the respective function reliably.

In a second experiment, only the measured points with an object distance of less than 2.9 m were used for calibration. In this experiment, the objective was to investigate how strong the estimated functions drift from the measured data outside the range of calibration.

Figure 9.3a shows the results corresponding to the first experiment. The red dots represent the calibration points of the five object distances which were used for calibration. The gray dots are the remaining measured points which were not used for calibration. As one can see, the physical model as well as the behavioral model are almost congruent. Both curves match the measured distances very well over the entire range of 0.85 m to 5.02 m. For the polynomials of orders three and five however, five object distances are not sufficient to approximate the conversion function from virtual depth v to object distance z_C accurately. Both functions fit to the points used for calibration but do not reflect the underlying model properly between the calibration points.

Figure 9.3b shows the results of the second experiment. Again, the points used for calibration are represented as red dots and the gray dots represent the remaining points. Both model based calibration approaches are, again, almost congruent and describe the measured distances in the range of calibration up to 2.9 m very well. In this range the estimated polynomials also fit the measured distances very well. Nevertheless, for object distances larger than 2.9 m especially the third-order, but also the fifth-order polynomial, drift from the measurements. The functions of both model-based approaches still match the measured values very well up to an object distance of 5.02 m. Nevertheless, the physical model still fits the data slightly better than the behavioral model which has one more degree of freedom. The results show that both model-based functions are able to convert the virtual depth to an object distance even outside the range of calibration points with good accuracy.

9.2.2 Bundle Adjustment based Plenoptic Camera Calibration

Based on the datasets recorded from our 3D calibration target different experiments were performed. This section presents the intrinsic parameters for all introduced camera models estimated on the basis of the datasets described in Section 9.1.2. Furthermore, the accuracy of the 3D points of the calibration target which are estimated in the bundle adjustment is measured. In addition, the measurement accuracy of the plenoptic camera for the different main lens focal lengths and for different model parameters is evaluated. The robustness of the calibration approach proposed in this thesis is compared to the method of Bok et al. [2017]. For the sake of simplicity, numbers are assigned to the various camera models. These numbers are as follows:

1. virtual image projection model (Section 6.2.1)
2. micro images projection model (Section 6.2.2)
3. extended micro images projection model (Section 6.2.3)

A fourth model is defined which is not taken into consideration for all experiments. Instead, this model is used to show the effect of applying a distortion model to micro image points as done by model #3. Hence, this fourth model is defined as follows:

4. micro images projection model which considers squinting micro lenses but no pixel distortion

Estimated Parameters

For each of the three lenses the camera parameters are estimated and the reprojection error as well as the accuracy of the 3D points estimated during the bundle adjustment are evaluated. This allows to assess the validity of the camera models and the robustness of the calibration approaches.

Table 9.1 shows the estimated intrinsic parameters for all three lenses and the corresponding reprojection errors using the three models introduced in Section 6.2. Here, s_x and s_y are the root mean square (RMS) values of the reprojection error in the x- and y-coordinate respectively.

From Table 9.1 one can see that for all three main lenses the reprojection errors are significantly smaller than one pixel. This confirms the validity of the complete projection model. It seems that the reprojection error is correlated with the main lens focal length f_L. Another indication for the validity of the defined projection model is that the estimated main lens focal length f_L is quite close to the nominal focal length of the respective lens. In addition, the estimated parameter B is similar for all three lenses. The parameter B is a constant of the plenoptic sensor and therefore does not depend on the main lens.

To evaluate the accuracy of the estimated 3D object coordinates, the point clouds received from the bundle adjustment for all three main lenses are registered with a reference point cloud using the Iterative Closest Point (ICP) algorithm (Besl and McKay [1992]). The reference point cloud is received from a bundle adjustment using a standard monocular camera (standard SfM approach on the basis of the calibration markers). For this a professional photogrammetric measurement software was used to estimate a highly accurate reference point cloud. Table 9.2 shows the root

Model	Lens	f_L [mm]	b_{L0} [mm]	B [mm]	c_{Lx} [pixel]	c_{Ly} [pixel]	s_x [pixel]	s_y [pixel]
#1	12.5 mm	12.619	11.702	0.389	1006.2	1042.7	0.093	0.092
	16 mm	16.277	15.427	0.380	1018.2	1053.9	0.076	0.078
	35 mm	34.837	33.968	0.366	1022.4	1034.1	0.079	0.084
#2	12.5 mm	12.614	11.787	0.353	1003.3	1044.0	0.213	0.221
	16 mm	16.273	15.482	0.357	1015.7	1056.3	0.123	0.126
	35 mm	34.869	33.993	0.367	1023.8	1042.3	0.063	0.067
#3	12.5 mm	13.181	12.209	0.399	1006.5	1041.5	0.202	0.208
	16 mm	16.748	15.893	0.376	1018.7	1054.2	0.108	0.109
	35 mm	35.368	34.471	0.370	1022.0	1031.0	0.058	0.062

Table 9.1: Estimated intrinsic camera parameters and reprojection errors for three different main lens focal lengths.

		Lens		
		12.5 mm	16 mm	35 mm
#1	RMSE	0.282 mm	0.308 mm	0.333 mm
	MAE	2.160 mm	2.345 mm	2.371 mm
#2	RMSE	0.290 mm	0.301 mm	0.332 mm
	MAE	2.124 mm	2.352 mm	2.409 mm
#3	RMSE	0.282 mm	0.310 mm	0.335 mm
	MAE	2.147 mm	2.339 mm	2.399 mm

Table 9.2: Accuracy of 3D object coordinates estimated during the bundle adjustment.

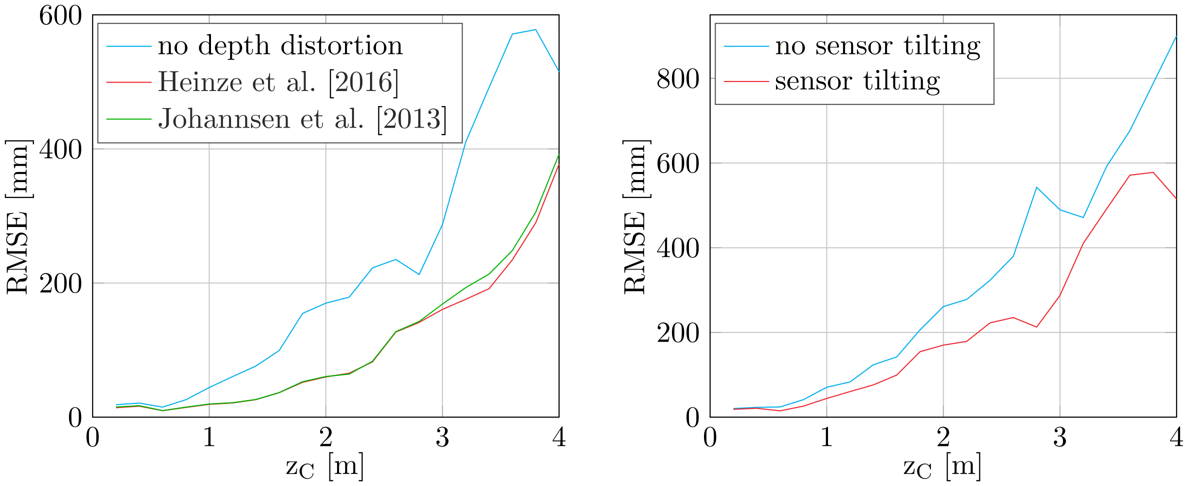
mean square error (RMSE) and the maximum absolute error (MAE) between the point clouds received from the plenoptic camera based bundle adjustment and the reference point cloud. For all three main lenses a RMSE of the point cloud of less than 1 mm was measured. The MAE is less than 2.5 mm for all three lenses.

Depth Accuracy of the Plenoptic Camera

In this section the depth accuracy of the plenoptic camera with respect to the parameters received from the bundle adjustment is evaluated. The depth accuracy is evaluated only based on model #1, since this is the only model for which the virtual depth can be estimated beforehand. For all other models, which define the projection directly to the micro images, the virtual depth would have to be estimated after the calibration. However, the results are expected to be similar to the ones presented here.

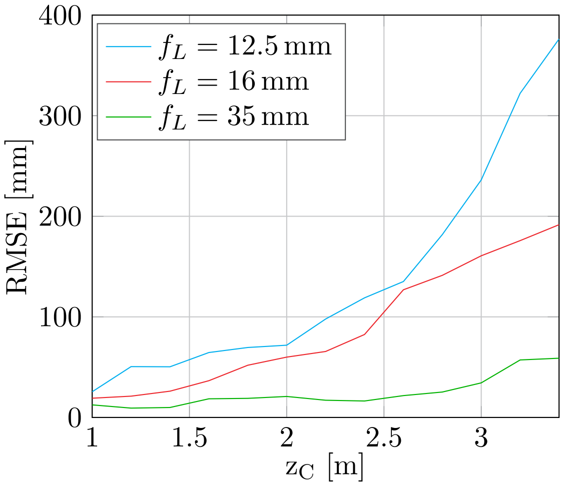
From the bundle adjustment the camera orientation for each recorded image is obtained. Furthermore, the precise 3D positions of all markers on the calibration target are known. To obtain the depth accuracy, a marker point detected in the totally focused image is projected to the object space, using the intrinsic camera model and the estimated virtual depth. In the object space, the error along the z_C domain between the projected and the real marker is calculated. Statistical values are calculated on the basis of all points within a range of ±10 cm around the object distance z_C of interest.

Figure 9.4a shows the depth accuracy while choosing various depth distortion models. As one can see, the accuracy is drastically improved when a depth distortion model is introduced.



(a) various depth distortion models

(b) sensor tilting



(c) various main lens focal lengths

Figure 9.4: Depth accuracy of a focused plenoptic camera for different model parameters and camera setups. (a) Depth accuracy for various depth distortion models. (b) Depth accuracy with and without modeling a tilted plenoptic sensor with respect to the main lens. For both (a) and (b) the $f_L = 16\,\mathrm{mm}$ dataset was used. (c) Depth accuracy for various main lens focal lengths. Here, the complete model as described in Section 6.2.1 was used.

		f_L [mm]	b_{L0} [mm]	B [mm]	c_{Lx} [pixel]	c_{Ly} [pixel]	f_x [pixel]	f_y [pixel]	K_1	K_2
#1	mean	16.30	15.34	0.441	1019	1047	*2869*	*2869*	*2.100*	*552.6*
	st. dev.	0.016	0.068	0.033	0.726	0.434	*6.372*	*6.372*	*0.020*	*47.13*
	rel. std.	0.10%	0.44%	7.41%	0.07%	0.04%	*0.22%*	*0.22%*	*0.97%*	*8.53%*
#2	mean	16.28	15.53	0.358	1015	1047	*2889*	*2889*	*2.039*	*689.4*
	st. dev.	0.003	0.018	0.007	0.708	0.395	*2.071*	*2.071*	*0.007*	*14.93*
	rel. std.	0.02%	0.12%	1.98%	0.07%	0.04%	*0.07%*	*0.07%*	*0.32%*	*2.17%*
#3	mean	16.93	15.76	0.509	1019	1048	*2959*	*2959*	*2.195*	*504.9*
	st. dev.	0.040	0.024	0.023	0.510	0.452	*0.241*	*0.241*	*0.015*	*23.55*
	rel. std.	0.24%	0.15%	4.58%	0.05%	0.04%	*0.01%*	*0.01%*	*0.67%*	*4.66%*
#4	mean	16.68	15.93	0.350	1015	1047	*2960*	*2960*	*2.089*	*740.3*
	st. dev.	0.011	0.008	0.007	0.708	0.395	*0.589*	*0.589*	*0.008*	*14.70*
	rel. std.	0.07%	0.05%	1.93%	0.07%	0.04%	*0.02%*	*0.02%*	*0.38%*	*1.99%*
Bok et al.	mean	–	–	–	1000	1049	2963	2963	1.925	767.4
	st. dev.	–	–	–	13.65	11.05	6.406	6.577	0.094	20.99
	rel. std.	–	–	–	1.37%	1.05%	0.22%	0.22%	4.91%	2.74%

Table 9.3: Robustness of the estimated intrinsic camera parameters using a main lens focal length of $f_L = 16\,\mathrm{mm}$.

However, one can also see that the model proposed by Heinze et al. [2016] supplies an accuracy similar to that of the model defined by Johannsen et al. [2013], even though it contains only three rather than five parameters.

Figure 9.4b shows how the depth accuracy is improved when the effect of a plenoptic sensor tilted with respect to the main lens is modeled. However, this model seems to have less impact on the depth accuracy than the depth distortion model. The estimated tilting angle is only about 0.25°.

Figure 9.4c shows the depth accuracy using various main lens focal lengths. Here, the complete model as described in Section 6.2.1 was used. As expected, the depth accuracy of the camera significantly increases with increasing focal length. In Zeller et al. [2016b], it was shown that the improvement is proportional to f_L^2 as long as $z_C \gg f_L$ holds.

Calibration Robustness

Another important factor for a calibration approach is the robustness of the estimated model parameters. Here, our calibration approaches are compared to the one presented by Bok et al. [2017] using the second dataset presented in Section 9.1.2. The method of Bok et al. [2017] also defines the complete projection from object space to the recorded micro images on the sensor. While the overall projection from object space to the micro images is quite similar to the model presented by us, we mainly hope to prove that the robustness of the calibration significantly benefits from the 3D calibration target in comparison to the planar checkerboard used by Bok et al. [2017].

To evaluate the robustness of the calibration approaches, the calibration was performed for each method 10 times by randomly picking a set of 20 images out of the complete collection of 64 images. The means and the standard deviations of all intrinsic parameters are given for all methods in Table 9.3.

The parameters f_x, f_y, K_1 and K_2 need not be calculated in our calibration approaches. However, for comparison purposes these parameters were calculated (numbers in italic font) on the basis of our calibration results using the definitions given by Bok et al. [2017].

If one compares only the camera models proposed in this thesis, one can see that the virtual image based model (#1) has the most problems in reproducing the estimation of the distance B between sensor and MLA. This is very likely due to the fact that in this model the parameter B can only be estimated from the noisy virtual depth estimates, while the virtual image coordinates in x and y direction are basically independent of these parameters. The models based on the projection to the micro images are better suited to estimate these parameters, since each micro image point adds information. The parameter B is also less robust for model #3 in comparison to model #2. Due to the additional pixel distortion, there is more variability in the model, which makes the estimation of B less robust. This actually can be improved by adding more views, with more variations in depth, to the bundle adjustment. Table 9.1, for instance, shows that B was estimated quite reliably for all three lenses using model #3.

For the purpose of comparison, model #4 was added, in which the squinting micro lenses are considered, as in model #3 but not the pixel distortion on the sensor. As one can see from Table 9.3, for this model the parameters are estimated with a robustness similar to #2. However, it will be shown below why it still makes sense to choose model #3 over model #4.

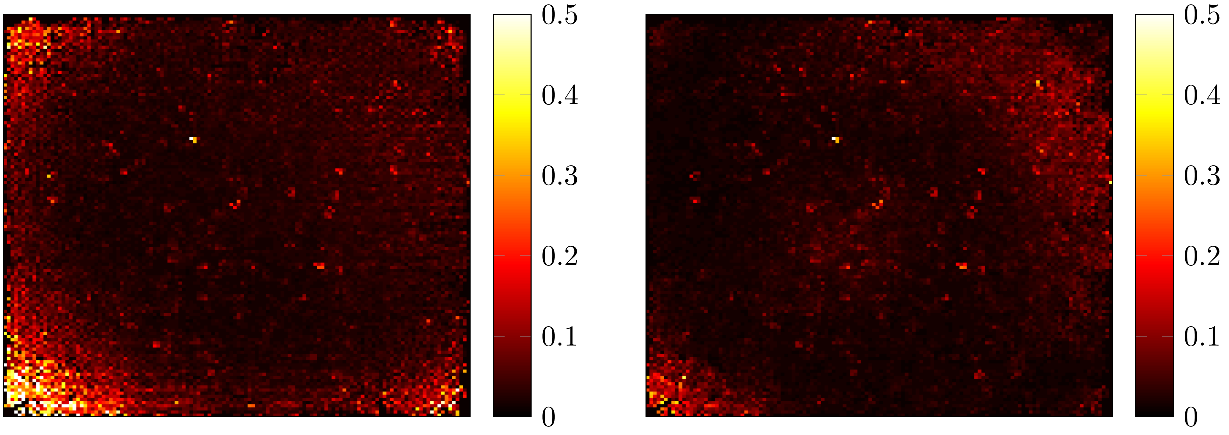
Furthermore, the results for the method of Bok et al. [2017] are shown, which also defines the projection on the micro images but performs calibration based on a planar checkerboard pattern. From Table 9.3, one obtains that all models proposed thesis, except for the parameter K_2, show parameter variations which are by an order of magnitude smaller than those from the methods of Bok et al. [2017]. A scattering similar to that of the parameter B is received for K_2, since both parameters are highly correlated (see Bok et al. [2017] for definition).

There is quite a large difference of the estimates in the intrinsic parameters f_x, f_y, K_1 and K_2, when comparing the methods under examination. Our method estimates f_x and f_y to have the same value, since the pixels are considered to be square. This is confirmed by the estimates of the method Bok et al. [2017]. There do not exist any absolute reference values and thus one cannot make any statement about the correctness of these values. All methods resulted in similar reprojection errors of less than 0.2 pixel which confirms the validity of all models.

Up to this point, none of the presented results explain why it is preferable to choose model #3 (with pixel distortion) instead of model #4 (without pixel distortion). The reason for choosing model #3 is shown in Figure 9.5. For the case that the distortion model is only applied to the micro image centers and not to the pixels in the respective micro images, the reprojection errors increase towards the image corners, as shown in Figure 9.5a. If additional pixel distortion is considered, the reprojection errors are more homogeneously distributed over the complete image, as shown in Figure 9.5b. However, Figure 9.5b still shows a slightly increasing error towards the bottom left and top right corner, which might be due to the slightly titled sensor, which was not considered in the model.

Effect of Squinting Micro Lenses

The previously presented calibration results did not point out the effect of correcting the micro lens centers with respect the micro image centers, which was called the squinting of the micro lenses. This effect can only be seen from the estimated extrinsic orientation of the plenoptic camera. In Figure 9.6 the relative translations from the extrinsic orientations received by model #2 (without squinting micro lenses) to the extrinsic orientations received by model #3 (with squinting micro lenses) are drawn. As one can see from the figure, along the x and y domain



(a) reprojection error map for model #4

(b) reprojection error map for model #3

Figure 9.5: Reprojection error map with and without the consideration of pixel distortion. The maps represent the distribution of the reprojection error over the complete image. (a) Reprojection error map for model #4 where no pixel distortion is considered. (b) Reprojection error map for model #3 where both micro lens center and pixel distortion are considered.

the estimated camera position is equivalent (with the exception of some noise) for both models. However, model #3 results in a camera closer to the object than model #2. This shift, of course, depends on the intrinsic camera parameters as they also define the amount of squinting.

Unfortunately, there is no possibility of measuring the correct position of the camera (which is, in fact, the position of the main lens' principal plane). However, the effect, that the measurements of the plenoptic camera have a constant negative offset with respect to the ground truth, has been observed by Johannsen et al. [2013] and Heinze et al. [2016]. Both measure the ground truth relative to the sensor plane.

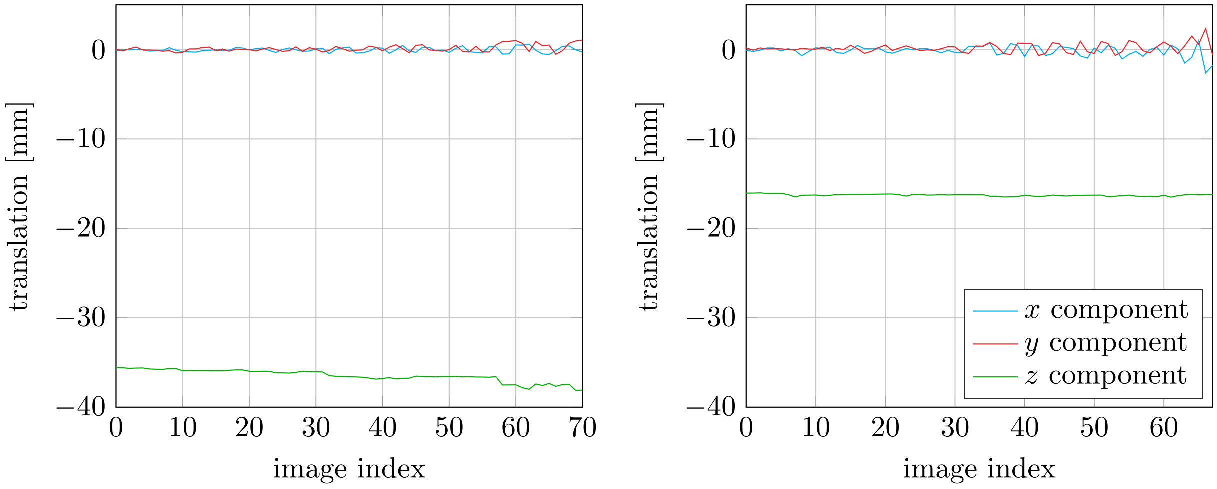
Heinze et al. [2016] explain this effect with the fact that they use a thin lens model rather than a thick lens. This explanation is inconsistent, as disregarding the extent of a real lens would result in measurements of the object distance with respect to the position of the sensor plane, which in turn are smaller than the real distances and not larger, as is, in fact, the case here.

As mentioned previously, disregarding the thickness of the real lens is not actually an issue with respect to camera tracking, as the object sided principal plane of the main lens will still be in the correct position.

9.3 Conclusion

In Chapter 6 a variety of plenoptic camera models and different calibration approaches was presented. Those were extensively evaluated in the previous sections.

The depth conversion functions presented in Section 6.1 can be used to convert the virtual depths estimated from the images of a focused plenoptic camera into metric object distances. The two model-based approaches proposed in this thesis are far more accurate and better suited for this purpose than the polynomial based function which was defined as reference. Thus, for the model-based approaches, only a small number of calibration points is needed and yet they are still valid beyond the range of calibration data. In combination with a standard pinhole camera model, these depth conversion functions may be sufficient for rough 3D measurement applications. However, the pinhole camera model only approximates the model of a focused plenoptic camera



(a) main lens $f_L = 35\,\text{mm}$

(b) main lens $f_L = 16\,\text{mm}$

Figure 9.6: Position offset between camera model #2 (without squinting micro lenses) and model #3 (with squinting micro lenses). (a) Relative offset on the $f_L = 35\,\text{mm}$ dataset. (b) Relative offset on the $f_L = 35\,\text{mm}$ dataset.

well for object distances which are significantly larger than the focal length. Furthermore, the introduced depth conversion functions do not consider any depth distortion, which limits the accuracy of the converted depth map.

For accurate 3D reconstruction purposes a complete intrinsic model of the plenoptic camera is needed. Essentially two different types of camera models were presented. In the first model the reconstructed virtual image is considered to be more or less the image of a RGB-D camera, which supplies a totally focused intensity image and a virtual depth map from the same perspective. The second type directly models the projection on the micro images. Both models supply accurate and plausible results. However, for the virtual image based model, in addition to lateral distortion, a depth distortion model has to be defined. For the micro images based model depth estimation can instead be performed based on already corrected micro images, which results in undistorted depth estimates. However, due to the reason that the image distortion is corrected beforehand, the regular grid of the MLA is lost.

A major advantage of the micro images based models is that they preserve the structure presented in Section 4.1. Therefore, the micro images based models enable plenoptic camera based multiple view geometry, as described in Section 4.2. To preserve the structure from Section 4.1, the fact that the plenoptic sensor might by tilted with respect to the main lens was neglected. However, the tilting angles estimated by model #1 are quite small. Furthermore, the error made in the model is compensated for quite well by the remaining parameters, as the reprojection error map in Figure 9.5b shows. Another advantage in comparison to the virtual image based models is that the micro images based models are estimated more robustly, as the results in Table 9.3 show.

In contrast to all previously published methods, the bundle adjustment based calibration approaches proposed in this thesis use a 3D target instead of a planar one. This, in turn, makes the proposed calibration approaches more robust than the previous methods.

Another crucial difference is that we were able to point out that the micro lens centers are, in fact, not equivalent to the micro image centers, which can be estimated from a white image, as has been done in most previous publications. This misalignment of the micro lens centers results in a bias in the camera position with respect to the real one. While this bias has also been observed in previous publications, it had not been correctly interpreted until now.

Finally, it was pointed out that for the purpose of plenoptic camera based VO model #3, which performs projection directly on the micro images while simultaneously considering the squinting micro lenses, is the most suitable model and was used for the experiments presented in Chapter 10.

10 Experiments – Full-Resolution Direct Plenoptic Odometry

10.1 Data

Currently, there are no public datasets available for plenoptic camera based visual odometry (VO). The few existing algorithms (Dansereau et al. [2011]; Dong et al. [2013]) were only tested on very simple and short sequences. Our goal is to evaluate the versatility of plenoptic VO algorithms and rank these algorithms with respect to methods based on other sensors (e.g. monocular or stereo cameras). Therefore, a new dataset was acquired by us to evaluate the DPO algorithms presented in this dissertation. There already exists a variety of different VO algorithms for monocular and stereo systems. Since all the systems are intended to perform robust tracking and mapping, we are of the opinion that plenoptic camera based algorithms also have to compete with those existing methods.

10.1.1 Data Acquisition Setup

To be able to compare plenoptic with stereo or monocular algorithms, a handheld platform was developed. On this platform a plenoptic camera and a stereo camera system are assembled. This platform is shown in Figure 10.1.

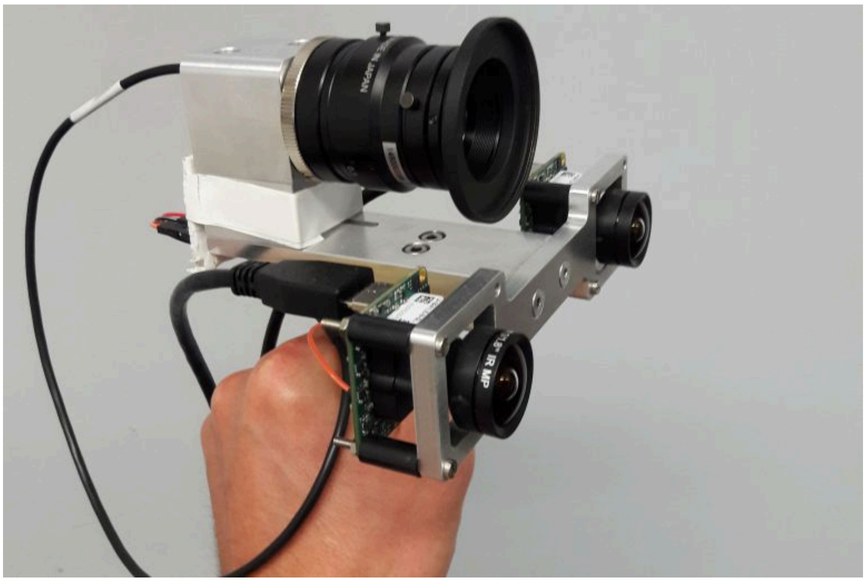


Figure 10.1: Handheld platform to acquire time synchronized image sequences from a focused plenoptic camera and a stereo camera system.

	plenoptic camera	stereo system
cameras	1	2
pixel size	5.5 µm	5.3 µm
resolution	2048 × 2048	1280 × 1024
color channels	3	1
focal length	16 mm	4 mm
aperture	f/2.8	f/1.8
stereo baseline	–	100 mm

Table 10.1: Camera specifications for the plenoptic camera and the stereo camera system.

The stereo camera system is based on two monochrome, global shutter, industrial cameras by IDS Imaging Development Systems GmbH (model: UI-3241LE-M-GL). On both cameras, a lens from Lensation GmbH (model: BM4018S118) with 4 mm focal length is mounted. Furthermore, the stereo system has a baseline distance of 100 mm.

The utilized plenoptic camera is a R5 by Raytrix GmbH. This camera is based on a xiQ sensor from Ximea GmbH (model: MG042CG-CM-TG). The xiQ is also a global shutter, industrial grade sensor. To achieve a suitable trade-off between a wide FOV and a high angular resolution of the captured light field, a main lens with focal length $f_L = 16\,\mathrm{mm}$ from Kowa (model: LM16HC) was mounted on the camera. Table 10.1 lists all important specifications for the plenoptic camera and the stereo camera system.

To receive synchronized image sequences from all three cameras, one camera of the stereo systems runs in master mode and generates a signal which in turn triggers the other two cameras (second camera of the stereo system and plenoptic camera) running in slave mode. To record the data, all three cameras are connected to a single laptop. Using this platform, one is able to record synchronized image sequences for all three cameras running at the maximum image resolution and 8 bit quantization with frame rates of more than 30 fps. Because of the small FOV, the images of the plenoptic camera in particular are affected by motion blur. To keep the motion blur in an acceptable range, for all cameras the exposure time is upper bounded at 8 ms. Below this boundary, the automatic exposure adjustment of the respective camera controls the exposure time[1].

10.1.2 A Synchronized Stereo and Plenoptic Visual Odometry Dataset

Based on the platform presented in Section 10.1.1, a synchronized dataset was recorded for the quantitative comparison of plenoptic and stereo VO systems. The inspiration for this dataset was taken from the Benchmark for monocular VO presented by Engel et al. [2016]. Especially for long, large scale trajectories, it is impossible to obtain reference measurements which are accurate enough to serve as ground truth. Hence, guided by the idea of Engel et al. [2016], all sequences in the dataset were performed as a single very large loop, for which beginning and end of the sequence capture the same scene.

For all recorded sequences geometric camera parameters and vignetting data for both the stereo camera system and the plenoptic camera are supplied. Furthermore, for each sequence, reference poses are obtained from the loop closure between beginning and end. This loop closure information can be used to measure the tracking accuracy of a certain algorithm.

---

[1] For the stereo cameras, automatic exposure adjustment is only performed for the master (left) camera, while the slave (right) camera adopts the master setting.

Geometric Camera Calibration

For the VO dataset the plenoptic camera was calibrated exactly as described in Section 6.3.3. Here, the most complete model, which considers micro lens center correction (squinting of micro lenses) as well as pixel distortion on the sensor, as described in Section 6.2.3, is applied. For the applied distortion model, two radial symmetric parameters (A_0, and A_1) and tangential distortions are considered. From the calibration is resulted that using only two radial symmetric distortion parameters is sufficient to accurately define the lens distortion.

While most monocular and stereo datasets perform camera calibration based on a planar calibration target, stereo camera calibration was performed based on the 3D target presented in Section 9.1.2 as well, due to the same reasons as for the plenoptic camera. For the two monocular cameras in the stereo setup, the pinhole camera model as given in eq. (10.1) is defined.

$$\eta \begin{bmatrix} x_I \\ y_I \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} \tag{10.1}$$

In contrast to the plenoptic camera model, different focal lengths (f_x, f_y) in x- and y-direction are taken into account. Thereby, the camera model us able to deal with rectangular, instead of squared, sensor pixels.

Lens distortion is applied on normalized image coordinates as given in eq. (10.2).

$$\boldsymbol{x}_{Id} = \begin{bmatrix} x_{Id} \\ y_{Id} \end{bmatrix} = \begin{bmatrix} f_x(x + \mathit{\Delta} x_{dist}) + c_x \\ f_y(y + \mathit{\Delta} y_{dist}) + c_y \end{bmatrix} \quad \text{with} \quad x = \frac{x_C}{z_C} \text{ and } y = \frac{y_C}{z_C} \tag{10.2}$$

Due to the much larger FOV, rather than two, one has to consider three parameters for radial symmetric distortion. Furthermore, the effect of tangential distortion is negligible.

$$\mathit{\Delta} x_{dist} = x(A_0 r^2 + A_1 r^4 + A_2 r^6) \tag{10.3}$$

$$\mathit{\Delta} y_{dist} = y(A_0 r^2 + A_1 r^4 + A_2 r^6) \tag{10.4}$$

The variable $r = \sqrt{x^2 + y^2}$ defines the distance to the principal point on the sensor in normalized image coordinates. In addition to the intrinsic parameters, the orientation of the slave (right) camera with respect to the master (left) camera is defined by a rigid body transformation $\boldsymbol{G}(\boldsymbol{\xi}_{MS}) \in \mathrm{SE}(3)$, which is represented by the respective tangent space element $\boldsymbol{\xi}_{MS} \in \mathfrak{se}(3)$.

As for the plenoptic camera model, the stereo camera model is estimated by performing a complete bundle adjustment using the 3D calibration target shown in Figure 9.1b.

Vignetting Correction

Especially for direct VO approaches, vignetting has a negative effect on the performance. The model parameters are estimated based on photometric measurements and therefore the measured intensity of a point must be independent of its location on the sensor. Indirect methods are more robust to vignetting, as extracted feature points generally rely on corners in the images, which are invariant to absolute intensity changes.

While in a monocular camera vignetting generally results in a continuously increasing attenuation of pixel intensities from the image center towards the boundaries, in a plenoptic camera further vignetting effects are present for each single micro lens.

Mathematically the vignetting can be described as follows:

$$I(\boldsymbol{x}) = \tau \cdot (V(\boldsymbol{x}) \cdot B(\boldsymbol{x}) + \epsilon_I) \,. \tag{10.5}$$

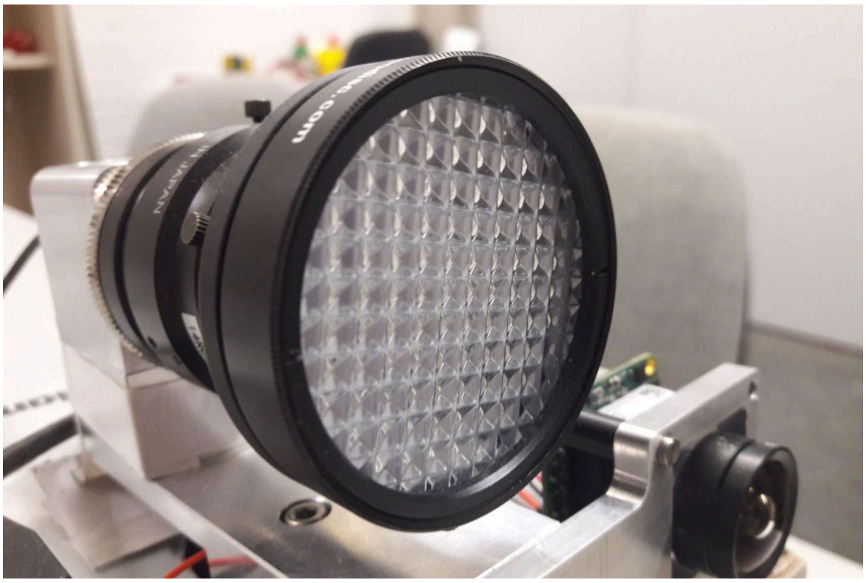


Figure 10.2: White balance filter used to record white images for the plenoptic camera and the stereo camera system. The white images are used for vignetting correction.

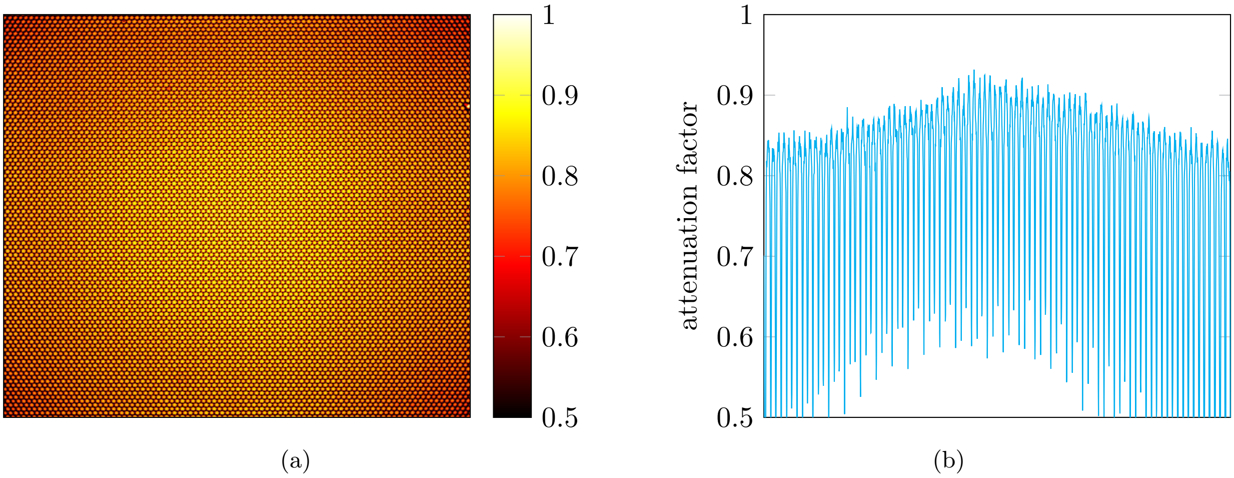
The function $I(\boldsymbol{x})$ is the observed intensity value measured by the sensor, while $B(\boldsymbol{x})$ is the irradiance image which represents the scene in a photometrically correct way. The vignetting $V(\boldsymbol{x})$ defines a pixel-wise attenuation, with $V(\boldsymbol{x}) \in [0,1]$. Furthermore, the variable τ denotes the exposure time while ϵ_I is the sensor noise. In this simplified model, the image sensor is considered to have a linear transfer characteristic which is, in fact, not the case for a real sensor.

While the nonparametric vignetting compensation based on white images recorded with a white balance filter is commonly applied to plenoptic cameras, the vignetting of the two cameras in the stereo system is corrected in the same way. For each camera a set of 10 white images was recorded and an average attenuation image was calculated from this set. Figure 10.2 shows the used white balance filter. For the two cameras of the stereo system, the attenuation images are additionally filtered using a Gaussian kernel. This cannot be done for the attenuation image of the plenoptic camera, as in this case, the MLA produces quite high frequent components in the attenuation image which must be preserved.

Engel et al. [2016] describe a different method, where the attenuation map is calculated based on a sequence of images capturing a white wall. However, the method of Engel et al. [2016] is quite time consuming and, in our experience, error prone. It was found that reflections and shadows on the white wall negatively affect the results. For the method of Engel et al. [2016], one has to capture a sequence of hundreds of images and then run the estimation for up to one hour. The attenuation map based on the white balance filter, by contrast, is obtained in just a few seconds. Furthermore, the goal was to apply comparable calibrations to both systems; the plenoptic camera as well as the stereo cameras.

Figure 10.3 shows the vignetting for the plenoptic camera. As one can see, vignetting is visible in each individual micro image. Furthermore, outer image regions are attenuated stronger than the image center. This is due to the influence of the main lens. In addition, there are some small irregularities visible in the map resulting from defect micro lenses and dirt on the MLA.

Figure 10.4 shows the attenuation maps estimated for the left camera of the stereo system. Figure 10.4a shows the result using the white balance filter, while Figure 10.4b shows the one obtained from the method of Engel et al. [2016]. From Figure 10.4c one can clearly see that for



(a)

(b)

Figure 10.3: Estimated attenuation image for the plenoptic camera. (a) Complete attenuation image. (b) Horizontal cross section through the attenuation map. Vignetting is visible in each micro image as well as across the complete image resulting from the main lens.

the white balance filter the attenuation is sightly stronger at the sensor boundaries than for the method of Engel et al. [2016]. However, the deviation is quite small and furthermore, it is difficult to evaluate which map describes the vignetting of the camera in a more accurate way.

Ground Truth and Evaluation Metric

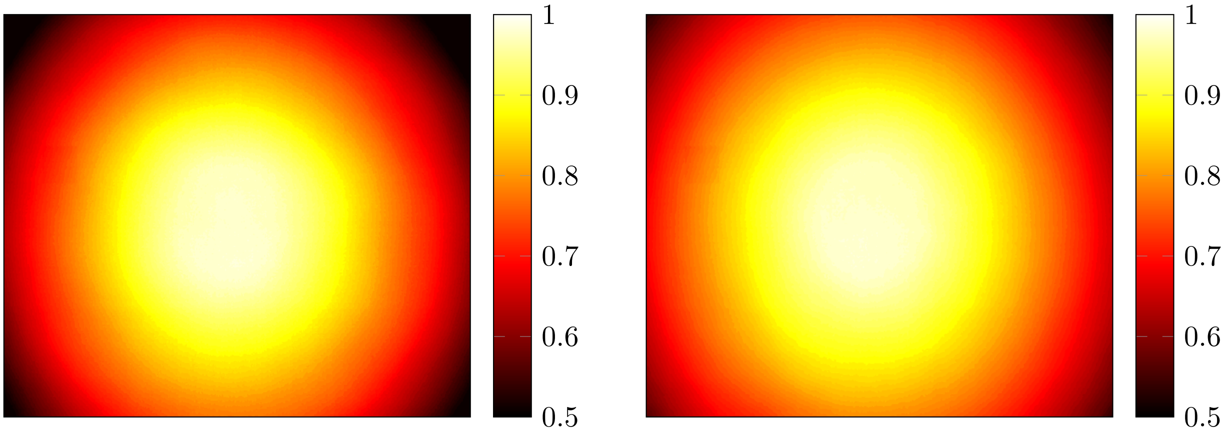
As already stated before, it is almost impossible to obtain ground truth trajectories for long and large-scale sequences recorded by handheld cameras. We decided to obtain reference measurements for the dataset in a similar way as suggested by Engel et al. [2016], where the accuracy of a VO algorithm is evaluated based on a single, larger loop closure. Each trajectory in the dataset starts with a winding sequence while capturing a nearby object. This starting sequence is followed by the actual trajectory which finally leads back to the starting point in a large loop, followed by a short, winding, finishing sequence.

Using the winding sequence at the beginning and at the end, both segments can be aligned to each other using a standard SfM approach. These aligned segments then can be used as ground truth information. In contrast to monocular datasets, also reference data for the absolute scale of the trajectory is needed. The absolute scale of the trajectory is obtained from stereo images. Since the stereo cameras have a much larger stereo baseline than the micro images in the plenoptic camera, the observed scale is accurate enough to serve as reference for the plenoptic sequences.

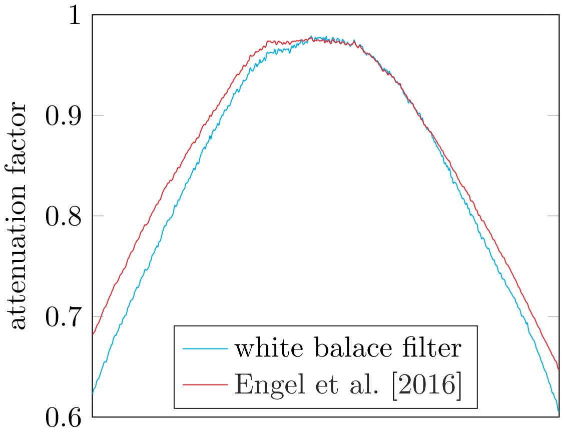
While one can basically use any SfM algorithm to align the start and end segment to each other, here a modified version of ORB-SLAM2 (stereo) was applied. Instead of selecting keyframes, a frame-wise pose graph was build up which is then optimized in a global bundle adjustment.

Figure 10.5 shows the aligned segments for the beginning and the end of the sequence in blue and red respectively, overlaid with the point cloud calculated from DPO.

Using the aligned reference poses, based on each recorded trajectory two similarity transformations $\boldsymbol{T}_s^{\text{gt}}$ and $\boldsymbol{T}_e^{\text{gt}}$ ($\in \mathrm{Sim}(3)$) with respect to the start and the end segment of the sequence



(a) white balance filter (b) Engel et al. [2016]



(c) cross section through attenuation images

Figure 10.4: Estimated attenuation images for the left camera of the stereo camera system. (a) Attenuation image recorded with the white balance filter. (b) Attenuation image calculated based on the method described by Engel et al. [2016]. (c) Horizontal cross section through the resulting attenuation images.

can be calculated, as given in eqs. (10.6) and (10.7).

$$\boldsymbol{T}_s^{\mathrm{gt}} := \underset{\boldsymbol{T} \in \mathrm{Sim}(3)}{\arg\min} \sum_{i \in S} (\boldsymbol{T}\boldsymbol{p}_i - \boldsymbol{p}_i^{\mathrm{gt}})^2 \tag{10.6}$$

$$\boldsymbol{T}_e^{\mathrm{gt}} := \underset{\boldsymbol{T} \in \mathrm{Sim}(3)}{\arg\min} \sum_{i \in E} (\boldsymbol{T}\boldsymbol{p}_i - \boldsymbol{p}_i^{\mathrm{gt}})^2 \tag{10.7}$$

The vectors $\boldsymbol{p}_i \in \mathbb{R}^3$ are the estimated points of the trajectory while $\boldsymbol{p}_i^{\mathrm{gt}} \in \mathbb{R}^3$ are the respective points of the ground truth. S and E define the sets of indices of the start end and segment respectively.

Using the similarity transformations $\boldsymbol{T}_s^{\mathrm{gt}}$ and $\boldsymbol{T}_e^{\mathrm{gt}}$ evaluation metrics are defined similar to that of Engel et al. [2016]. From the two transformations the accumulated drift $\boldsymbol{T}_{\mathrm{drift}} \in \mathrm{Sim}(3)$ from the start to the end of the trajectory can be calculated as follows:

$$\boldsymbol{T}_{\mathrm{drift}} := \begin{bmatrix} e_s\boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} = \boldsymbol{T}_e^{\mathrm{gt}} (\boldsymbol{T}_s^{\mathrm{gt}})^{-1} = \begin{bmatrix} s_e\boldsymbol{R}_e & \boldsymbol{t}_e \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} s_s\boldsymbol{R}_s & \boldsymbol{t}_s \\ \boldsymbol{0} & 1 \end{bmatrix}^{-1}. \tag{10.8}$$

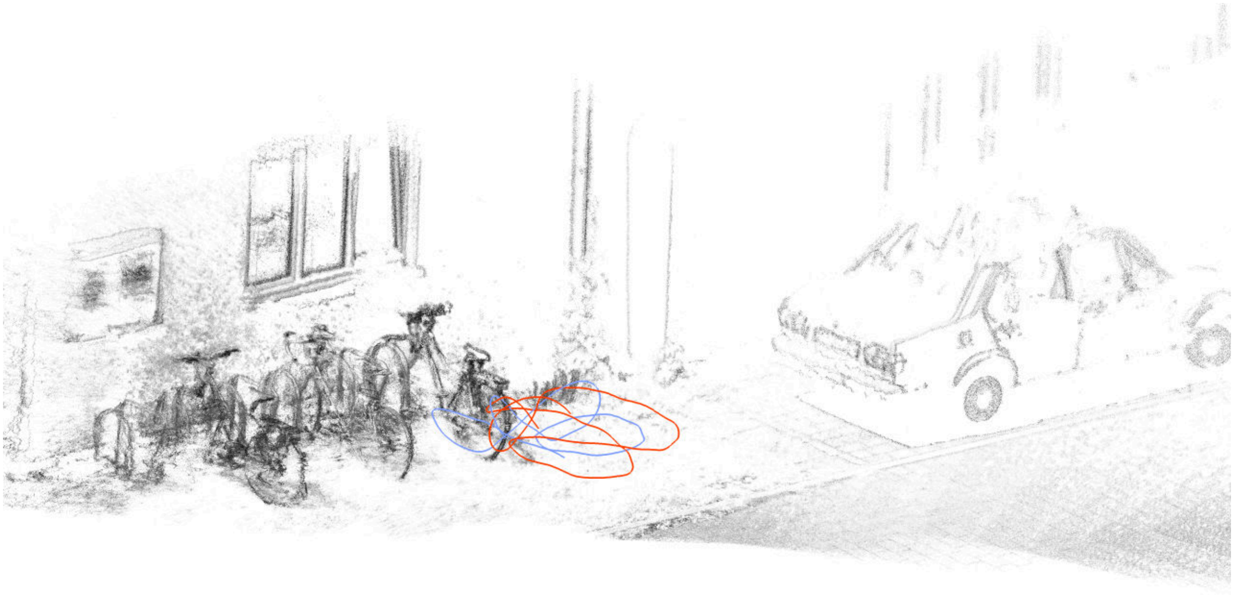


Figure 10.5: Semi-dense point cloud generated by DPO overlaid with the loop closure trajectory of the start and end segment, used as ground truth, and obtained from SfM on the stereo images. Start segment of the trajectory is marked in red and the end segment in blue.

From $\boldsymbol{T}_{\text{drift}}$ one can directly extract the scale drift e_s, the rotational drift e_r, and the translation drift $e_t := \|\boldsymbol{t}\|$. The rotational drift e_r is defined by the rotation angle around the Euler axis corresponding to the rotation matrix $\boldsymbol{R}$:

$$e_r := \|\boldsymbol{w}\| \cdot \frac{180^\circ}{\pi} \qquad \text{with } \boldsymbol{w} = \log_{\mathrm{SO}(3)}(\boldsymbol{R}). \tag{10.9}$$

For an easier interpretation of the scale drift $e'_s := \max\{e_s, e_s^{-1}\}$ is defined.

The drift metrics e'_s, e_r, and e_t define more or less independent quality measures. Looking at just one of these values offers only a very limited insight into the overall quality of the estimated trajectory. Furthermore, e_t as it is defined here, is proportional to the absolute scale of the estimated trajectory and therefore is not meaningful at all without considering the absolute scale. Engel et al. [2016] already defined the alignment error e_{align} as a more meaningful combined metric:

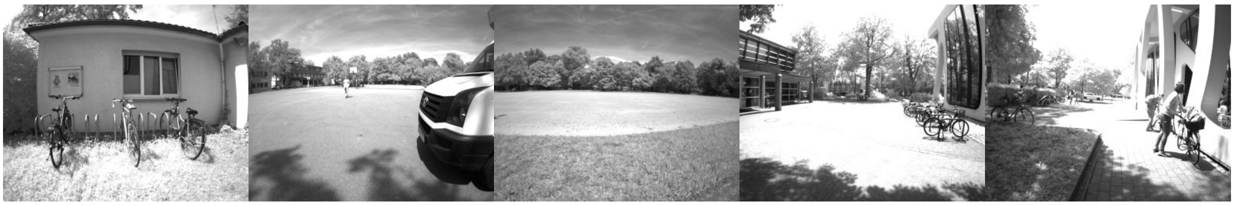
$$e_{\text{align}} := \sqrt{\frac{1}{N}\sum_{i=1}^{N} \left\|\boldsymbol{T}_s^{\text{gt}}\boldsymbol{p}_i - \boldsymbol{T}_e^{\text{gt}}\boldsymbol{p}_i\right\|_2^2}. \tag{10.10}$$

The parameter N is the number of points (frames) in the complete trajectory. In comparison to e_t, e_{align} is always scaled with respect to the ground truth and implicitly incorporates all drifts e'_s, e_r, e_t in a single number.

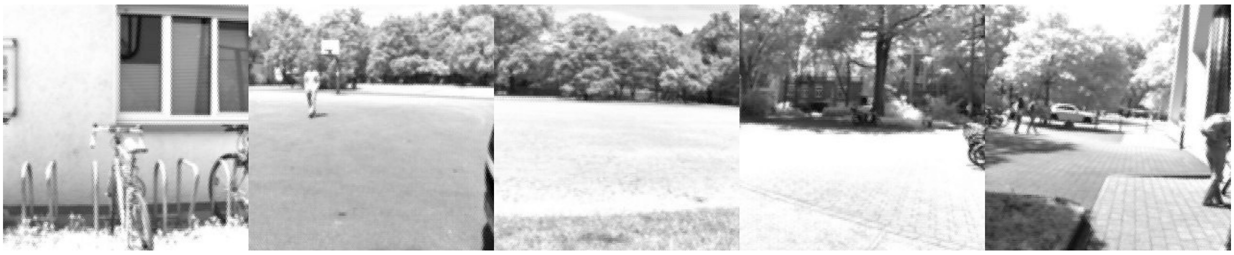
All these metrics consider only the relative drift from the beginning to the end of the trajectory, but not the error of the absolute scale. Hence, the absolute scale difference d_s of the front and end segment is defined as another metric:

$$d_s := \sqrt{\operatorname{scale}(\boldsymbol{T}_e^{\text{gt}}\boldsymbol{T}_s^{\text{gt}})} = \sqrt{s_e \cdot s_s}. \tag{10.11}$$

Similar to the scale drift e'_s, the metric $d'_s := \max\{d_s, d_s^{-1}\}$ is defined.



(a) sample images from the sequence of the left camera in the stereo camera system



(b) sample images from the sequence of the plenoptic camera

Figure 10.6: Sample images for one sequence of the synchronized stereo and plenoptic VO dataset. (a) Sample image from the left camera of the stereo camera system. (b) Sample images from the plenoptic camera. The images of both cameras correspond to exactly the same point in time.

Of course, d_s must be considered only for plenoptic and stereo algorithms and not for monocular approaches. Furthermore, d_s has significance only in combination with the scale drift e'_s:

$$s_{\max} = d_s \cdot \sqrt{e'_s}, \tag{10.12}$$

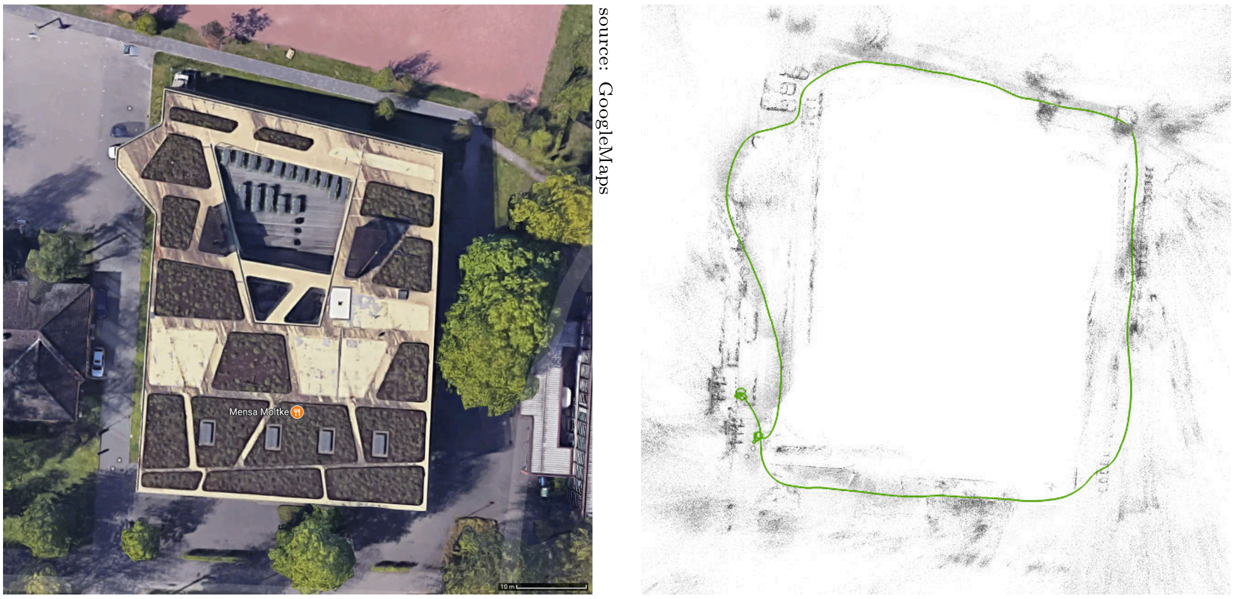
$$s_{\min} = \frac{d_s}{\sqrt{e'_s}}. \tag{10.13}$$

To obtain reliable ground truth data, all sequences start and end in a scene showing objects in a distance of several meters, which are easy to track. However, for these nearby objects it is easier to estimate the correct scale. To consider only the scale drift e'_s or the absolute scale d_s might be misleading. In combination with the alignment error e_{align}, these values become more meaningful.

Following the scheme described above, a set of 11 sequences in versatile environments was recorded. The recorded scenes range from large scales to small scales, from man-made environments to environments with abundant vegetation. The sequences capture moving objects like pedestrians, bikes or cars. The sequences also cover difficult and changing lighting conditions due to shadows, moving clouds, and automatic exposure adjustment.

Figure 10.6 shows a set of sample images extracted from a single sequence. The corresponding trajectory is shown in Figure 10.7, while Figure 10.5 visualizes the registered start and end segments to calculate the metrics. As one can see, the monocular images of the stereo system (Fig. 10.6a) have a much wider FOV than the images of the plenoptic camera (Fig. 10.6b). In Figure 10.6b, the images of the plenoptic camera seem to be a bit blurred. This is not, in fact, the case and is only due to the multiple projections of a point in neighboring micro images. Due to the narrower FOV and the higher number of pixels on the sensor, images of the plenoptic camera actually have a much higher spatial resolution than those from the monocular cameras.

Figure 10.8 shows a magnified subsection of the trajectory and the point cloud of Figure 10.7b. This subsection shows the beginning and end of the sequence. One can clearly see the drift accumulated over the complete sequence, resulting in the same scene being reconstructed twice in slightly different locations.



source: GoogleMaps

(a) top view real scene

(b) estimated trajectory and 3D point cloud

Figure 10.7: Example sequence of the dataset used for evaluation. (a) Top view of the real scene. (b) Trajectory (green) and 3D point cloud estimated by DPO.

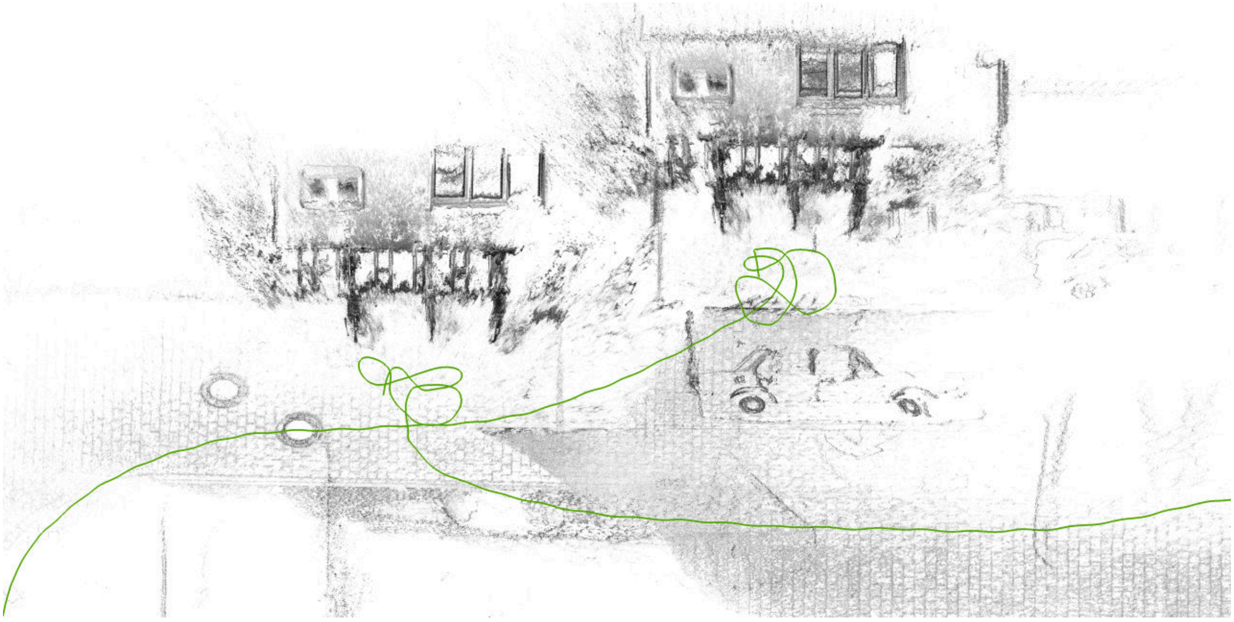


Figure 10.8: Example of the drift accumulated by DPO from the beginning to the end of a sequence. Due to the drift in the trajectory, the same scene is reconstructed twice at different locations. The green line represents the camera trajectory estimated by the algorithm.

Known Limitations

While the two monocular cameras of the stereo system both have a monochromatic sensor, the plenoptic camera has a RGB sensor. Even though all image sensors have a similar pixel size, a pixel of the plenoptic camera captures only approximately a third of the light energy compared to a pixel of the monocular cameras[2], when we, in fact, assume that all other parameters are similar. For the plenoptic camera the F-number is predefined by construction, due to the aperture of the micro lenses. This F-number is higher than the one of the lenses used for the two monocular cameras. For this reason, the monocular cameras gather even more light energy on the same sensor area compared to the plenoptic camera. To compensate for these two issues, a hardware-sided amplification of 6 dB was set for the plenoptic camera. Thus, for all cameras the exposure times are within the same order of magnitude, although they do not match exactly. Due to the amplification, the images of plenoptic cameras will contain more noise when compared with the monocular cameras of the stereo system.

For the stereo camera system, it is important that both cameras run synchronized and with the same exposure time. For this reason, the automatically calculated exposure time of the master camera must be used to set the exposure time of the slave camera. It can happen that if the exposure time changes, an image pair is captured for which the two cameras had slightly different exposure times.

Currently, there exists no plenoptic camera based VO algorithm which performs loop closures. Therefore, the loop closure ground truth, calculated on the basis of the stereo images, is also used as ground truth for the plenoptic camera. With respect to the plenoptic camera, the ground truth might be slightly inaccurate due to the slightly different positions of the master camera of the stereo system and the plenoptic camera. Furthermore, there is a shift of few milliseconds between the recording times of the plenoptic and stereo cameras which, again, leads to slight inaccuracies. The superior way would be to calculate separate reference poses on the basis of the plenoptic images – which is currently not possible because no appropriate algorithm exists.

Due to the reason that different sensors and lenses which have different properties are used, and the fact that the cameras see the scene from sightly different perspectives, one has to keep in mind that even though one is able to perform quantitative evaluations based on the presented dataset, these quantities are only valid up to a certain degree. However, the dataset helps to emphasize the strength of VO based on a certain sensor with respect to the other sensors.

10.2 Results

This section presents the results which were obtained based on the dataset introduced in the previous section. First quantitative results are presented which are measured with respect to the ground truth. Both the light field and stereo sequences are used to evaluate on one side the proposed DPO algorithm and on the other side to compare it to a variety of state-of-the-art VO algorithms. In the second part of this section the results of the presented DPO algorithm are investigated from a qualitative perspective.

---

[2] For indoor sequences the light energy gathered by the plenoptic camera is even less than a third, since the ambient light here contains almost no infrared components and thus, red pixel are totally underexposed.

10.2.1 Quantitative Results

The DPO algorithm is compared to several other VO algorithms which are either based on monocular, stereo or RGB-D data[3]. These algorithms are:

- monocular:
  - LSD-SLAM (Engel et al. [2014])
  - DSO (Engel et al. [2018])
  - ORB-SLAM2 (Mur-Artal et al. [2015]; Mur-Artal and Tardós [2017])
- stereo:
  - ORB-SLAM2 (Mur-Artal and Tardós [2017])
- RGB-D:
  - RGBD-VO[4]

For none of the algorithms real time processing was enforced. For the algorithms which include a full SLAM framework (ORB-SLAM2 and LSD-SLAM), large scale loop closure detection and relocalization was disabled. The implementations of Direct Sparse Odometry (DSO) and LSD-SLAM are not able to handle the high image resolution of 1.3 megapixel of the monocular images. Thus, for DSO, LSD-SLAM and RGBD-VO the image resolution is reduced to 960 pixel × 720 pixel. Both versions of ORB-SLAM2 run at the full image resolution of 1280 pixel × 1024 pixel.

In Section 7.7 a global scale optimization framework was presented which enhances the scale awareness of DPO. The scale optimization can either be performed online or offline. Thus, for the DPO algorithm each sequence was evaluated in three different settings:

1. DPO: algorithm is running without scale estimation and scale optimization
2. DPO (offline): scale optimization is performed only after the complete trajectory is finished
3. DPO (online): scale is optimized and updated online when a new keyframe is created

In this section only a subsection of meaningful results will be presented. All calculated evaluation metrics for all sequences can be found in Table B.1 in Appendix B.3. Table B.2 contains the lengths of all trajectories.

Depending on the implementation a VO algorithm either signals a tracking failure or results in an abnormally high tracking error. Therefore, in the following Figures 10.9 and 10.11 appropriate graph limits were chosen. All values at the upper graph border signify that the respective algorithm either failed, and therefore no metric could be measured, or that the measured metric exceeds the upper graph limit.

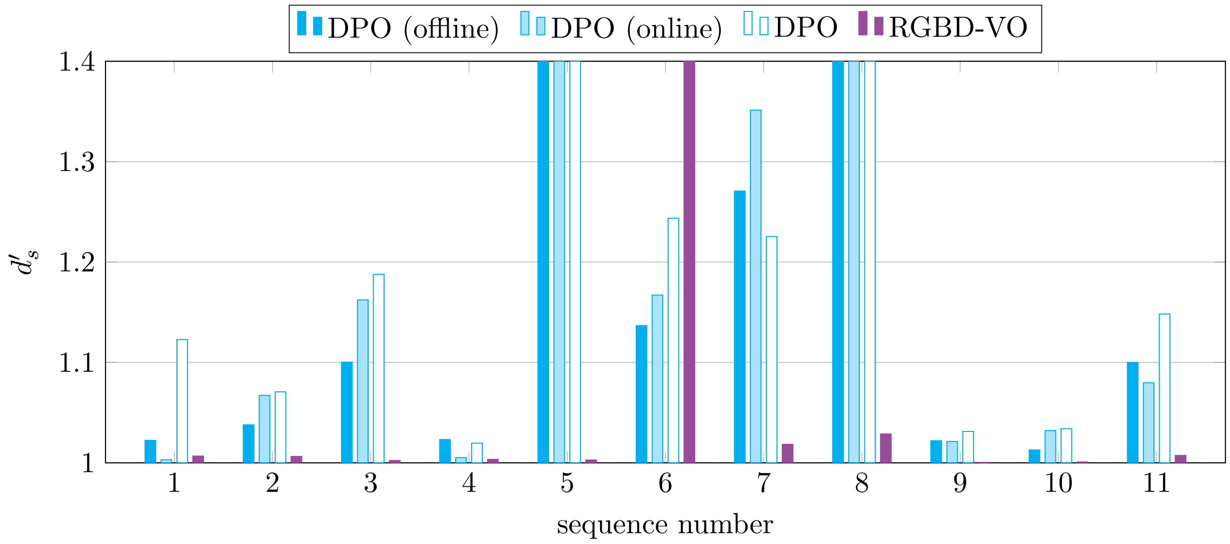
Scale Awareness and Scale Drift

Figure 10.9 shows the results for the estimated absolute scale and the scale drift for a selection of algorithms based on all sequences in the dataset. In Figure 10.9a, the estimated absolute scale

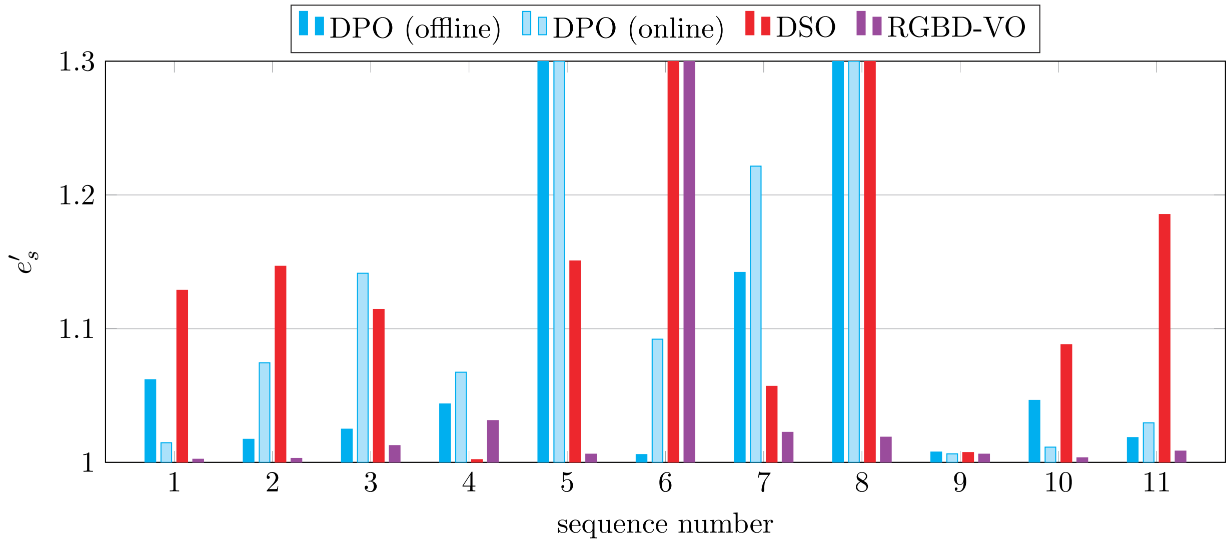
---

[3] RGB-D data is obtained from the stereo images using Semi-Global Matching (SGM) (Hirschmüller [2008]).

[4] RGBD-VO is a VO formulation based on RGB-D data. The algorithm performs keyframe based tracking using a modified version of the LSD-SLAM framework without pose graph optimization. The principle of this method is similar to Kerl et al. [2013b] and Klose et al. [2013].



(a) absolute scale error



(b) scale drift

Figure 10.9: Measured absolute scale errors d'_s and scale drifts e'_s. (a) Absolute scale measured for different settings of DPO and RGBD-VO. (b) Scale drift from the beginning to the end of a sequence measured for different settings of DPO, DSO and RGBD-VO.

is drawn for all three settings of DPO as well as for the RGBD-VO. No other algorithms are plotted here since the monocular approaches are not able to observe the absolute scale, while ORB-SLAM2 (stereo) essentially uses the same data from which the ground truth was obtained and therefore will have a similar scale accuracy as the ground truth.

Figure 10.9a shows that DPO without performing any scale optimization is able to observe the absolute scale with an accuracy of 10–20 %. By introducing the scale optimization, in both the offline and online mode a scale uncertainty of less than 10 % is obtained for most of the sequences. It seems that the results in the offline mode are slightly better than in the online mode. The reason for this might be that the scale estimation is not completely invariant of the current scale of a keyframe. However, running DPO in the offline mode is not necessarily a disadvantage. The correct scale still can be calculated online and merely the new keyframe will not be updated (see update scale path to new keyframe in Figure 7.4).

Of course, RGBD-VO shows a better absolute scale accuracy than DPO since the approach takes advantage of the large baseline of the stereo camera system. However, for sequence #4, where the object distances are in the range of only a few meters, DPO (online) estimated an absolute scale similar to RGBD-VO. For sequence #1, DPO (online) again shows an absolute scale error similar to that of RGBD-VO, even though this sequence consists of larger object distances.

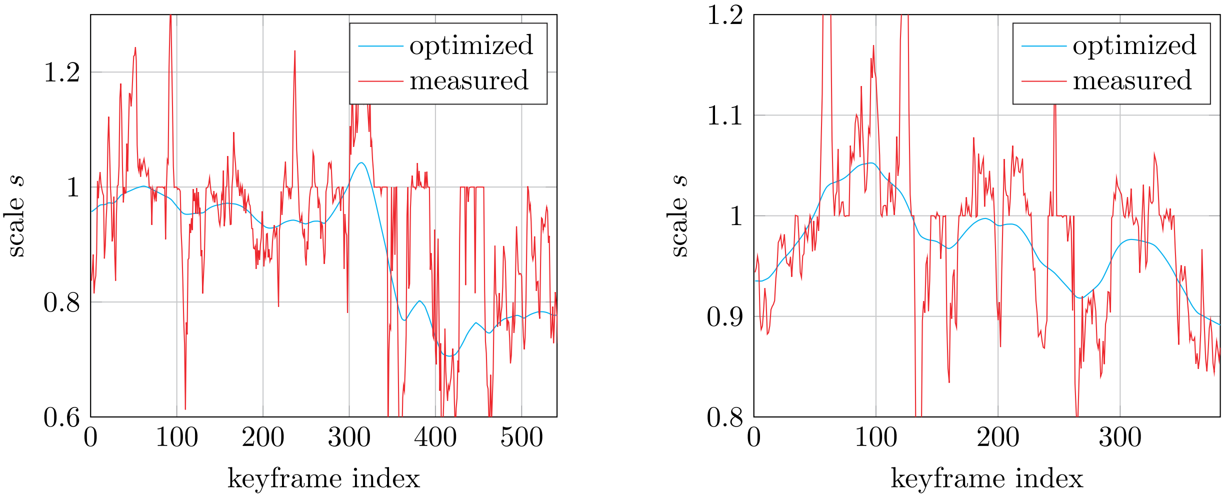
The anomalous results for sequence #7 can be explained, as this is an indoor sequence with a narrow staircase showing mostly white walls. Here, especially tracking on the plenoptic images is difficult due to the narrow FOV. Some of the images in the sequence contain only little amounts of texture which makes tracking very difficult. In these regions of poor tracking conditions, the assumption that scales of subsequent keyframes are highly correlated is violated, which results in a suboptimal performance of the scale optimization.

Figure 10.9b shows the scale drift for all sequences for different algorithms. From the figure one can see that DPO in the offline mode outperforms DSO, which represents the state-of-the-art in monocular VO, in most of the sequences. Furthermore, due to the possibility that the scale can be estimated from the stereo baselines between micro lenses, it can be expected that for long sequences, the scale drift of DPO is bounded, while a scale drift which happens in a monocular approach cannot be compensated and in the limit case will strive to infinity.

In Figure 10.10, by way of example, the measured and optimized scale is shown as a function of the keyframe index calculated by DPO in the offline mode. As one can see, the measured scale contains a high amount of noise. Our optimization framework drastically filters out that noise, resulting in a smooth scale estimate along the sequence. Even though the optimized absolute scale shows a high variation, the overall scale drift is approximately an order of magnitude smaller (see Figure 10.9b). The scale drift of DPO without scale optimization is 18 % for sequence #1 and 5 % for sequence #6. This conforms quite well to the result of the scale optimization when the first scale in the graphs is compared to the last one.

Rotational Drift and Alignment Error

The results presented in the previous section confirmed what was expected from a plenoptic camera based VO algorithm. DPO forms a scale aware VO framework which mostly outperforms monocular approaches with respect to scale drifts. However, one challenge of DPO is the narrow FOV which was chosen for the plenoptic camera. Therefore tracking becomes much more difficult and due to little perspective distortion in the image, ambiguities between rotation and translation have to be expected.



(a) sequence #1 (b) sequence #6

Figure 10.10: Measured and optimized scale for DPO as a function of the keyframe index.

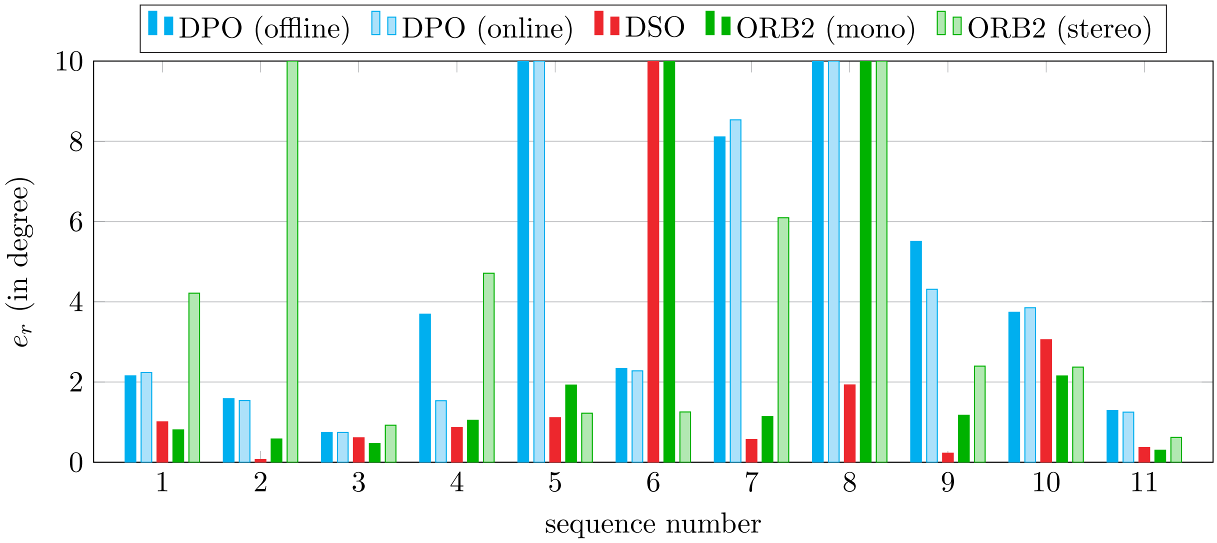
Figure 10.11 shows the rotational drifts (Figure 10.11a) and the alignment error (Figure 10.11b) for different algorithms. For a better comparison of the different sequences, the alignment error is plotted as percentages of the respective sequence length. From the presented algorithms it cannot be pointed out one method that performs best for all the sequences. However, the monocular DSO has an accuracy similar to ORB-SLAM2 (stereo) and even performs better on some sequences. With respect to the alignment error, DPO shows results better than ORB-SLAM2 (mono) and is competitive to DSO and ORB-SLAM2 (stereo) for most of the sequences.

LSD-SLAM is the algorithm most closely related to DPO. Thus it is interesting to note that DPO significantly outperforms LSD-SLAM (see Table B.1 in Appendix B.3).

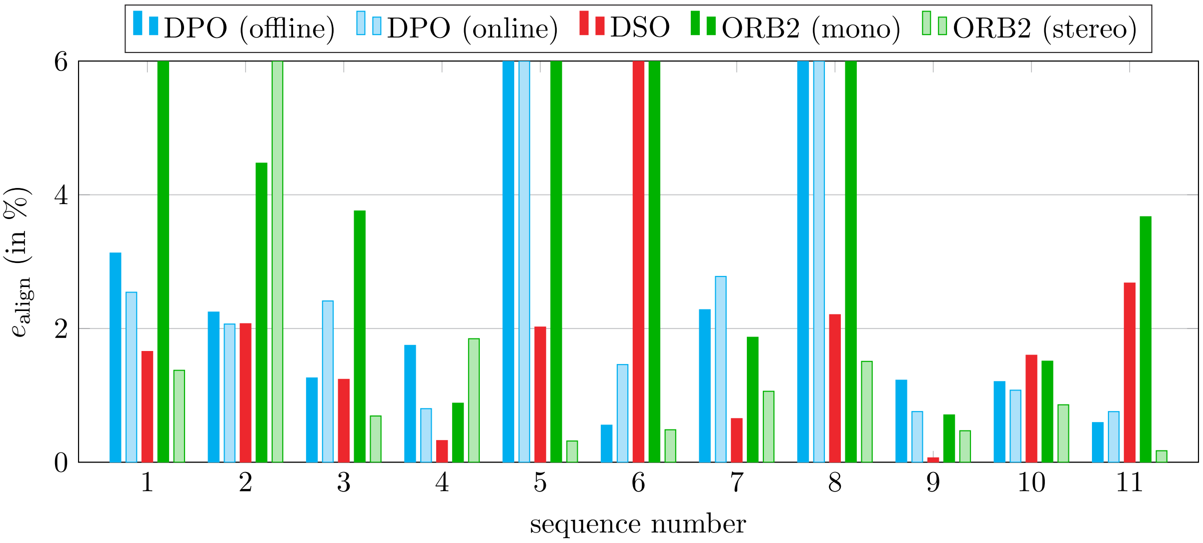
The dataset contains two sequences for which DPO was not able to track the complete sequence. For sequence #5, the tracking failed due to a van moving through the scene. For a short period of time large parts of the image are covered by the moving van which causes the algorithm to fail. Figure 10.12 shows some sample images of the respective part in the plenoptic sequence and the images for the same points in time of the left camera in the stereo system. Here, the monocular and stereo approaches benefit from the wider FOV. The second failure was due to badly exposed images in a dark sequence (sequence #8).

At this point it again has to be emphasized that this quantitative comparison of algorithms, which are based on different sensor systems, offers no absolute ranking of the algorithms among each other. However, it states the benefits of plenoptic camera based VO in comparison to traditional methods. All results are based on a single run of each algorithm for every sequence. Hence, no conclusion about the statistical behavior of the algorithms can be made.

To give an impression of the kind of sequences that were recorded, Appendix B.4 presents top views of the trajectories and point clouds calculated by DPO, for all sequences for which the algorithm succeeded.

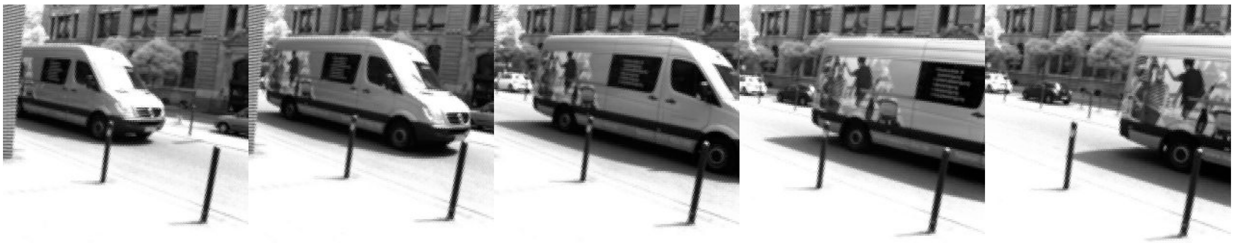


(a) rotational drift

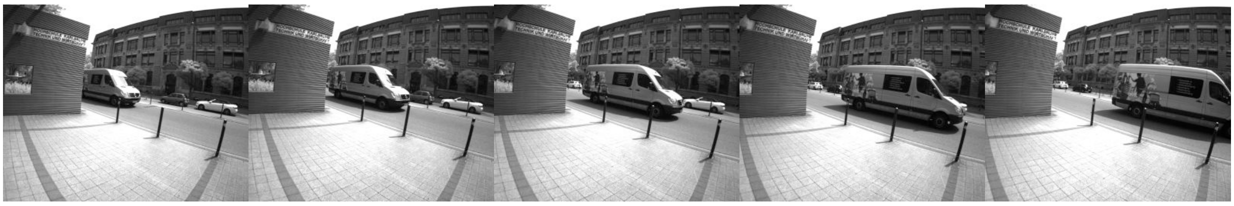


(b) alignment error

Figure 10.11: Measured rotational drift and alignment error. (a) Rotational drift measured for various settings of DPO, DSO and the monocular and stereo version of ORB-SLAM2. (b) Alignment error measured for different settings of DPO, DSO and the monocular and stereo version of ORB-SLAM2. The alignment error is given as percentages of the respective sequence length.



(a) sequence recorded by the plenoptic camera



(b) sequence recorded by the left camera of the stereo system

Figure 10.12: Sample images of sequence #5 recorded by the plenoptic camera and the left stereo camera. The images show the sequences at a fifth of the frame rate. Tracking of DPO fails for this sequence due to a van driving through the scene.

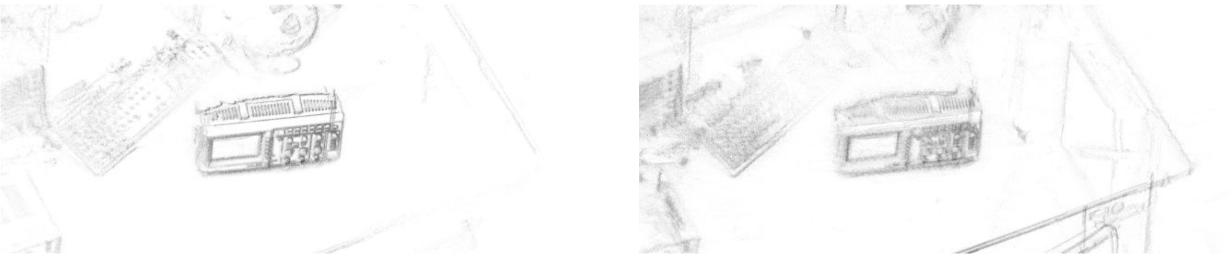
10.2.2 Qualitative Results

The previous section compared DPO quantitatively to existing state-of-the-art VO algorithms. In this section, the algorithms are supposed to be rated on a qualitative basis, and particularly based on the calculated point clouds. For all DPO point clouds presented in this section, the visualizations show the point clouds only at a point density of 20 %. This setting was chosen to be able to better handle the large amount of data in the point cloud viewer.

Because of the narrow FOV, the plenoptic camera offers a much smaller ground sampling distance than the monocular cameras in the stereo system. This is shown in Figure 10.13. While Figure 10.13a shows the point cloud generated by DPO, Figure 10.13b shows the point cloud of the same scene generated by LSD-SLAM. Furthermore, one must add that Figure 10.13a shows only 20 % of the points of the actual point cloud, while Figure 10.13b shows the complete set of points. The point cloud calculated by DPO is far more detailed than the one received from LSD-SLAM. At the same time one can see, for instance, from the edge of the desk that LSD-SLAM results in a more complete point cloud. This is due to the wider FOV of the monocular camera, when compared to the plenoptic camera. For comparison, Figure 10.14 shows the totally focused image (Figure 10.14a) calculated by DPO and the monocular image (Figure 10.13b) from the same point in time. Figure 10.14c shows a magnified subsection of the monocular image, showing the portion of the scene similar to the totally focused image of the plenoptic camera, though at a much lower resolution.

For each 3D point in the point cloud estimated by DPO, a corresponding inverse effective depth variance is available. Figure 10.15 shows how noisy points can be removed from the point cloud based on a threshold for the variance. While the high threshold results in a point cloud with high point density, but also a significant amount of noise, the low threshold removes noisy points and consequently results in a much lower point density.

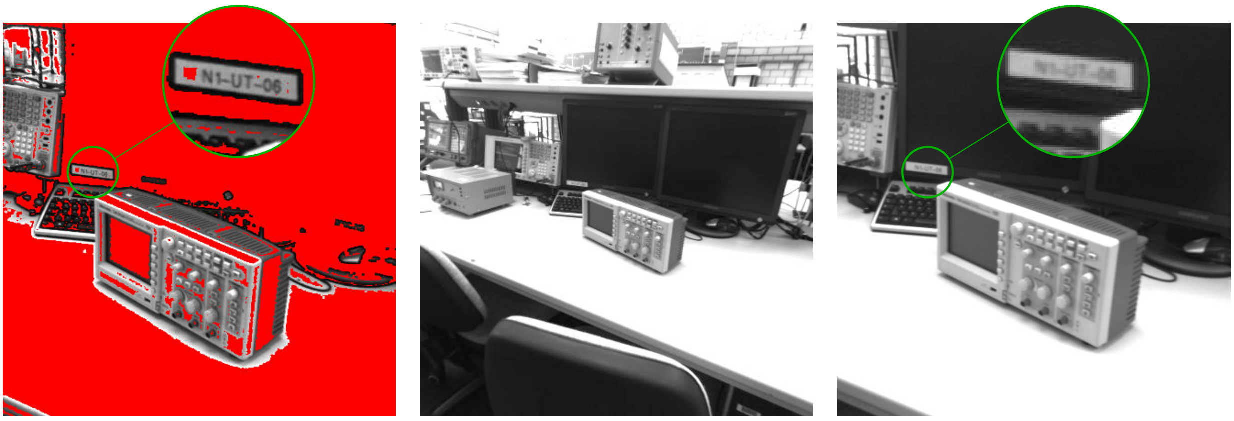
Figure 10.16 shows the point could and the trajectory calculated by DPO for a sequence with abundant vegetation, where the camera was moved through bushes. Even though the point cloud looks quite random, the camera trajectory was estimated accurately. Furthermore, from Figure 10.16 one can clearly see the nature of point clouds received in VO and SLAM algorithms.



(a) DPO

(b) LSD-SLAM

Figure 10.13: Comparison of point clouds received from DPO and LSD-SLAM (sequence #4). (a) Point cloud calculated by DPO visualized with a point density of 20 % compared to the actual point cloud. (b) Point cloud calculated by LSD-SLAM at full point density.



(a) totally focused imge

(b) monocular image

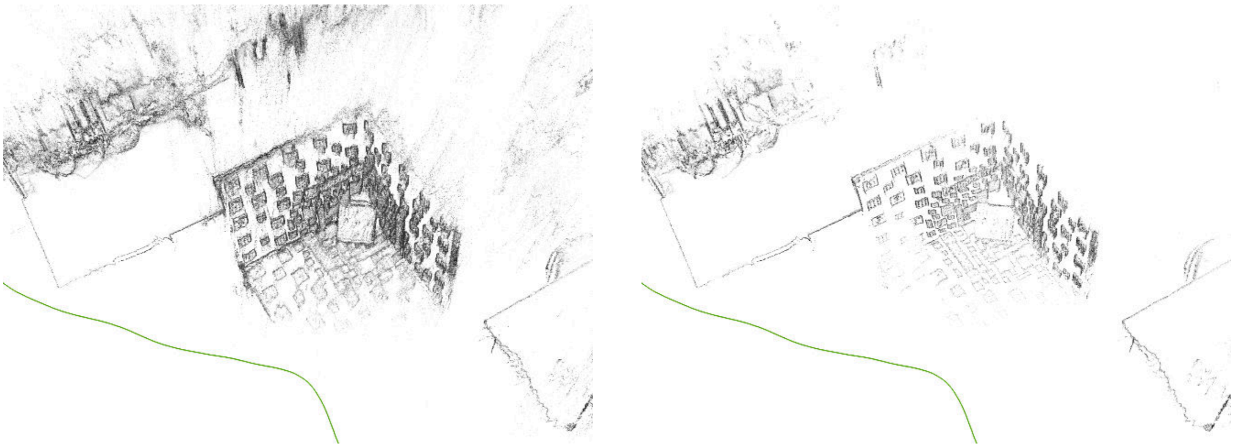
(c) monocular image (magnified)

Figure 10.14: Comparison of image details for the plenoptic and the monocular camera (sequence #4). (a) Totally focused image synthesized by the DPO algorithm. For red regions no intensities are synthesized, since they contain no depth information. (b) Image recorded by the left camera of the stereo camera system. Both images shown in the figure have a resolution of 1024 pixel × 1024 pixel. (c) Magnified subsection of (b) showing approximately the same portion of the scene as (a). (c) has a resolution of 512 pixel × 512 pixel.

For large object distances, with respect to the effective stereo baseline, the noise of the obtained 3D points is dominant in the z_C direction with respect to the camera orientation. For nearby objects the noise is drastically reduced as one can see from the bicycle in the figure.

Figure 10.17 shows the reconstruction of a hallway calculated by DPO. Such straight walks are generally very challenging for monocular VO approaches and result in significant scale drift. However, due to additional parallax gained from the micro images, DPO is able to compensate for scale drifts and performs quite well for such sequences, as can be seen in Figure 10.17.

Figure 10.18 shows the reconstructions of two different staircases. These sequences are especially challenging for DPO, since at the end of the stairs there is usually a narrow corridor with untextured walls where the camera has to turn. Due to the narrow FOV, the recorded images generally contain only few textured areas and tracking has to be performed mainly based on the little texture on the floor. However, our algorithm is still able to perform quite well for these sequences. However, as one can already see from the intensities of the point cloud in Figure 10.18a, the recorded sequence is quite underexposed. Therefore, the intensity gradient threshold in the algorithm is decreased. While this makes the algorithm capable of performing in such a dark



(a) high threshold (b) low threshold

Figure 10.15: Point cloud reconstructed by DPO while considering only points with an inverse virtual depth variance below a certain threshold (sequence #4). (a) High threshold: While the point cloud contains most of the points estimated by DPO, it shows a significant amount of noise. (b) Low threshold: While the point cloud results in a much lower point density, by reducing the threshold, especially noisy points are removed.

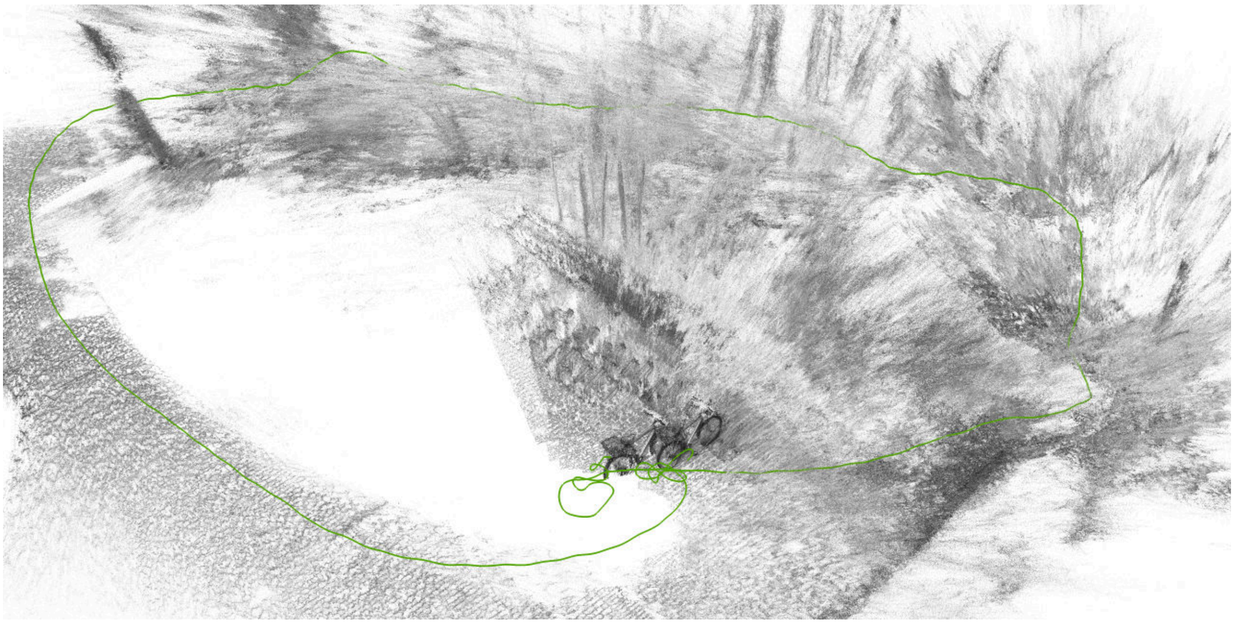


Figure 10.16: Complete trajectory and point cloud calculated by DPO for a scene with abundant vegetation (sequence #3). Even though the point cloud looks quite random, the camera trajectory, which has a length of approximately 80 meters, was estimated accurately.

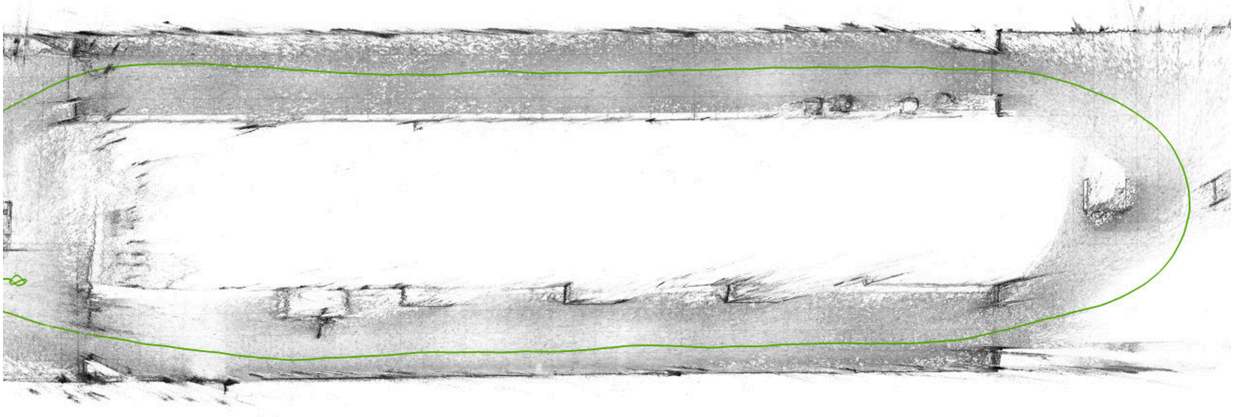
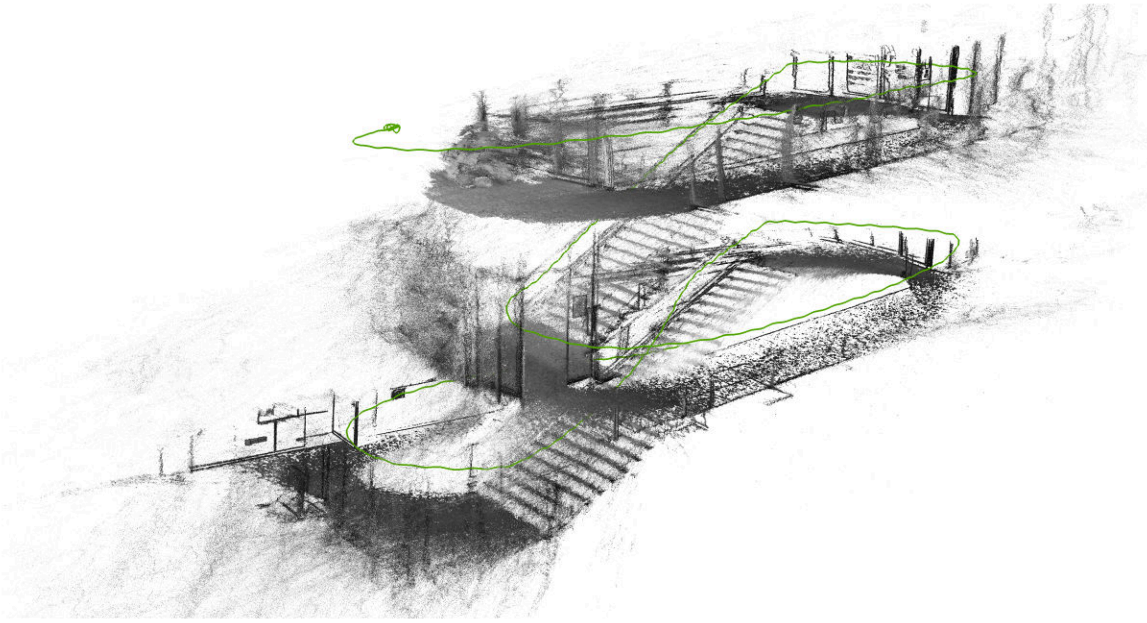


Figure 10.17: Point cloud of a hallway reconstructed by DPO (sequence #7). Due to the additional parallax gained from the micro images, DPO results in only a marginal scale drift for a sequence which is quite challenging for monocular approaches. LSD-SLAM, for instance, failed completely for this sequence.

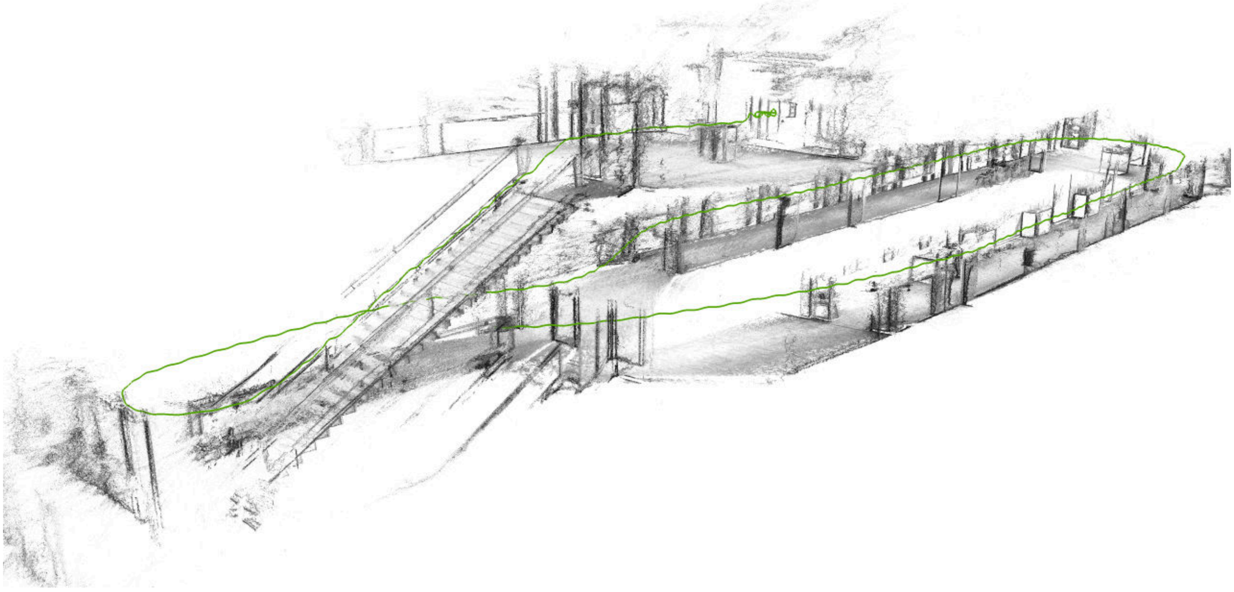
sequence, it also adds noise to the point cloud. The algorithm still failed in a dark spot with next to no texture.

Further samples of the point clouds calculated by DPO are shown in Figure 10.19. The point cloud samples show the versatility of the DPO algorithm, which is able to perform on large scale outdoor and small scale indoor scenes using the same camera setup. Sequence #4 was recorded in our laboratory. For short object distances in the range of few meters, as is the case for this sequence, DPO estimates the absolute scale with an uncertainty of about 1 % and shows only marginal pose drift over the complete sequence. In this laboratory sequence, the camera performed many turns. Hence, baselines used for stereo matching are rather small and certain areas of the scene are observed only for a short period of time. This can be seen from the point cloud in Figure 10.19d, where points far away from the camera trajectory are very noisy. However, nearby objects are reconstructed accurately, as can be seen from Figure 10.19c. This figure shows the starting segment of the trajectory. Here, the camera captured measurement equipment placed on a table. Since the camera was in a range of around 0.5 m to the table, the point cloud contains many details, such as the buttons on the measurement devices.

For the outdoor sequences with object distances of several meters, DPO is still able to calculate accurate point clouds with an absolute scale uncertainty of 5–10 %. DPO especially benefits from objects which are observed for a long period of time and for which the respective depth estimates are refined over multiple keyframes. Figures 10.19a and 10.19b show subsections of the point cloud obtained from such an outdoor sequence. From Figure 10.19a one can clearly see that the point clouds calculated by DPO have a strong variation in terms of accuracy of the 3D points. While the cars in the scene were directly observed by the camera over a long period of time and therefore are reconstructed accurately, the bike racks in the bottom right region of Figure 10.19a are reconstructed very noisily. Sequence #2, for which a subsection of the DPO reconstruction is shown in Figure 10.19d, was recorded on the campus of Karlsruhe University of Applied Sciences. The figure shows stairs which were remodeled very accurately.

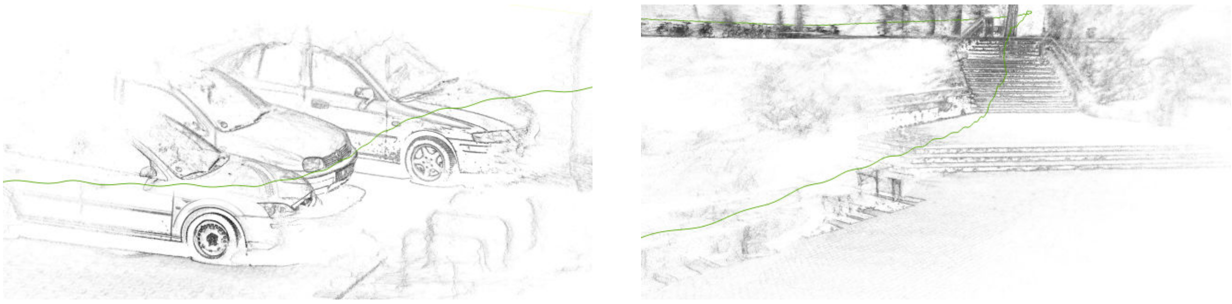


(a) front part of sequence #8



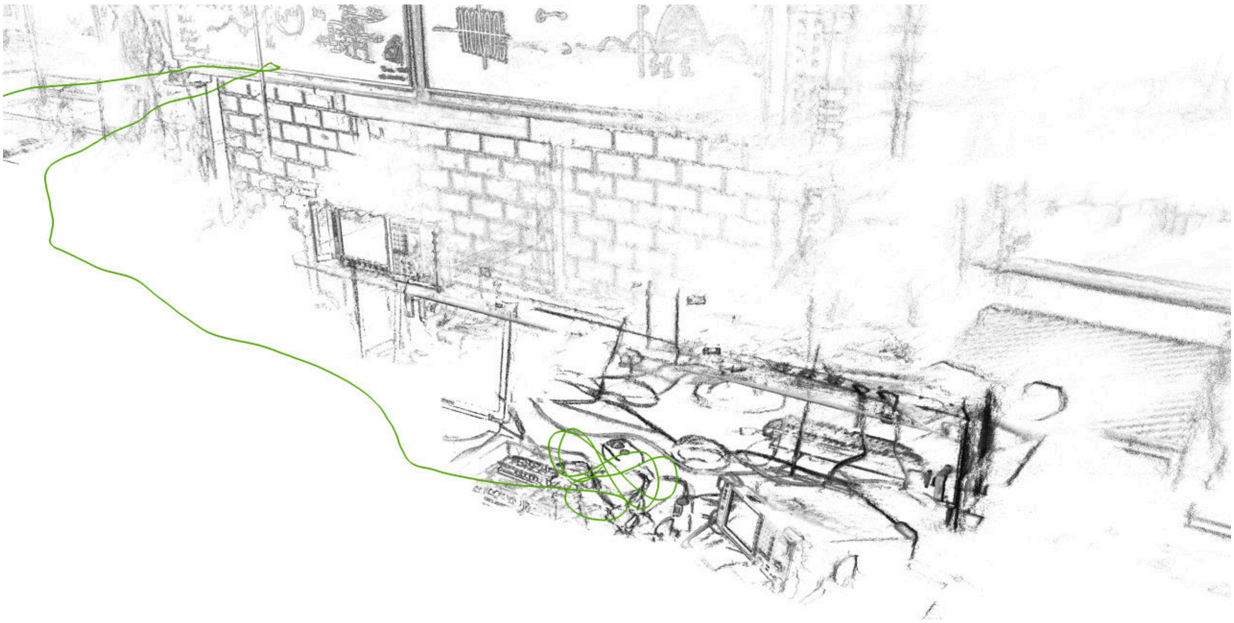
(b) back part of sequence #7

Figure 10.18: Point clouds of stairways reconstructed by DPO.

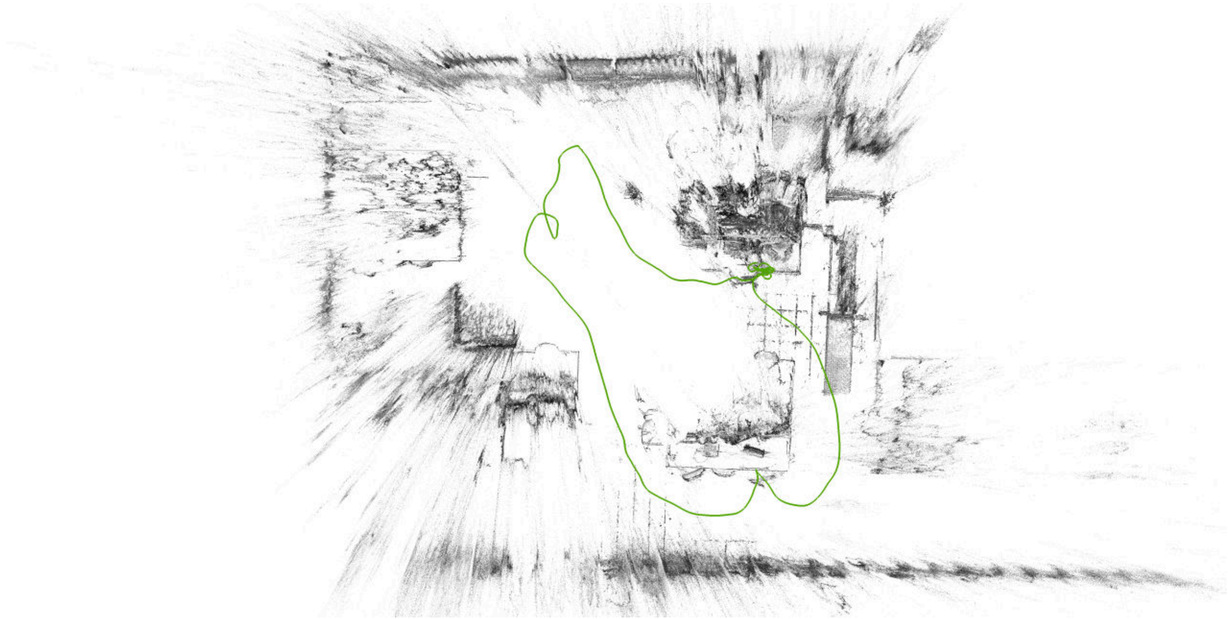


(a) part of sequence #1

(b) part of sequence #2



(c) part of sequence #4



(d) complete trajectory of sequence #4

Figure 10.19: Samples of the point clouds calculated by DPO. The point cloud samples show the versatility of DPO, which is able to perform on large scale outdoor and small scale indoor scenes using the same camera setup.

10.3 Conclusion

On the basis of the results presented in this chapter, it can be concluded that DPO or plenoptic VO generally supplies a promising alternative to VO based on classical camera systems. Especially with respect to monocular approaches, DPO has the ability to estimate the absolute scale with acceptable accuracy. Depending on the application, an absolute scale uncertainty of 10 % might already be sufficient. However, by shifting to larger focal lengths and larger sensor sizes this uncertainty can be significantly improved. In scenes with object distances in the range of few meters, DPO already shows a scale uncertainty of as low as 1-2 %.

To rate the quality of the estimated absolute scale while taking the small stereo baseline between the micro images into account, one can use eq. (4.27) to calculate the uncertainty of a single disparity value for a keyframe. For this calculation an average object distance $z_C = 4\,\mathrm{m}$ is assumed, which is considered to be a plausible value. For the given average object distance, a scale error of 5 % means that the pixel disparity has an uncertainty of 0.036 pixel. In this calculation, the intrinsic parameters presented in Table 9.1 were used. Furthermore, one can argue that if the system is able to reach a disparity uncertainty of 0.036 pixel, it would also be able to obtain a scale uncertainty of 1.55 % if the main lens focal length would be doubled. This expected scale uncertainty is again obtained based on eq. (4.27). For this assessment all camera parameters are kept unchanged and only f_L and b_{L0} respectively are adapted. The parameter b_{L0} is adapted, since, for a larger focal length, the main lens image also moves farther away from the main lens. Thus, one can claim that if the main lens focal length is increased and simultaneously larger sensor is used (e.g. to full-frame), one will be able to significantly improve the performance of DPO.

In addition to the awareness of the absolute scale, DPO benefits from the scale optimization, especially for long sequences. Due to the fact that a scale estimate is obtained for every keyframe, scale drifts can be compensated for. While a monocular camera based algorithm is not able to compensate for scale drifts which occured in the earlier part of the sequence, the scale drift of DPO is bounded. Hence, for long sequences it seems that the drift in the absolute position benefits from the bounded scale error. Monocular approaches instead may, in the limit case of an infinite sequence length, result in an infinite scale difference between start and end of a sequence.

A bottleneck of plenoptic VO is the narrow FOV. This is yet another reason to use larger sensors to obtain a wider FOV, while maintaining the scale awareness. Using a focal length $f_L = 32\,\mathrm{mm}$ in combination with a full-frame sensor would result in a FOV of 59° and a scale awareness superior to the one presented here. At the same time, the narrow FOV of the plenoptic camera results in a small ground sampling distance when compared to the monocular cameras. Thus, the point cloud calculated by DPO shows more details than the one obtained from the monocular LSD-SLAM (Engel et al. [2014]), for instance.

A way to further improve the scale awareness of DPO without affecting the FOV would be to increase the aperture of the micro lenses. This would also come with the positive side effect that a sensor pixel gathers more light energy during the same exposure time. The resulting smaller depth of field (DOF) might be acceptable for most applications.

With respect to the overall tracking accuracy and tracking robustness, there are other algorithms, like ORB-SLAM2 and DSO, which perform better than DPO. Although, both ORB-SLAM2 and DSO use much longer motion stereo baselines between keyframes for mapping and pose optimization. DPO instead performs keyframe based tracking without pose optimization, using motion stereo baselines that range from several centimeters up to only few meters. The missing pose optimization back end, in combination with the small FOV, is also the reason why DPO generally has a rotational drift which is larger than those of DSO and ORB-SLAM2 (mono).

11 Summary and Prospects

11.1 Summary

In this dissertation the benefits of plenoptic cameras with respect to the task of visual odometry (VO) were investigated. To accomplish the task of plenoptic camera based VO a multiple view geometry (MVG) for plenoptic cameras was derived. A probabilistic depth estimation algorithm was introduced which searches for stereo correspondences in the micro images recorded by the camera. The algorithm models observation uncertainties and merges multiple depth observations to a single hypothesis. In order to use the depth estimation approach in the proposed VO algorithm, it was extended by the introduced MVG. In this way, depth can also be estimated based on micro image pairs between subsequent frames. Different plenoptic camera models were derived and examined and calibration approaches were developed based on these models. This enables plenoptic cameras to be used for metrical 3D reconstruction tasks such as VO. The calibration methods perform a bundle adjustment based on a 3D calibration target. Finally, all insights previously gained were combined in a plenoptic camera based VO algorithm called Direct Plenoptic Odometry (DPO). DPO estimates the frame-wise camera position on the basis of a set of selected keyframes as well as a 3D point cloud of the surroundings.

The main advantage of the depth estimation algorithm proposed in this thesis is that it leads to a low number of false estimates. While the method of Hog et al. [2017] produces smooth depth maps, which at first glance look much more appealing than the ones of the proposed algorithm, it also produces clearly visible false estimates. The proposed method instead produces only very few false estimates. At the same time, the depth estimates of this method are noisier than those of the method of Hog et al. [2017]. In terms of its later use in the VO algorithm, stochastic noise is much better tempered than clusters of false estimates. With respect to the algorithm implemented in the software of Raytrix GmbH [2013], RxLive, algorithm proposed in this thesis performs better in all respects. Another advantage, especially with respect to the DPO algorithm, is that the proposed method supplies uncertainties in conjunction with the depth estimates. These uncertainties are used to correctly weight the residuals of the respective 3D points during tracking in the DPO algorithm and to merge multiple depth observations.

The most significant contribution in terms of camera calibration is the bundle adjustment based on a 3D calibration target. Compared to existing methods based on a 2D target, the robustness of the calibration is improved by one order of magnitude. It was also pointed out that the models, which define the complete projection of a single object point to multiple micro images, are estimated more reliably than the ones which are based on the virtual image. Furthermore, the importance of modeling the effect of squinting micro lenses, which was rather neglected in most previous works, is illustrated. The micro image centers received from a recorded white image, in contrast to the assumption often made in previous models (Johannsen et al. [2013]; Heinze et al. [2016]), do not directly represent the centers of the respective micro lenses.

The results obtained for DPO show that a plenoptic camera based VO system provides a promising alternative to existing systems based on monocular or stereo cameras. DPO, based on the current camera setup, is able to estimate the camera trajectory with a scale uncertainty of around 5 % for most of the sequences in the recorded dataset. For close range sequences with object distances in the range of only few meters, the uncertainty is even better. With respect to scale drifts, DPO outperforms state-of-the-art monocular algorithms. Due to the proposed scale optimization strategy, scale drifts for DPO are bounded, as long as good tracking conditions are assured. This is not the case for monocular approaches. Regarding the absolute tracking drifts, DPO is competitive to existing monocular algorithms. Though, the images recorded by the plenoptic camera have a much smaller FOV than the images used by the monocular and stereo algorithms.

11.2 Future Work

We hope that the presented work is only the beginning of visual odometry (VO) and SLAM based on plenoptic (or light field) cameras. There are plenty of ideas left on how to proceed in this field of research.

Plenoptic Camera

This work focused on plenoptic camera based VO from a methodical perspective. Hence, the complete research is based on a standard plenoptic camera available on the market. Even though it was shown that based on this camera promising results already are obtained using the proposed DPO algorithm, this camera is not optimized for the presented task.

To improve the capabilities of DPO, one would need to increase the main lens focal length and simultaneously the FOV of the camera. The only way to achieve this is to increase the sensor size. One suggestion would be to switch to a full frame sensor and simultaneously to double the focal length to approximately $f_L = 32\,\mathrm{mm}$. This would lead to a FOV of approximately 57°, which is about 20° more than for the setup used in this thesis.

Another way to improve the FOV of a plenoptic camera was recently proposed by Dansereau et al. [2017], who built a monocentric light field camera with a FOV of approximately 140°. A monocentric lens supplies a wide FOV, while still having a large aperture as it is needed for light field capturing. While the camera proposed by Dansereau et al. [2017] is at a very early prototype stage, such a camera would push the limits of plenoptic camera based VO. However, for a monocentric lens the focused image is formed on a sphere, instead of on a plane, which demands completely new sensor designs and camera models.

For some indoor sequences, the plenoptic cameras suffers from underexposed images. One method to overcome this is to replace the RGB sensor with a monochromatic sensor. Furthermore, one can increase the F-number of the plenoptic camera to gather more light. The F-number of a plenoptic camera is given by the MLA dimensions and thus has to be adjusted at the stage of construction. Increasing the micro image aperture while keeping all other camera parameters unchanged will also result in a better depth accuracy and therefore in a better scale awareness. The tradeoff for this would be a smaller DOF, which should actually not be an issue for the purpose of VO.

A problem for all kinds of cameras is the limited dynamic range, which may cause trouble for scenes with very bright and dark areas in the same frame. One option would be to build a high dynamic range plenoptic camera as proposed by Georgiev et al. [2009]. Georgiev et al. [2009] use an MLA with different micro lens apertures to produce micro images with different brightness.

Due to the redundancy in the micro images, a well exposed image of the scene can be extracted from the micro images.

Plenoptic Camera Calibration

In the calibration of the plenoptic camera, it was focused on a camera model which conforms to the MVG presented in Chapter 4. In this camera model, the main lens aberration is approximated by distortions defined on the sensor, or MLA plane. A next step would be to relax the constrains given by the MVG in its current form, where the projected MLA is placed on a plane parallel to the main lens. Instead, the lens aberration could be included directly in the main lens projection. This would result in a MLA projected on a curved surface rather than a plane.

Since one will never be able to reliably detect the centers of the calibration markers directly in the micro images, one way to obtain measurements directly from the micro images is to perform a final optimization of the model parameters based on a photometric, instead of a geometric, optimization term. In this way, the geometric camera calibration could even be extended by a photometric calibration.

Plenoptic VO

The DPO algorithm showed that based on a plenoptic camera, accurate odometry can be estimated which is aware of the true scale. To further improve plenoptic camera based VO, strategies known from monocular VO can be adopted. Consequently, the next step would be to apply a pose graph optimization to the keyframes received from DPO. First investigations with respect to pose graph optimization were done already. For this a pose optimization framework similar to the one of Engel et al. [2014] was tested. The results were unsatisfactory and affected the estimated trajectory negatively, especially for long straight walks. Instead, the proper way to do the optimization would be a way similarly as done by Engel et al. [2018]. Here, all model parameters are optimized simultaneously, including depth. If one considers the improvements made from LSD-SLAM (Engel et al. [2014]) to DSO (Engel et al. [2018]) one gets motivated to test such strategies with a plenoptic camera. For certain applications, it might also be desirable to implement SLAM components like loop closure detection or relocalization after tracking failures.

The next step will be to implement DPO such that the algorithm runs in real-time. The prospects for this step are quite promising, as the depth estimation approach presented in Chapter 5 is already running in real time on a CUDA device (Vasko et al. [2015]), while direct image alignment is known to be able to run with very high frame rates on a standard CPU (Kerl et al. [2013b]).

Appendices

A Light Field based Depth Estimation

A.1 Relationship between Virtual Depth and Focus Parameter of Hog et al. [2017]

This section states the prove that the focus parameter g_f introduced by Hog et al. [2017] is equivalent to the virtual depth v estimated by the depth estimation algorithm proposed in Chapter 5 and the one implemented in the RxLive software. This means that $g_f = v$ holds by definition.

One can see that eq. (7) in Hog et al. [2017] (restated in eq. (A.1)) is equivalent to eqs. (5.27) and (5.28) when g is substituted with $v = z^{-1}$.

$$(X, Y) = \left(s\left(g\left(x - c_x\right) + c_x\right), s\left(g\left(y - c_y\right) + c_y\right)\right) \tag{A.1}$$

The coordinates X and Y given in eq. (A.1) are equivalent to the virtual image coordinates x_V and y_V defined in Section 5.5. The coordinates x and y are the raw image coordinates x_R and y_R, and c_x and c_y correspond to the micro lens center coordinates c_{MLx} and c_{MLy}. The parameter s is a scaling factor which is omitted by the definition in Section 5.5, but is actually used in the implementation.

Hog et al. [2017] show that the correct focus parameter g_f is the value g for which the following equation is fulfilled (restatement of eq. (10) in Hog et al. [2017]):

$$g = \frac{D}{D - \delta}. \tag{A.2}$$

Hog et al. [2017] consider only the case of two directly adjacent micro lenses. Hence, D represents the diameter of a micro lens and can be substituted by the baseline distance $\mathit{\Delta} c_{ML} = D_M$. From the derivations in Section 2.4 it results that δ, which is defined by Hog et al. [2017] as the distance between two corresponding raw image points, is equivalent to $\mathit{\Delta} c_{ML} - \mu$ (see Hog et al. [2017] for detailed definitions of D and δ), where μ is the disparity in the micro images. Thus, after substitution eq. (A.2) can be rearranged as follows:

$$g = \frac{\mathit{\Delta} c_{ML}}{\mathit{\Delta} c_{ML} - (\mathit{\Delta} c_{ML} - \mu)} = \frac{\mathit{\Delta} c_{ML}}{\mu}. \tag{A.3}$$

Thus, by definition the focus parameter g_f is equivalent to the virtual depth v (see eq. (2.12) for comparison).

B Direct Plenoptic Odometry

B.1 Pseudo Code of the DPO Algorithm

Algorithm 1: Direct Plenoptic Odometry

```
Initialization:
currenKeyFrame ← get first light field frame in sequence $I_{ML0}(\boldsymbol{x}_R)$;
/* frame initially consists only of raw intensity image $I_{ML}(\boldsymbol{x}_R)$ */
$D_{ML}(\boldsymbol{x}_R)$ ← do in-frame depth estimation on currentKeyFrame; /* see Sec. 7.3.2 */
$D_V(\boldsymbol{x}_V)$ ← calculate virtual image depth map from $D_{ML}(\boldsymbol{x}_R)$;
$D_V(\boldsymbol{x}_V)$ ← regularize depth map $D_V(\boldsymbol{x}_V)$ and remove outliers; /* see Sec. 7.3.4 */
$D_{ML}(\boldsymbol{x}_R)$ ← remove outliers in $D_{ML}(\boldsymbol{x}_R)$ which were detected in $D_V(\boldsymbol{x}_V)$;
$I_V(\boldsymbol{x}_V)$ ← calculate totally focused image for currentKeyFrame using $D_V(\boldsymbol{x}_V)$ and $I_{ML}(\boldsymbol{x}_R)$;
/* indices of current keyframe depth maps and intensity images are
   ignored since they exist once only */
for $j = 1 \rightarrow$ end of sequence do
    currentFrame ← get next light field frame in sequence $I_{MLj}(\boldsymbol{x}_R)$;
    Tracking:
    {$\boldsymbol{\xi}_{kj}$, pointUsage, trackingResidual} ← estimate pose from currentKeyFrame to currentFrame using $D_V(\boldsymbol{x}_V)$, $I_V(\boldsymbol{x}_V)$, $I_{MLj}(\boldsymbol{x}_R)$, $\boldsymbol{\xi}_{k(j-1)}$ and $\dot{\boldsymbol{\xi}}_{j-1}$; /* see Sec. 7.5 */
    Depth Estimation:
    $D_{ML}(\boldsymbol{x}_R)$ ← estimate inter-frame depth from currentKeyFrame to currentFrame using $I_{ML}(\boldsymbol{x}_R)$, $D_{ML}(\boldsymbol{x}_R)$, $I_{MLj}(\boldsymbol{x}_R)$ and $\boldsymbol{\xi}_{kj}$; /* see Sec. 7.3.3 */
    /* new depth estimates are merged into existing depth map $D_{ML}(\boldsymbol{x}_R)$
       as described in Sec. 7.4 */
    $D_V(\boldsymbol{x}_V)$ ← recalculate virtual image depth map from $D_{ML}(\boldsymbol{x}_R)$;
    $D_V(\boldsymbol{x}_V)$ ← regularize depth map $D_V(\boldsymbol{x}_V)$ and remove outliers; /* see Sec. 7.3.4 */
    $D_{ML}(\boldsymbol{x}_R)$ ← remove outliers in $D_{ML}(\boldsymbol{x}_R)$ which were detected in $D_V(\boldsymbol{x}_V)$;
    $I_V(\boldsymbol{x}_V)$ ← recalculate totally focused image for currentKeyFrame using $D_V(\boldsymbol{x}_V)$ and $I_{ML}(\boldsymbol{x}_R)$;
    /* loop continues on next page */
```

Algorithm 1: Direct Plenoptic Odometry (cont.)

```
cont. for
    Keyframe Selection:
    $newKeyFrameScore \leftarrow$ calculate new keyframe score using $\boldsymbol{\xi}_{kj}$ and $pointUsage$;
     /* see Sec. 7.6 */
    if $newKeyFrameScore > keyFrameSelectionThreshold$ then
        Scale Optimization:
        $\{\boldsymbol{\xi}_s^{(0)}, \boldsymbol{\xi}_s^{(1)}, \cdots, \boldsymbol{\xi}_s^{(k)}\} \leftarrow$ add keyframe pose $\boldsymbol{\xi}_s^{(k)} \in \mathfrak{sim}(3)$ to $keyFramePoseList$;
        $\{\rho^{(k)}, (\sigma_\rho^{(k)})^2\} \leftarrow$ estimate log-scale for $currentKeyFrame$; /* see Sec. 7.7.1
         */
        $\{\hat{\rho}^{(0)}, \hat{\rho}^{(1)}, \cdots, \hat{\rho}^{(k)}\} \leftarrow$ optimize keyframe scale based on estimates
         $\{\{\rho^{(0)}, (\sigma_\rho^{(0)})^2\}, \{\rho^{(1)}, (\sigma_\rho^{(1)})^2\}, \cdots, \{\rho^{(k)}, (\sigma_\rho^{(k)})^2\}\}$;     /* see Sec. 7.7.2 */
        $\{\boldsymbol{\xi}_s^{(0)}, \boldsymbol{\xi}_s^{(1)}, \cdots, \boldsymbol{\xi}_s^{(k)}\} \leftarrow$ update $keyFramePoseList$ based on optimized scales
         $\{\hat{\rho}^{(0)}, \hat{\rho}^{(1)}, \cdots, \hat{\rho}^{(k)}\}$;
        $\{D_{ML}(\boldsymbol{x}_R), D_V(\boldsymbol{x}_V)\} \leftarrow$ update scale of $currentKeyFrame$ based on $\hat{\rho}^{(k)}$;
         /* optional operation, not performed in offline mode (see
         Sec. 10.2.1) */
        Replacing Keyframe:
        $newKeyFrame \leftarrow currentFrame$;
        $D_{ML\mathrm{tmp}}(\boldsymbol{x}_R) \leftarrow$ estimate in-frame depth on $newKeyFrame$;            /* see
         Sec. 7.3.2 */
        $D_{ML}(\boldsymbol{x}_R) \leftarrow$ propagate $D_{ML}(\boldsymbol{x}_R)$ from $currentKeyFrame$ to $newKeyFrame$
         using $\boldsymbol{\xi}_{kj}$;
        $D_{ML}(\boldsymbol{x}_R) \leftarrow$ merge $D_{ML}(\boldsymbol{x}_R)$ with $D_{ML\mathrm{tmp}}(\boldsymbol{x}_R)$;
        $currentKeyFrame \leftarrow newKeyFrame$;
        $D_V(\boldsymbol{x}_V) \leftarrow$ recalculate virtual image depth map from $D_{ML}(\boldsymbol{x}_R)$;
        $D_V(\boldsymbol{x}_V) \leftarrow$ regularize depth map $D_V(\boldsymbol{x}_V)$ and remove outliers;         /* see
         Sec. 7.3.4 */
        $D_{ML}(\boldsymbol{x}_R) \leftarrow$ remove outliers in $D_{ML}(\boldsymbol{x}_R)$ which were detected in $D_V(\boldsymbol{x}_V)$;
        $I_V(\boldsymbol{x}_V) \leftarrow$ recalculate totally focused image for $currentKeyFrame$ using
         $D_V(\boldsymbol{x}_V)$ and $I_{ML}(\boldsymbol{x}_R)$;
    end
end
```

B.2 Recursive Solution of the Global Scale Estimator

Equation (B.1) restates the global scale estimator proposed in eq. (7.24).

$$\hat{\rho}^{(l)} = \left(\sum_{m=-N}^{N} \rho^{(m+l)} \cdot \frac{c^{|m|}}{\left(\sigma_\rho^{(m+l)}\right)^2} \right) \cdot \left(\sum_{m=-N}^{N} \frac{c^{|m|}}{\left(\sigma_\rho^{(m+l)}\right)^2} \right)^{-1} \tag{B.1}$$

To avoid confusions, in this section, the log-scale estimates $\rho^{(l)}$ obtained for each keyframe (with index l), as described in Section 7.7.1, will be called log-scale measurements and the outputs of the scale estimator $\hat{\rho}_k^{(l)}$ log-scale estimates.

In the following a recursive formulation of this estimator is presented. Therefore, instead of solving the complete sum (eq. (B.1)) multiple times for each new keyframe, the existing estimates $\hat{\rho}^{(l)}$ merely have to be updated.

Here, l defines the index of the keyframe for which the estimate is calculated, while k is the index of the current keyframe. Thus, the final estimate $\hat{\rho}^{(l)}$ is not available before $k \geq l + N$.

For $k \leq l + N$, the estimates can be refined in the following recursive structure. Here $\hat{\rho}_k^{(l)}$ is the log-scale estimate for the l-th keyframe at the current keyframe index k, while $w_k^{(l)}$ is the corresponding weight. The estimate $\hat{\rho}_k^{(l)}$ can be calculated based on the previous estimate $\hat{\rho}_{k-1}^{(l)}$ as well as the current log-scale measurement $\rho^{(k)}$, and corresponding variance, as follows:

$$\begin{aligned} \hat{\rho}_k^{(l)} &= \left(\hat{\rho}_{k-1}^{(l)} \cdot w_{k-1}^{(l)} + \rho^{(k)} \cdot \frac{c^{k-l}}{\left(\sigma_\rho^{(k)}\right)^2} \right) \cdot \left(w_k^{(l)}\right)^{-1} \\ w_k^{(l)} &= w_{k-1}^{(l)} + \frac{c^{k-l}}{\left(\sigma_\rho^{(k)}\right)^2} \end{aligned} \qquad \text{for } k > l,\ k - l \leq N, \tag{B.2}$$

$$\hat{\rho}_k^{(l)} = \hat{\rho}_{k-1}^{(l)} = \hat{\rho}^{(l)}, \qquad w_k^{(l)} = w_{k-1}^{(l)} = w^{(l)} \qquad \text{for } k - l > N. \tag{B.3}$$

However, there exists no estimate $\hat{\rho}_{k-1}^{(l)}$ for $k \leq l$. Nevertheless, the initial estimate $\hat{\rho}_k^{(k)}$ can be initialized based on the previous estimate of the previous keyframe $\hat{\rho}_{k-1}^{(k-1)}$ as follows:

$$\begin{aligned} \hat{\rho}_k^{(k)} &= \left(\hat{\rho}_{k-1}^{(k-1)} w_{k-1}^{(k-1)} c - \frac{\rho^{(k-N-1)} c^{N+1}}{\left(\sigma_\rho^{(k-N-1)}\right)^2} + \frac{\rho^{(k)}}{\left(\sigma_\rho^{(k)}\right)^2} \right) \cdot \left(w_k^{(k)}\right)^{-1}, \\ w_k^{(k)} &= w_{k-1}^{(k-1)} c - \frac{c^{N+1}}{\left(\sigma_\rho^{(k-N-1)}\right)^2} + \frac{1}{\left(\sigma_\rho^{(k)}\right)^2}. \end{aligned} \tag{B.4}$$

For the formulation given in eq. (B.4) it is considered that $\sigma_\rho^{(k)} \to \infty$ holds for $k < 0$.

In this way, for each new keyframe one only has to update $N + 1$ estimated instead of solving the complete eq. (B.1) $N + 1$ times.

B.3 Results from the Stereo and Plenoptic VO Dataset

Alg.	Metric	Sequence number										
		1	2	3	4	5	6	7	8	9	10	11
DPO	d'_s	1.12	1.07	1.19	1.02	xx	1.24	1.23	xx	1.03	1.03	1.15
	e'_s	1.18	1.08	1.34	1.06	xx	1.05	1.34	xx	1.01	1.08	1.08
	e_r [°]	2.30	1.61	0.75	3.57	xx	2.34	8.03	xx	5.53	3.84	1.33
	e_{align} [%]	3.63	1.41	4.82	2.93	xx	1.09	3.17	xx	0.91	1.56	1.67
DPO (offline)	d'_s	1.02	1.04	1.10	1.02	xx	1.14	1.27	xx	1.02	1.01	1.10
	e'_s	1.06	1.02	1.02	1.04	xx	1.01	1.14	xx	1.01	1.05	1.02
	e_r [°]	2.15	1.59	0.74	3.69	xx	2.34	8.11	xx	5.50	3.74	1.29
	e_{align} [%]	3.13	2.25	1.26	1.75	xx	0.55	2.28	xx	1.23	1.20	0.59
DPO (online)	d'_s	1.00	1.07	1.16	1.01	xx	1.17	1.35	xx	1.02	1.03	1.08
	e'_s	1.01	1.07	1.14	1.07	xx	1.09	1.22	xx	1.01	1.01	1.03
	e_r [°]	2.24	1.54	0.74	1.53	xx	2.28	8.54	xx	4.31	3.85	1.25
	e_{align} [%]	2.54	2.07	2.41	0.80	xx	1.46	2.78	xx	0.76	1.08	0.76
DSO	d'_s	–	–	–	–	–	–	–	–	–	–	–
	e'_s	1.13	1.15	1.11	1.00	1.15	xx	1.06	1.50	1.01	1.09	1.19
	e_r [°]	1.01	0.07	0.61	0.87	1.11	xx	0.57	1.93	0.23	3.05	0.37
	e_{align} [%]	1.66	2.07	1.24	0.32	2.02	xx	0.65	2.21	0.06	1.60	2.68
LSD-SLAM	d'_s	–	–	–	–	–	–	–	–	–	–	–
	e'_s	2.19	2.66	10.5	1.68	1.88	xx	xx	xx	1.44	1.42	7.88
	e_r [°]	2.80	9.67	27.5	9.49	33.2	xx	xx	xx	9.96	30.8	1.13
	e_{align} [%]	11.6	26.8	19.1	9.76	29.0	xx	xx	xx	2.99	10.2	39.0
ORB2 (mono)	d'_s	–	–	–	–	–	–	–	–	–	–	–
	e'_s	1.56	1.34	1.34	1.10	1.84	xx	1.22	xx	1.09	1.09	1.27
	e_r [°]	0.81	0.58	0.47	1.05	1.92	xx	1.14	xx	1.17	2.15	0.30
	e_{align} [%]	6.00	4.47	3.76	0.88	9.19	xx	1.87	xx	0.71	1.51	3.67
ORB (stereo)	d'_s	1.01	xx	1.00	1.01	1.00	1.00	1.00	1.04	1.00	1.00	1.00
	e'_s	1.00	xx	1.00	1.00	1.01	1.01	1.02	1.02	1.00	1.01	1.01
	e_r [°]	4.21	xx	0.92	4.71	1.22	1.25	6.10	18.0	2.40	2.37	0.62
	e_{align} [%]	1.37	xx	0.69	1.85	0.32	0.49	1.06	1.51	0.47	0.86	0.17
RGBD-VO	d'_s	1.01	1.01	1.00	1.00	1.00	xx	1.02	1.03	1.00	1.00	1.01
	e'_s	1.00	1.00	1.01	1.03	1.01	xx	1.02	1.02	1.01	1.00	1.01
	e_r [°]	5.06	4.58	69.2	2.08	22.3	xx	38.1	76.0	4.67	4.42	3.94
	e_{align} [%]	2.65	1.27	18.3	1.16	8.68	xx	6.21	4.69	0.83	5.45	1.35

Table B.1: All results from the stereo and plenoptic VO dataset. For all monocular algorithms no absolute scale d'_s can be estimated. The alignment error e_{align} is given in percentages of the respective sequence length. xx means that the respective algorithm failed on a sequence and therefore no metric is obtained.

Sequence number	1	2	3	4	5	6	7	8	9	10	11
Length in meter	240	195	81	25	274	229	149	161	29	117	248

Table B.2: Path lengths for all sequences in the stereo and plenoptic VO dataset.

B.4 Trajectories Estimated by DPO

Figures B.1 to B.9 show the trajectories and corresponding point clouds calculated by DPO of all sequences for which the algorithm succeeded.

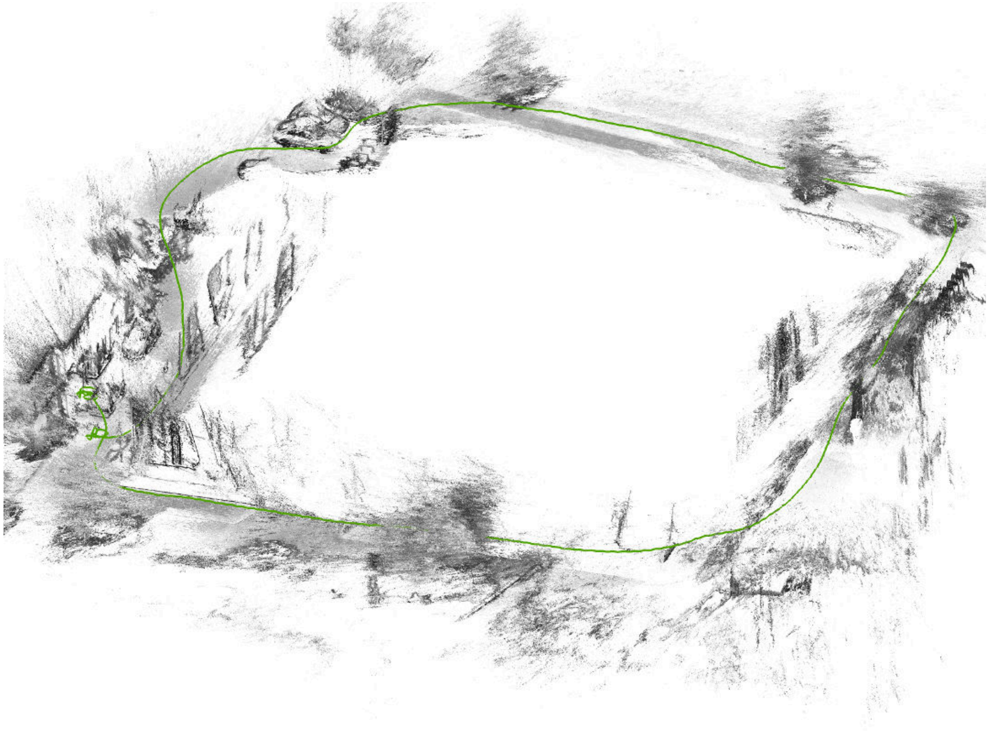


Figure B.1: Top views of the trajectory of sequence #1 calculated by DPO. Path length: 240 m. Walk around “Mensa Moltke” in Karlsruhe. On the top left, one can see the parked cars which were shown in Figure 10.19a. Both views were rendered from exactly the same point cloud data.

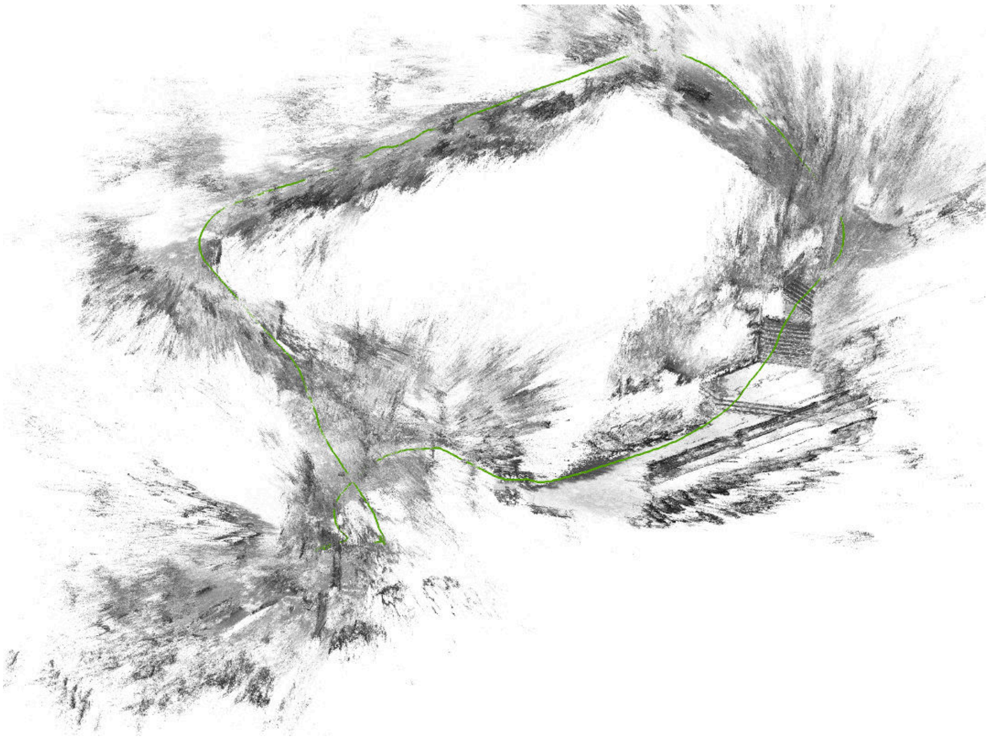


Figure B.2: Top views of the trajectory of sequence #2 calculated by DPO. Path length: 195 m. Walk on the campus of Karlsruhe University of Applied Sciences. On the right side, one can see stairs and parts of a building (Building A).

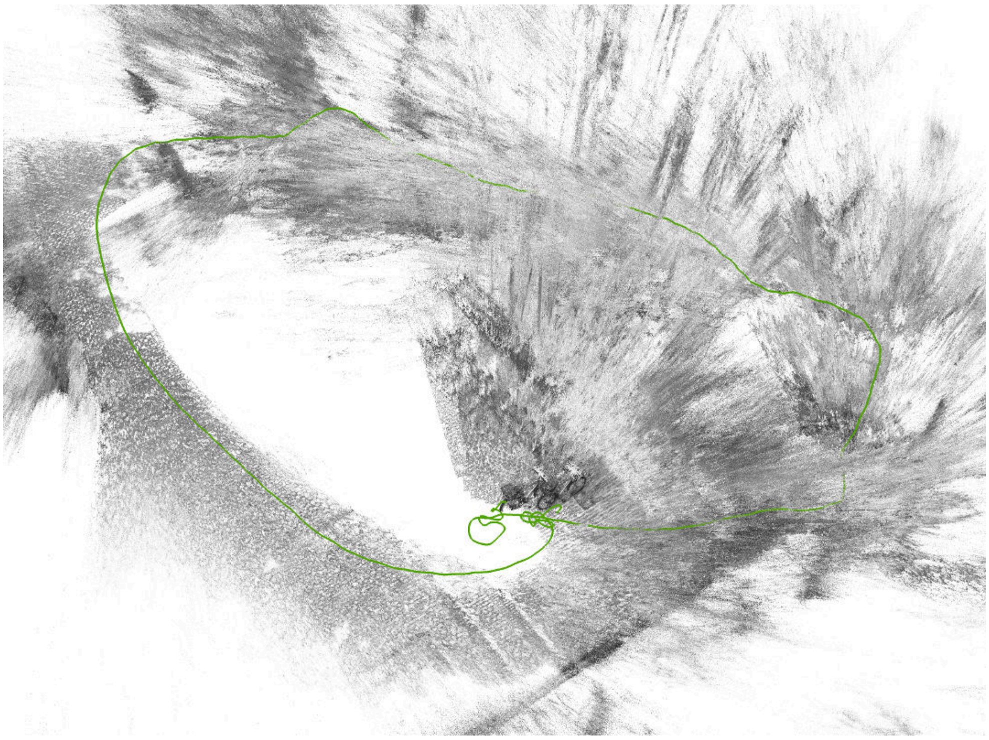


Figure B.3: Top views of the trajectory of sequence #3 calculated by DPO. Path length: 81 m. Walk through bushes on the campus of Karlsruhe University of Applied Sciences.

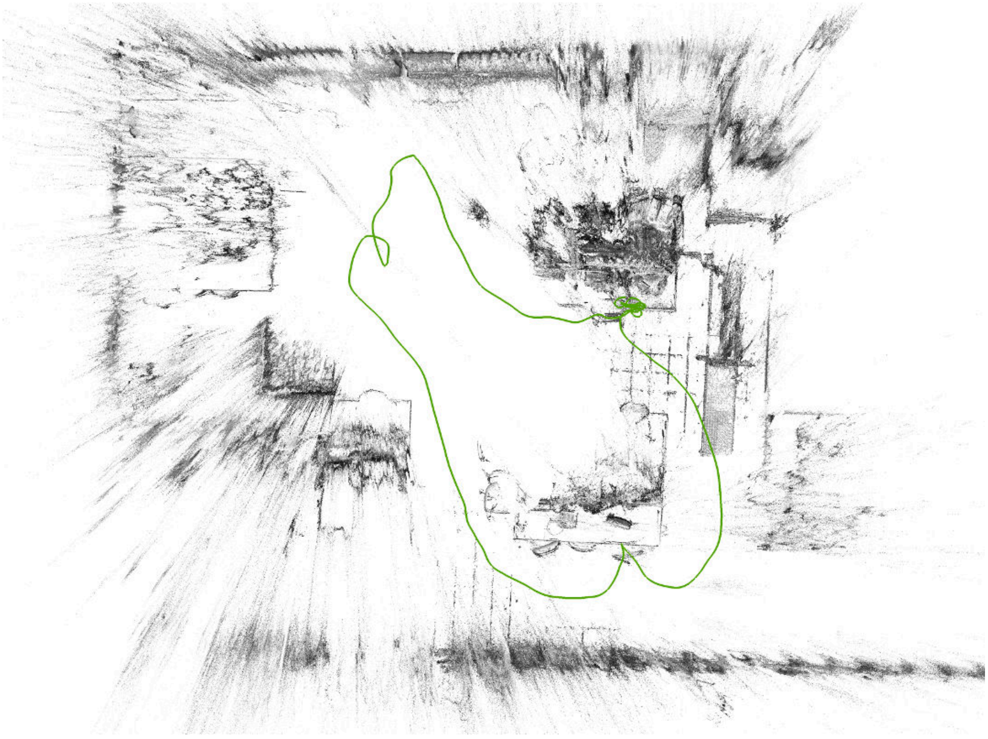


Figure B.4: Top views of the trajectory of sequence #4 calculated by DPO. Path length: 25 m. Walk though the laboratory of communications engineering at Karlsruhe University of Applied Sciences (room: LI-013a).

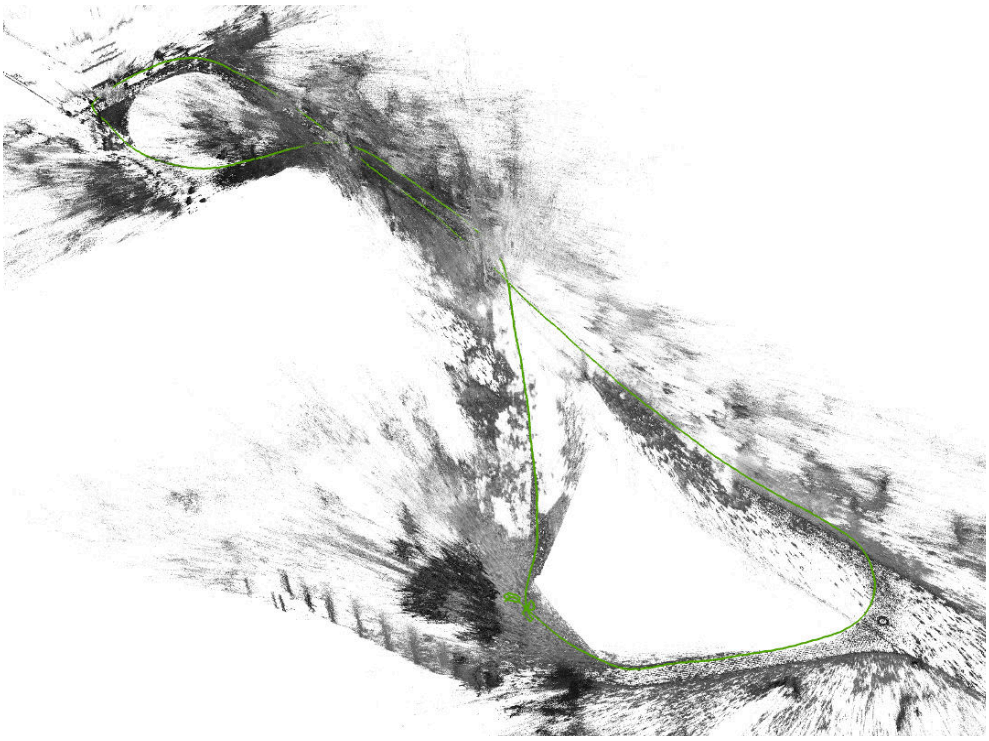


Figure B.5: Top views of the trajectory of sequence #6 calculated by DPO. Path length: 229 m. Walk on the campus of Karlsruhe University of Applied Sciences (behind Building LI, at Knielinger Allee and Adenauerring). In the top left corner, one can see cars parking in the street.

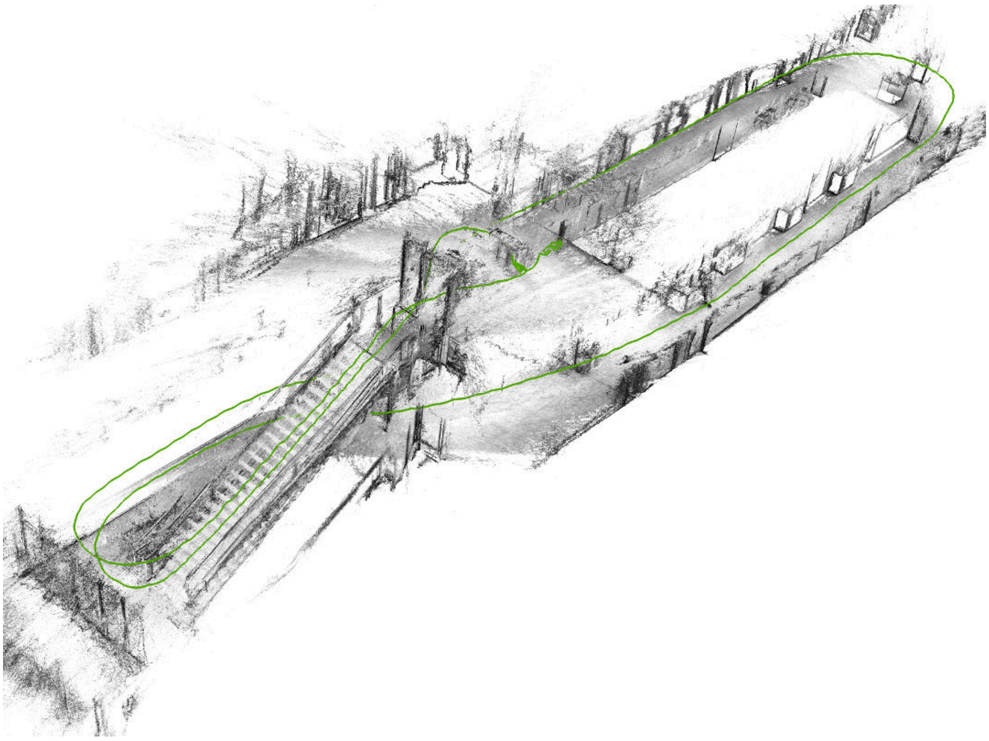


Figure B.6: Top views of the trajectory of sequence #7 calculated by DPO. Path length: 149 m. Walk through building B at Karlsruhe University of Applied Sciences.

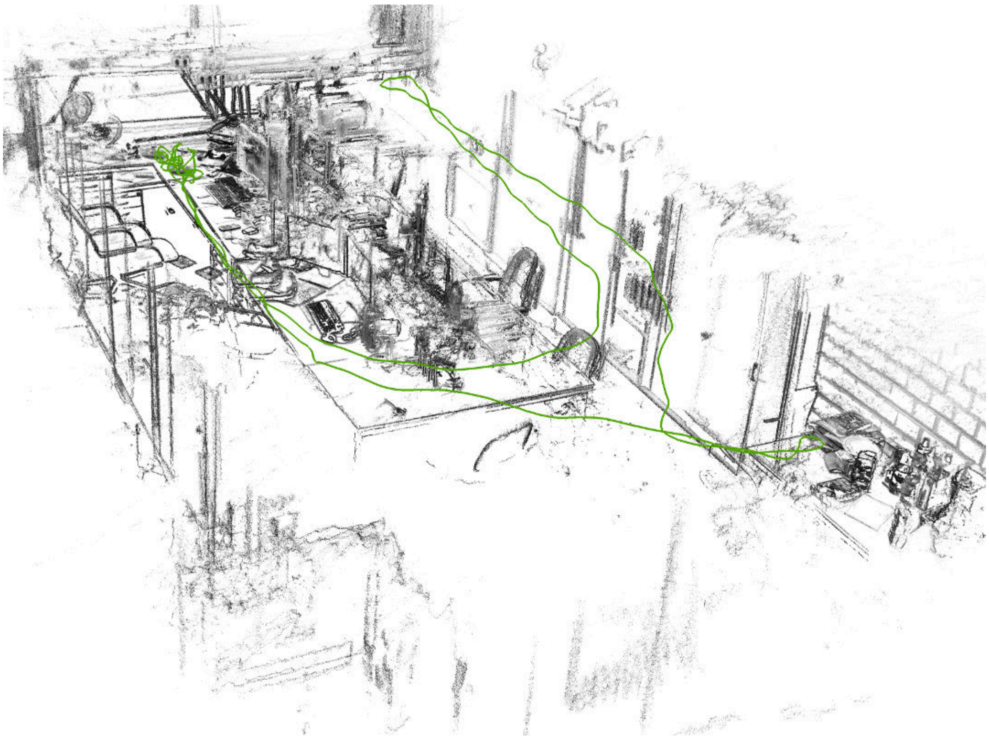


Figure B.7: Top views of the trajectory of sequence #9 calculated by DPO. Path length: 29 m. Walk through my office (room LI-036) at Karlsruhe University of Applied Sciences.

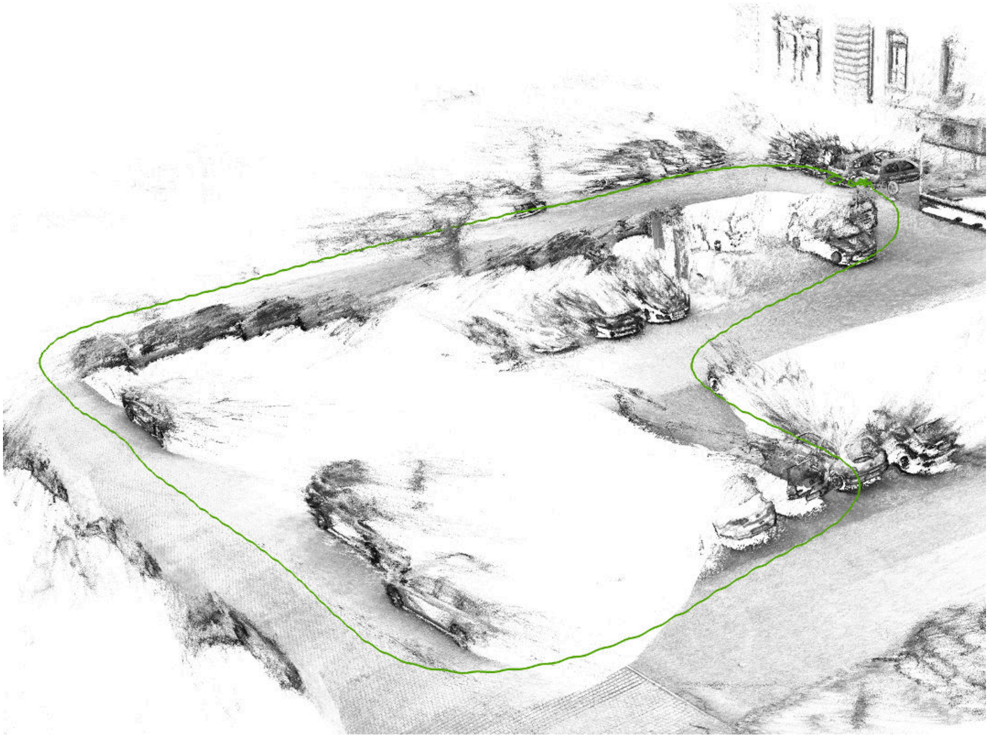


Figure B.8: Top views of the trajectory of sequence #10 calculated by DPO. Path length: 117 m. Walk across parking lot at Karlsruhe University of Education.

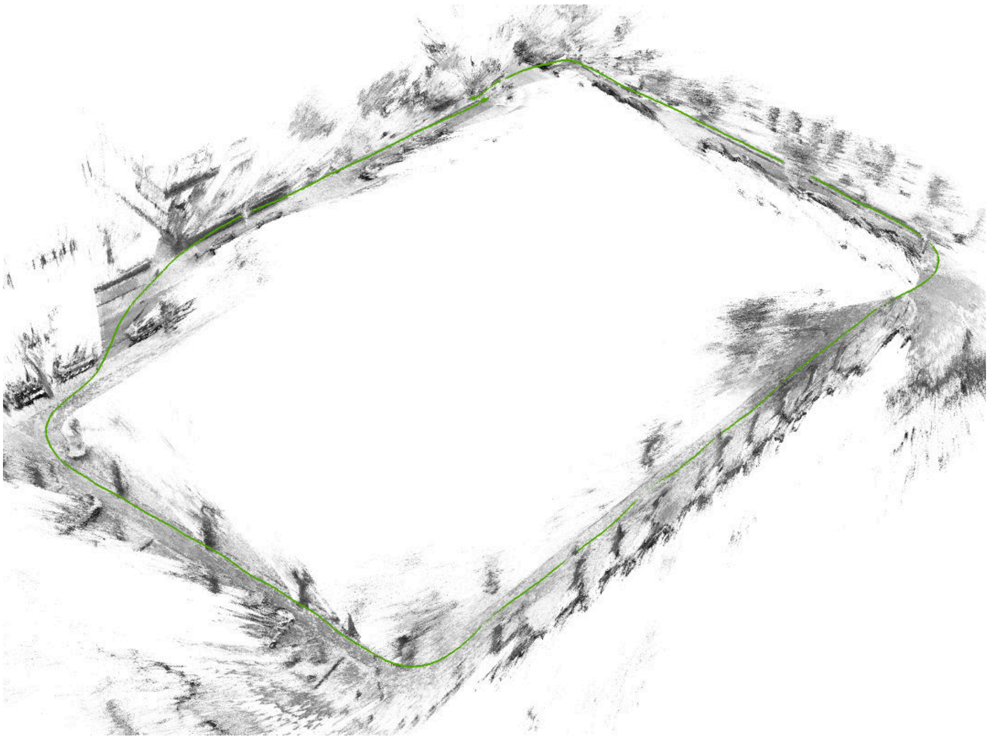


Figure B.9: Top views of the trajectory of sequence #11 calculated by DPO. Path length: 248 m. Walk around Scheffelplatz in Karlsruhe.

Bibliography

Adelson EH and Bergen JR (1991). The Plenoptic Function and the Elements of Early Vision. In *Computational Models of Visual Processing*, pages 3–20. MIT Press.

Adelson EH and Wang JYA (1992). Single Lens Stereo with a Plenoptic Camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):99–106.

Besl PJ and McKay ND (1992). A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.

Bishop TE and Favaro P (2009). Plenoptic Depth Estimation from Multiple Aliased Views. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1622–1629.

Bishop TE and Favaro P (2011). Full-Resolution Depth Map Estimation from an Aliased Plenoptic Light Field. In *Computer Vision – ACCV 2010*, volume 6493 of *Lecture Notes in Computer Science*, pages 186–200.

Bishop TE and Favaro P (2012). The Light Field Camera: Extended Depth of Field, Aliasing, and Superresolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):972–986.

Bok Y, Jeon HG, and Kweon IS (2014). Geometric Calibration of Micro-Lens-Based Light-Field Cameras Using Line Features. In *Computer Vision - ECCV 2014*, volume 8694 of *Lecture Notes in Computer Science*, pages 47–61. Springer International Publishing.

Bok Y, Jeon HG, and Kweon IS (2017). Geometric Calibration of Micro-Lens-Based Light Field Cameras Using Line Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):287–300.

Brown DC (1966). Decentering Distortion of Lenses. *Photogrammetric Engineering*, 32(3):444–462.

Bylow E, Sturm J, Kerl C, Kahl F, and Cremers D (2013). Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions. In *Robotics: Science and Systems Conference (RSS)*.

Chen C, Lin H, Yu Z, Bing Kang S, and Yu J (2014). Light Field Stereo Matching Using Bilateral Statistics of Surface Cameras. In *IEEE Conference on Conputer Vision and Pattern Recognition (CVPR)*, pages 1518–1525.

Chen Y and Medioni G (1992). Object Modelling by Registration of Multiple Range Images. *Image and Vision Computing*, 10(3):145–155.

Chiuso A, Favaro P, Jin H, and Soatto S (2002). Structure from Motion Causally Integrated over Time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):523–535.

Cho D, Lee M, Kim S, and Tai YW (2013). Modeling the Calibration Pipeline of the Lytro Camera for High Quality Light-Field Image Reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3280–3287.

Concha A and Civera J (2014). Using Superpixels in Monocular SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 365–372.

Curless B and Levoy M (1996). A Volumetric Method for Building Complex Models from Range Images. In *ACM Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 303–312.

Dansereau DG (2013). *Plenoptic Signal Processing for Robust Vision in Field Robotics.* PhD thesis, University of Sydney; Graduate School of Engineering and IT; School of Aerospace, Mechanical and Mechatronic Engineering.

Dansereau DG, Mahon I, Pizarro O, and Williams S (2011). Plenoptic Flow: Closed-Form Visual Odometry for Light Field Cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4455–4462.

Dansereau DG, Pizarro O, and Williams S (2013). Decoding, Calibration and Rectification for Lenselet-based Plenoptic Cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1027–1034.

Dansereau DG, Schuster G, Ford J, and Wetzstein G (2017). A Wide-Field-of-View Monocentric Light Field Camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3757–3766.

Davison AJ, Reid ID, Molton ND, and Stasse O (2007). MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067.

Dong F, Ieng SH, Savatier X, Etienne-Cummings R, and Benosman R (2013). Plenoptic Cameras in Real-Time Robotics. *The International Journal of Robotics Research*, 32(2):206–217.

Drazic V and Sabater N (2012). A Precise Real-time Stereo Algorithm. In *Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 138–143.

Dryanovski I, Jaramillo C, and Xiao J (2012). Incremental Registration of RGB-D Images. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1685–1690.

Dryanovski I, Valenti RG, and Xiao J (2013). Fast Visual Odometry and Mapping from RGB-D Data. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2305–2310.

Eade E and Drummond T (2009). Edge Landmarks in Monocular SLAM. *Image and Vision Computing*, 27(5):588–596.

Endres F, Hess J, Sturm J, Cremers D, and Burgard W (2014). 3-D Mapping with an RGB-D Camera. *IEEE Transactions on Robotics*, 30(1):177–187.

Engel J, Koltun V, and Cremers D (2018). Direct Sparse Odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625.

Engel J, Schöps T, and Cremers D (2014). LSD-SLAM: Large-Scale Direct Monocular SLAM. In *European Conference on Computer Vision (ECCV)*, pages 834–849.

Engel J, Stückler J, and Cremers D (2015). Large-Scale Direct SLAM with Stereo Cameras. In *International Conference on Intelligent Robots and Systems (IROS).*

Engel J, Sturm J, and Cremers D (2013). Semi-dense Visual Odometry for a Monocular Camera. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1449–1456.

Engel J, Usenko V, and Cremers D (2016). A Photometrically Calibrated Benchmark For Monocular Visual Odometry. In *arXiv:1607.02555.*

Forster C, Pizzoli M, and Scaramuzza D (2014). SVO: Fast Semi-direct Monocular Visual Odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22.

Forster C, Zhang Z, Gassner M, Werlberger M, and Scaramuzza D (2017). SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. *IEEE Transactions on Robotics*, 33(2):249–265.

Galvez-López D and Tardós JD (2012). Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197.

Georgiev T, Lumsdaine A, and Goma S (2009). High Dynamic Range Image Capture with Plenoptic 2.0 Camera. In *Signal Recovery and Synthesis*. Optical Society of America.

Georgiev T, Zheng KC, Curless B, Salesin D, Nayar S, and Intwala C (2006). Spatio-Angular Resolution Tradeoffs in Integral Photography. In *17th Eurographics Conference on Rendering Techniques*, pages 263–272.

Gershun A (1936). The Light Field. *Journal of Mathematics and Physics, MIT*, XVIII:51–151. Translated by P. Moon and G. Timoshenko.

Gomez-Ojeda R, Briales J, and Gonzalez-Jimenez J (2016). PL-SVO: Semi-direct Monocular Visual Odometry by Combining Points and Line Segments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4211–4216.

Gortler SJ, Grzeszczuk R, Szeliski R, and Cohen MF (1996). The Lumigraph. In *23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 43–54.

Hahne C (2016). *The Standard Plenoptic Camera: Applications of a Geometrical Light Field Model*. PhD thesis, University of Bedfordshire.

Hahne C, Aggoun A, Haxha S, Velisavljevic V, and Fernández JCJ (2014). Baseline of Virtual Cameras Acquired by a Standard Plenoptic Camera Setup. In *2014 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–3.

Hartley R and Zisserman A (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition.

Heber S and Pock T (2014). Shape From Light Field meets Robust PCA. In *European Conference on Computer Vision (ECCV)*, pages 751–767.

Heber S and Pock T (2016). Convolutional Networks for Shape from Light Field. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3746–3754.

Heber S, Ranftl R, and Pock T (2013). Variational Shape from Light Field. In Heyden A, Kahl F, Olsson C, Oskarsson M, and Tai XC, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 8081 of *Lecture Notes in Computer Science*, pages 66–79.

Heinze C, Spyropoulos S, Hussmann S, and Perwaß C (2015). Automated Robust Metric Calibration of Multi-focus Plenoptic Cameras. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 2038–2043.

Heinze C, Spyropoulos S, Hussmann S, and Perwaß C (2016). Automated Robust Metric Calibration Algorithm for Multifocus Plenoptic Cameras. *IEEE Transactions on Instrumentation and Measurement*, 65(5):1197–1205.

Henry P, Krainin M, Herbst E, Ren X, and Fox D (2010). RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments. In *International Symposium on Experimental Robotics (ISER)*.

Henry P, Krainin M, Herbst E, Ren X, and Fox D (2012). RGB-D Mapping: Using Kinect-Style Depth Cameras for Dense 3D Modeling of Indoor Environments. *The International Journal of Robotics Research*, 31(5):647–663.

Hirschmüller H (2008). Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341.

Hog M, Sabater N, Vandame B, and Drazic V (2017). An Image Rendering Pipeline for Focused Plenoptic Cameras. *IEEE Transactions on Computational Imaging*, 3(4):811–821.

Honauer K, Johannsen O, Kondermann D, and Goldlücke B (2017). A Dataset and Evaluation Methodology for Depth Estimation on 4D Light Fields. In *Computer Vision – ACCV 2016*, pages 19–34. Springer International Publishing.

Horn BKP (1984). Extended Gaussian Images. *Proceedings of the IEEE*, 72(12):1671–1686.

Hu G, Huang S, Zhao L, Alempijevic A, and Dissanayake G (2012). A Robust RGB-D SLAM Algorithm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1714–1719.

Huang AS, Bachrach A, Henry P, Krainin M, Fox D, and Roy N (2011). Visual Odometry and Mapping for Autonomous Flight using an RGB-D Camera. In *International Symposium of Robotics Research (ISRR)*.

Huber PJ (1964). Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73–101.

Izadi S, Kim D, Hilliges O, Molyneaux D, Newcombe R, Kohli P, Shotton J, Hodges S, Freeman D, Davison A, and Fitzgibbon A (2011). KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *24th Annual ACM Symposium on User Interface Software and Technology*, pages 559–568.

Jeon HG, Park J, Choe G, Park J, Bok Y, Tai YW, and Kweon IS (2015). Accurate Depth Map Estimation from a Lenslet Light Field Camera. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1547–1555.

Jin H, Favaro P, and Soatto S (2003). A Semi-direct Approach to Structure from Motion. *The Visual Computer: International Journal of Computer Graphics*, 19(6):377–394.

Johannsen O, Heinze C, Goldlücke B, and Perwaß C (2013). On the Calibration of Focused Plenoptic Cameras. In *Time-of-Flight and Depth Imaging : Sensors, Algorithms, and Applications*, number 8200 in Lecture notes in computer science, pages 302–317.

Johannsen O, Sulc A, and Goldlücke B (2015). On Linear Structure from Motion for Light Field Cameras. In *IEEE International Conference on Computer Vision (ICCV)*, pages 720–728.

Kerl C, Stückler J, and Cremers D (2015). Dense Continuous-Time Tracking and Mapping with Rolling Shutter RGB-D Cameras. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2264–2272.

Kerl C, Sturm J, and Cremers D (2013a). Dense Visual SLAM for RGB-D Cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2100–2106.

Kerl C, Sturm J, and Cremers D (2013b). Robust Odometry Estimation for RGB-D Cameras. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3748–3754.

Kim C, Zimmer H, Pritch Y, Sorkine-Hornung A, and Gross M (2013). Scene Reconstruction from High Spatio-Angular Resolution Light Fields. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 32(4).

Klein G and Murray D (2007). Parallel Tracking and Mapping for Small AR Workspaces. In *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, volume 6, pages 225–234.

Klein G and Murray D (2008). Improving the Agility of Keyframe-Based SLAM. In *European Conference on Computer Vision (ECCV)*, pages 802–815.

Klein G and Murray D (2009). Parallel Tracking and Mapping on a Camera Phone. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 83–86.

Klose S, Heise P, and Knoll A (2013). Efficient Compositional Approaches for Real-Time Robust Direct Visual Odometry from RGB-D Data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1100–1106.

Konz J, Zeller N, and Quint F (2016). Depth Estimation from Micro Images of a Plenoptic Camera. In *BW-CAR Symposium on Information and Communication Systems (SInCom)*, pages 17–23.

Kühefuß A, Zeller N, and Quint F (2016). Feature Based RGB-D SLAM for a Plenoptic Camera. In *BW-CAR Symposium on Information and Communication Systems (SInCom)*, pages 25–29.

Leutenegger S, Lynen S, Bosse M, Siegwart R, and Furgale P (2015). Keyframe-based Visual-Inertial Odometry Using Nonlinear Optimization. *International Journal of Robotics Research*, 34(3):314–334.

Levoy M and Hanrahan P (1996). Light Field Rendering. In *23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 31–42.

Li M and Mourikis AI (2013). High-Precision, Consistent EKF-Based Visual-Inertial Odometry. *The International Journal of Robotics Research*, 32(6):690–711.

Lin H, Chen C, Kang SB, and Yu J (2015). Depth Recovery from Light Field Using Focal Stack Symmetry. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3451–3459.

Lovegrove S, Patron-Perez A, and Sibley G (2013). Spline Fusion: A Continuous-Time Representation for Visual-Inertial Fusion with Application to Rolling Shutter Cameras. In *British Machine Vision Conference (BMVC)*.

Luhmann T, Robson S, Kyle S, and Boehm J (2014). *Close-Range Photogrammetry and 3D Imaging*. Walter de Gruyter.

Lumsdaine A and Georgiev T (2009). The Focused Plenoptic Camera. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–8.

Lytro (2013). `https://www.lytro.com`. (Retrieved: November 2017).

Ma Y, Soatto S, Kosecka J, and Sastry SS (2004). *An Invitation to 3-D Vision – From Images to Geometric Models*. Springer.

Mei C, Sibley G, Cummins M, Newman P, and Reid I (2011). RSLAM: A System for Large-Scale Mapping in Constant-Time Using Stereo. *International Journal of Computer Vision*, 94(2):198–214.

Mourikis AI and Roumeliotis SI (2007). A Multi-State Constraint Kalman Filter for Vision-Aided Inertial Navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3565–3572.

Mur-Artal R, Montiel JMM, and Tardós JD (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163.

Mur-Artal R and Tardós JD (2017). ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5):1–8.

Murray RM, Li Z, and Sastry SS (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press.

Newcombe RA, Lovegrove SJ, and Davison AJ (2011). DTAM: Dense Tracking and Mapping in Real-Time. In *IEEE International Conference on Computer Vision (ICCV)*.

Ng R and Hanrahan P (2006). Digital Correction of Lens Aberrations in Light Field Photography. In *SPIE 6342, International Optical Design Conference*, volume 6342.

Ng R, Levoy M, Brédif M, Duval G, Horowitz M, and Hanrahan P (2005). Light Field Photography with a Hand-Held Plenoptic Camera. Technical report, Stanford University Computer Science Tech Report CSTR 2005-02.

Osteen PR, Owens JL, and Kessens CC (2012). Online Egomotion Estimation of RGB-D Sensors using Spherical Harmonics. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1679–1684.

Perwaß C and Wietzke L (2012). Single Lens 3D-Camera with Extended Depth-of-Field. In *SPIE 8291, Human Vision and Electronic Imaging XVII.*

Pomerleau F, Magnenat S, Colas F, Liu M, and Siegwart R (2011). Tracking a Depth Camera: Parameter Exploration for Fast ICP. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3824–3829.

Ranftl R, Vineet V, Chen Q, and Koltun V (2016). Dense Monocular Depth Estimation in Complex Dynamic Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4058–4066.

Raytrix GmbH (2013). `http://www.raytrix.de`. (Retrieved: November 2017).

Rublee E, Rabaud V, Konolige K, and Bradski G (2011). ORB: An Efficient Alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571.

Schöps T, Engel J, and Cremers D (2014). Semi-dense Visual Odometry for AR on a Smartphone. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 145–150.

Schöps T, Schönberger JL, Galliani S, Sattler T, Schindler K, Pollefeys M, and Geiger A (2017). A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos. In *Conference on Computer Vision and Pattern Recognition (CVPR).*

Skinner KA and Johnson-Roberson M (2016). Towards Real-Time Underwater 3D Reconstruction with Plenoptic Cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2014–2021.

Steinbrücker F, Sturm J, and Cremers D (2011). Real-Time Visual Odometry from Dense RGB-D Images. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 719–722.

Strasdat H (2012). *Local Accuracy and Global Consistency for Efficient Visual SLAM.* PhD thesis, Imperial College London.

Strasdat H, Montiel JMM, and Davison AJ (2010a). Real-Time Monocular SLAM: Why Filter? In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2657–2664.

Strasdat H, Montiel JMM, and Davison AJ (2010b). Scale Drift-Aware Large Scale Monocular SLAM. In *Robotics: Science and Systems (RSS).*

Strasdat H, Montiel JMM, and Davison AJ (2012). Visual SLAM: Why Filter? *Image and Vision Computing*, 30(2):65–77.

Strobl KH and Lingenauber M (2016). Stepwise Calibration of Focused Plenoptic Cameras. *Computer Vision and Image Understanding*, 145:140–147.

Tao MW, Hadap S, Malik J, and Ramamoorthi R (2013). Depth from Combining Defocus and Correspondence Using Light-Field Cameras. In *IEEE International Conference on Computer Vision (ICCV)*, pages 673–680.

Tao MW, Srinivasan PP, Malik J, Rusinkiewicz S, and Ramamoorthi R (2015). Depth from Shading, Defocus, and Correspondence using Light-Field Angular Coherence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1940–1948.

Tao MW, Wang TC, Malik J, and Ramamoorthi R (2014). Depth Estimation for Glossy Surfaces with Light-Field Cameras. In Agapito L, Bronstein MM, and Rother C, editors, *Computer Vision - ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part II*, pages 533–547.

Thomason CM, Fahringer TF, and Thurow BS (2014). Calibration of a Microlens Array for a Plenoptic Camera. In *52nd Aerospace Sciences Meeting, AIAA SciTech*.

Tosic I and Berkner K (2014). Light Field Scale-Depth Space Transform for Dense Depth Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 441–448.

Usenko V, Engel J, Stückler J, and Cremers D (2016). Direct Visual-Inertial Odometry with Stereo Cameras. In *International Conference on Robotics and Automation (ICRA)*.

Valgaerts L, Bruhn A, Mainberger M, and Weickert J (2012). Dense Versus Sparse Approaches for Estimating the Fundamental Matrix. *International Journal of Computer Vision*, 96(2):212–234.

Vasko R, Zeller N, Quint F, and Stilla U (2015). A Real-Time Depth Estimation Approach for a Focused Plenoptic Camera. In *Advances in Visual Computing (Proc. ISVC 2015)*, volume 9475 of *Lecture Notes in Computer Science*, pages 70–80.

Wang TC, Efros AA, and Ramamoorthi R (2015). Occlusion-Aware Depth Estimation Using Light-Field Cameras. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3487–3495.

Wanner S, Fehr J, and Jähne B (2011). Generating EPI Representations of 4D Light Fields with a Single Lens Focused Plenoptic Camera. In *International Symposium on Visual Computing (ISVC)*.

Wanner S and Goldlücke B (2012). Globally Consistent Depth Labeling of 4D Lightfields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wanner S and Goldlücke B (2014). Variation Light Field Analysis fo Disparity Estimation and Super-Resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):606–619.

Wanner S, Meister S, and Goldlücke B (2013). Datasets and Benchmarks for Densely Sampled 4D Light Fields. In *Vision, Modelling and Visualization (VMV)*.

Whelan T, Johannsson H, Kaess M, Leonard JJ, and McDonald J (2013a). Robust Real-Time Visual Odometry for Dense RGB-D Mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5724–5731.

Whelan T, Kaess M, Fallon M, Johannsson H, Leonard J, and McDonald J (2012). Kintinuous: Spatial Extended KinectFusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*.

Whelan T, Kaess M, Leonard JJ, and McDonald J (2013b). Deformation-based Loop Closure for Large Scale Dense RGB-D SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 548–555.

Whelan T, Leutenegger S, Moreno RS, Glocker B, and Davison AJ (2015). ElasticFusion: Dense SLAM without a Pose Graph. In *Robotics: Science and Systems*.

Wu G, Masia B, Jarabo A, Zhang Y, Wang L, Dai Q, Chai T, and Liu Y (2017). Light Field Image Processing: An Overview. *IEEE Journal of Selected Topics in Signal Processing*.

Yu J, McMillan L, and Gortler SJ (2004). Surface Camera (SCAM) Light Field Rendering. *International Journal of Image and Graphics*, 4(4):605–625.

Yu Z, Guo X, Ling H, Lumsdaine A, and Yu J (2013). Line Assisted Light Field Triangulation and Stereo Matching. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2792–2799.

Yu Z, Yu J, Lumsdaine A, and Georgiev T (2012). An Analysis of Color Demosaicing in Plenoptic Cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 901–908.

Yucer K, Kim C, Sorkine-Hornung A, and Sorkine-Hornung O (2016). Depth from Gradients in Dense Light Fields for Object Reconstruction. In *4th International Conference on 3D Vision (3DV)*, pages 249–257.

Zeller N, Noury CA, Quint F, Teulière C, Stilla U, and Dhôme M (2016a). Metric Calibration of a Focused Plenoptic Camera based on a 3D Calibration Target. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences (Proc. ISPRS Congress 2016)*, III-3:449–456.

Zeller N, Quint F, and Guan L (2014a). Hinderniserkennung mit Microsoft Kinect. In *34. Wissenschaftlich-Technische Jahrestagung der DGPF (DGPF Tagungsband 23 / 2014)*.

Zeller N, Quint F, and Guan L (2014b). Kinect based 3D Scene Reconstruction. In *International Conference on Computer Graphics, Visualization and Computer Vision (WSCG)*, volume 22, pages 73–81.

Zeller N, Quint F, and Stilla U (2014c). Applying a Traditional Calibration Method to a Focused Plenoptic Camera. In *BW-CAR Symposium on Information and Communication Systems (SInCom)*.

Zeller N, Quint F, and Stilla U (2014d). Calibration and Accuracy Analysis of a Focused Plenoptic Camera. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences (Proc. PCV 2014)*, II-3:205–212.

Zeller N, Quint F, and Stilla U (2014e). Kalibrierung und Genauigkeitsuntersuchung einer fokussierten plenoptischen Kamera. In *34. Wissenschaftlich-Technische Jahrestagung der DGPF (DGPF Tagungsband 23 / 2014)*.

Zeller N, Quint F, and Stilla U (2015a). Establishing a Probabilistic Depth Map from Focused Plenoptic Cameras. In *International Conference on 3D Vision (3DV)*, pages 91–99.

Zeller N, Quint F, and Stilla U (2015b). Filtering Probabilistic Depth Maps Received from a Focused Plenoptic Camera. In *BW-CAR Symposium on Information and Communication Systems (SInCom)*.

Zeller N, Quint F, and Stilla U (2015c). Narrow Field-of-View Visual Odometry based on a Focused Plenoptic Camera. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences (Proc. PIA 2015)*, II-3/W4:285–292.

Zeller N, Quint F, and Stilla U (2016b). Depth Estimation and Camera Calibration of a Focused Plenoptic Camera for Visual Odometry. *ISPRS Journal of Photogrammetry and Remote Sensing*, 118:83 – 100.

Zeller N, Quint F, and Stilla U (2017). From the Calibration of a Light-Field Camera to Direct Plenoptic Odometry. *IEEE Journal of Selected Topics in Signal Processing (JSTSP)*, 11(7):1004–1019.

Zeller N, Quint F, and Stilla U (2018). Scale-Awareness of Light Field Camera based Visual Odometry. In *European Conference on Computer Vision (ECCV)*, pages 715–730.

Zeller N, Quint F, Sütterlin M, and Stilla U (2016c). Investigating Mathematical Models for Focused Plenoptic Cameras. In *International Symposium on Electronics and Telecommunications (ISETC)*, volume 12, pages 301–304.

Zeller N, Quint F, Zangl C, and Stilla U (2014f). Edge Segmentation in Images of a Focused Plenotic Camera. In *International Symposium on Electronics and Telecommunications (ISETC)*, pages 269–272.

Zhang C, Ji Z, and Wang Q (2016). Decoding and Calibration Method on Focused Plenoptic Camera. *Computational Visual Media*, 2(1):57–69.

Zhang Y, Li Z, Yang W, Yu P, Lin H, and Yu J (2017). The Light Field 3D Scanner. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–9.

Acknowledgments

There are many people without whom the completion of this thesis would not have been possible, or at least would not have ended in such a success.

First of all there is Franz Quint. He was the one who offered me the opportunity to do this research at Karlsruhe University of Applied Sciences and was the best supervisor I could have ever imagined. Furthermore, he has been a great mentor to me ever since my second year at university. Therefore, I want to give my biggest thanks to him.

Uwe Stilla gave me the chance to become a doctoral candidate at Technische Universität München. Despite the distance between Karlsruhe and Munich, he always gave active input to my work and was always available for discussions. I am very grateful for his support.

I want to thank Daniel Cremers for agreeing to review this thesis. His feedback, coming from a leading expert in the field of visual odometry and SLAM, really means a lot to me.

I want to thank Ling Guan from Ryerson University, Toronto, for introducing me to the exciting field of computer vision.

At Karlsruhe University of Applied Sciences, I found perfect working conditions for my research. For this I especially want to thank Urban Brunner, who supplied me with any possible support. I want to thank Felix Wöhrle for designing and building multiple hardware prototypes for me.

I want to thank Wolfgang Proß and Gerhard Schäfer for numerous discussion and for convincing me to stay at Karlsruhe University of Applied Sciences for this thesis.

My former colleagues and the people from DocNet – I want to thank them for making my time as a doctoral candidate unforgettable.

I want to thank Mareike Jäntsch for carefully proofreading this entire thesis.

Last but not least, I want to thank my mother and my entire family for their patience and for always supporting and believing in me. A very special thanks goes to my wife Tanja for accompanying me through all ups and downs over the last fourteen years and to my little daughter Miriam for reminding me that there will always be something more important than work.

Curriculum Vitae

Name	Niclas Zeller
Date of Birth	June 7, 1987
Place of Birth	Stuttgart, Germany
Place of Residency	Karlsruhe, Germany

Education/Employment

2003 – 2006	Apprenticeship as Micro Technologist Specialization in Semiconductors AIM Infrarot Module GmbH, Heilbronn
2003 – 2006	A-Levels (extra-occupational) Wilhelm Maybach School, Heilbronn
2006 – 2007	Employee at AIM Infrarot Module GmbH, Heilbronn
2007 – 2011	Bachelor of Engineering in Electrical Engineering Specialization in Communication and Information Technology Karlsruhe University of Applied Sciences
2011 – 2013	Master of Engineering in Electrical and Computer Engineering Ryerson University, Toronto
2011 – 2013	Master of Engineering in Electrical Engineering Specialization in Communication and Information Sciences Karlsruhe University of Applied Sciences
2012 – 2018	Research Assistant at Karlsruhe University of Applied Sciences Faculty of Electrical Engineering and Information Technology / Institute of Applied Research (IAF)
2013 – 2018	Doctoral Candidate at Technische Universität München Department of Photogrammetry and Remote Sensing