Dissertation
submitted
to the
Combined Faculty for the Natural Sciences and
Mathematics
of
Heidelberg University, Germany
for the degree of
Doctor of Natural Sciences

put forward by
Dipl.-Inf. Hendrik Schilling
born in Bergisch-Gladbach, Germany
Date of oral exam: 15.02.2022

Light Field Camera Arrays Calibration and Depth Estimation

Advisor: Prof. Dr. Bernd Jähne
Second Referee: Prof. Dr. Filip Sadlo

Light Field Camera Arrays Calibration and Depth Estimation

Advisor: Prof. Dr. Bernd Jähne

Abstract

Scene reconstruction from camera images is a challenging task which benefits many applications, from visual effects to virtual and augmented reality applications (VR/AR) over industrial quality inspection to robotics and autonomous driving.

This thesis is composed of four contributions to the field of passive 3D scene reconstruction: a) A fractal, passive, self-identifiying calibration target which provides a high number of calibration points independent of magnification. b) A general, ray-based calibration approach which allows highly accurate calibration of central and non-central single and multi-camera setups, from a passive calibration target. The approach is capable of estimating the target imperfections which enables fabrication of the calibration target via simple printing, where previously an active target had to be employed. c) A light field depth estimation method, which exploits the rich constraints from the many views of a light field capture to improve depth estimation. To this end an improved occlusion model is introduced and the resultant error terms are locally optimized using a fast patch-match based optimization scheme. The accuracy of this approach improves upon the previous state-of-the-art, as demonstrated in a multi-metric light field evaluation benchmark. d) Finally, the light field depth estimation approach is extended to exploit polarization cues captured with a light field polarization camera, which improves the reconstruction quality on smooth, glossy surfaces.

Zusammenfassung

Szenenrekonstruktion aus Kamerabildern ist eine zentral Aufgabe von der viele Anwendungsgebiete profitieren. Von Visual Effects über Virtual und Augmented Reality (VR/AR) über Qualitätskontrolle zu Robotik und selbstfahrenden Autos.

Die Vorliegende Arbeit besteht aus vier Beiträgen zum Gebiet der passiven 3D Szenenrekonstruktion: a) Einem passiven, selbstidentifizierenden fraktalen Kalibrier-Ziel, welches ein hohe Anzahl von Kalibrationspunkten unabhänig von der Vergrößerung bereitstellt. b) Einem generischen, strahlenbasiertem Kalibrationsverfahren, welches präzise Kalibration für zentrale und nicht-zentrale Einzel- und Mehrkameraaufbauten ermöglicht. Das Verfahren ermöglicht die Berechnung der Verformungen des Kalibrierziels, was es erlaubt einfache gedruckte Ziele zu verwenden, anstelle der vorher benötigten aktiven Kalibrierziele. c) Eine Methode zur Tiefenberechnung aus Lichtfelddaten, welche die vielen Sichtpunkte der Lichtfelddaten ausnutzt um die Genauigkeit der Tiefenberechnung zu steigern. Zu diesem Zweck findet ein verbessertes Verdeckungsmodell Verwendung. Die Resultierenden Fehlerterme werden durch einen schnellen Patch-Match basierten Ansatz optimiert. Die Genauigkeit wird gegenüber dem vorherhigen Stand der Technik verbessert, was anhand eines Benchmarkdatensatzes demonstriert wird. d) Außerdem wird die Tiefenberechnung um die Möglichkeit erweitert Polarisationsinformationen auszunutzen, was die Genauigkeit im Bereich glatter, glänzender Oberflächen verbessert.

Acknowledgements

This work would not exist without the many great people that have supported me in the past four years.

My special thanks go to Bernd Jähne who made this thesis possible and to the Stuttgart Technology Center of Sony Europe Limited who financed it. I especially want to thank Alexander Gatto from the Computational Imaging Group for the extensive collaboration and support. My thanks also go to Filip Sadlo, for agreeing to be my second referee.

Furthermore, I want to thank all the great people who supported and endured me, or whom I had the pleasure of working with, during the last years: Kathi, Lara, Lukas, Merlin, Janko, Fanni, Marcel, Karsten, Alexander, Kathrin, Oliver, Hamsa, as well as Carsten and everyone from the VLL Group.

Special mention also to the people who helped proofread my thesis: Kathrin, Karsten, Marcel and Alexander.

Contents

1 Introduction

1.1 Motivation

Scene reconstruction from camera images is useful for a wide range of applications, from visual effects to virtual and augmented reality applications (VR/AR) over industrial quality inspection to robotics and autonomous driving. Light field imaging is one of the most promising approaches in the range of passive measurement techniques, providing very high accuracy compared to for example stereo imaging. In addition, light field imaging allows the extraction of additional scene properties such as reflectance parameters, which is not directly possible for other methods.

To make light field imaging usable in real world scenarios, the camera calibration, which describes the relation between the images processed by the scene reconstruction and the real world geometry, has to be of equally high accuracy as the light field processing. Hence, highly accurate, but still easy to use calibration approaches are required.

1.1.1 Calibration

Common methods for calibrating camera systems utilize passive targets with checkerboard patterns, which can be fabricated by printing. Such targets are easy to detect by a large range of available software, like the OpenCV library [13].

In computer vision tasks, it is already possible to locate image features with an accuracy down to a few hundreds of a pixel, for example in light-field measurements [23]. However, the camera calibration, critical in describing the relation between the world and the observed features, commonly achieves an accuracy not much below a tenth of a pixel, also visible in the evaluation, see section 3.4.4. Thus, a lot of the accuracy cannot be exploited due to the lack of a suitable calibration.

Research in geometric camera calibration dates back to the photogrammetry community, see *e.g.* [14, 16, 26]. Over the decades much research has been

conducted, improving both the models used to describe the image formation, and the methods available to estimate the parameters of these models.

Due to its simplicity in usage, the current de facto standard for camera calibration is Zhang's method [106], implemented famously in the Camera Calibration Toolbox by Jean-Yves Bouguet [12] and also ported to OpenCV [13] as well as many other libraries. For this approach the user takes images of a planar calibration target from several viewpoints. From the images, well-known target locations are detected, and the parametric calibration model is estimated from those target-to-image matches via a least-squares approach which simultaneously estimates the camera extrinsic and intrinsic parameters.

While this approach is both robust and simple to use, it comes with some limitations. Firstly, imperfections in the target reduce accuracy, and should be corrected when estimating the calibration parameters [3, 8, 91]. The wide range of lens designs and associated distortions may require secondary and specialized distortion terms [50, 83]. And finally, extreme lens designs, such as ultra-wide, fisheye, omnidirectional and catadioptric lenses require a non-central formulation where the camera does not actually possess a single center of projection [61].

All of these limitations have been addressed in the past using specialized models. However, finding the right parametric model for a given camera/lens combination is difficult, and eventually it is difficult to assess whether a limited accuracy is caused by an imperfect calibration model, inaccurately fabricated target or limited localization accuracy. Indeed, even estimating the accuracy of the calibration requires extra constraints in the calibration setup, like external reference points, or a stereo setup [8].

The general camera model, also called unconstrained calibration, as described by Grossberg and Nayar [40], models a camera as a collection of 3D rays, called raxels by them, where each 3D ray maps to a 2D point on the camera sensor. This formulation can describe any of the effects mentioned above. Hence, a single model can be used for any type of camera, be it central or non-central. In addition, even central, high quality lenses may be limited by the parametric model as shown by Bergamasco *et al.* [10], in which case usage of the ray based model can increase accuracy.

However, from a practical perspective, ray based calibration methods come with one crucial disadvantage: With current methods, deriving the general camera model is much more involved, compared to the simple planar target based methods described above. While Bergamasco *et al.* [10] make a good case for improving calibration accuracy using the ray based model, they require a calibration setup with an active target. Such a requirement severely limits the usability of the ray based model. Hence, in practice such approaches are only used in extreme cases,

where the classic approaches fail to deliver acceptable results. In these cases this work provides an approach which offers the accuracy of the ray based method, but without the considerable effort and the limitations imposed by active calibration targets.

1.1.2 Light Field Depth Estimation

Light-field imaging allows highly accurate depth estimation, by sampling a scene from many viewpoints. The oversampling increases accuracy and the high number of viewpoints reduce the chance of encountering a sample which is occluded in all other views.

This means that compared to the otherwise similar problems of stereo depth and optical flow, occlusion handling has a much bigger impact in light-field depth estimation. An inaccurate occlusion model will limit reconstruction, as foreground and background samples are confused within the data term around object boundaries.

The importance of occlusion handling is well known, and all state-of-the-art methods for light-field depth estimation implement some form of explicit occlusion handling. However, previous methods implement occlusion handling as a problem independent of the optimization step, often discarding a lot of data from the input light field, and making no effort to enforce consistency between the reconstructed depth and the used occlusion model.

Hence, it would be beneficial to merge the previously separated occlusion handling into the optimization. While this makes cost-volume based optimization unfeasible, a randomized solver based on PatchMatch Belief Propagation (PatchMatch Belief Propagation (PMBP)) [73] can effectively solve such a model.

Thus, the first goal for this work is to provide a calibration approach which is both very accurate, and easy to use, using a novel fractal calibration target, as well as a generic ray based camera calibration.

The second goal is the design of a new depth estimation method for light field imaging, which makes better use of available light field data, by improving occlusion handling which leads to improved estimation accuracy. Finally, this method is extended to light field polarization imaging, which improves depth estimation on specular unstructured surfaces.

1.2 How to read this Thesis

This thesis makes extensive use of internal references, which can directly be followed when using the PDF version of this document. These references link the different sections of this document, to support non-linear reading. In addition, several glossaries are provided, a terminology glossary, which contains terminology and abbreviations used throughout this thesis, a list of symbols, which is also directly accessible by clicking on any math symbol when using the PDF version of this thesis, as well as a list of figures, list of tables and the bibliography.

The structure of the thesis consists of an introduction in chapter 1, which details the motivation and the contribution of this work. It also lists the related work in section 1.3. The final chapter lists the conclusion in chapter 6 and gives a summary and an outlook. The technical introductions appear at the beginning of the respective topical chapters.

The rest of the work is structured around the two main topics: Light field depth estimation, and camera calibration. In chapter 2 basic concepts and definitions of camera geometry and calibration are introduced. On this basis chapter 3 introduces the novel fractal calibration target, and an evaluation which compares it against the classic checkerboard target. Based on the fractal target is the novel ray based calibration approach presented in chapter 4, including an evaluation comparing against parametric calibration approaches. Chapter 5 is concerned with the novel depth estimation method build around an improved occlusion handling approach, and extensions to the method based on BRDF estimation and polarization imaging in section 5.7. Finally, chapter 6 gives a summary of the thesis and points out possible future works.

1.3 Related Work

The following details the related work grouped into four categories: Calibration target, ray based camera calibration, light field depth estimation and polarization imaging. Note that this section is concerned with the closely related works and the current state-of-the-art, for more background information please refer to the introduction into camera calibration in chapter 2, as well as to the introduction to light fields in chapter 5.

1.3.1 Passive Planar Calibration Targets

For passive calibration targets the related work can be grouped in two categories. First calibration point localization, which is concerned with high accuracy localization of feature points from images. And secondly identification, which is concerned with the overall design of a calibration target, so feature points can be matched between multiple images.

Localization For passive targets there are two main types of localization features: Checkerboard corners and circular markers.

For the checkerboard localization three main principles have been established. The first method uses line intersection, where lines are refined individually and the intersection of two lines defines the calibration points [4]. The second approach uses functional descriptions of saddle points [55, 59, 67], where the local neighborhood is approximated by a 2D polynomial in which the corner point can be derived analytically. In comparison to line intersection, the saddle point method is more suited to smaller neighborhoods and at higher resolutions achieves either no improvements [67] or even worse results [59].

The third method makes use of orthogonal gradients, as implemented for example by OpenCV [13]. It exploits the property that the vector from the desired corner point to any edge pixel is orthogonal to the gradient vector in this pixel. This provides better results compared to line intersection as shown by Atcheson et al. [4]. Note that line intersection additionally suffers from distortion bias [59].

For circular markers the modeling of the transformation from a perfect circle to the observed image plays the largest role. Mallon *et al.* [59] evaluate localization performance on real and simulated images, under blur, noise, perspective and distortion. They conclude that circular patterns are dominated by perspective and distortion bias and recommend, for circular patterns, the usage of small (10px) circles and the correction according to the distortion model. Consequently, Datta *et al.* [22] as well as Douxchamps and Chihara [24] use an iterative refinement process which alternates between updating the camera model and refinement of the calibration marker image location. While this gives high quality results this approach inherently links the camera calibration with the marker refinement which makes the process slow and inflexible, when compared to a two-step approach where marker refinement and camera calibration are separate.

Identification Calibration targets combine feature points, as described above, with a specific structure or code-pattern, to enable automatic matching of feature points between different images of the same target.

Before being used for camera calibration, self-identifying planar markers were common in augmented reality applications, see Zhang et al. [105] for a comparative study. More recent examples of this are the works of Sattar et al. [75] and Bergamasco et al. [9].

Based on these ideas Fiala [28, 29] and Fiala and Shu [30] proposed the ARTag marker system, explicitly for camera calibration, which is based on square markers with a binary payload for identification. The binary code makes use of forward error correction, to improve robustness and to uniquely identify the otherwise ambiguous orientation, while localization is provided by line intersections of the marker borders. The CALTag marker system by Atcheson et al. [4] improves on this by utilizing a checkerboard like structure, which improves localization accuracy as checkerboard corner locations can be localized using subpixel saddle point refinement. At the same time Olson [66] improve the robustness of the identification code of ARTag, by optimizing the coding scheme. Both approaches can also deal with limited occlusions. Garrido-Jurado et al. [34] also improve the robustness of the coding and add explicit occlusion handling, mostly targeted at AR applications.

Another approach by Daftry et al. [21] uses circular markers with an angular binary code, while da Camara Neto et al. [20] use a checkerboard pattern, where squares are colored to provide identification.

A possible alternative to these passive methods that should also be mentioned here are active targets, for example using fringe projection or active phase targets (LCD screens). While such approaches can achieve high accuracy they come with their own set of constraints, like requiring an extra device with control and synchronization. This work focuses on passive targets which can easily be fabricated by printing, and are therefore more practical outside of controlled lab setups, see also section 2.4.2.

1.3.2 Ray Based Camera Calibration

Camera calibration using a ray based model has a long history, going back to two plane calibration methods by Martin *et al.* [60]. Grossberg and Nayar [40] introduce the concept of the raxel which associates a ray with an image pixel and also describes the radiometric properties of the imaging system. They calibrate cameras using known positions of an active target.

Ramalingam et al. [69] and Sturm and Ramalingam [93] perform the calibration using a passive target observed from unknown positions and interpolate intermediate samples from the sparse target using local homographies. However, while the resultant model is useful to assess camera properties such as non-centrality, they do not provide a method for undistortion of this irregular, sparse ray model. In addition, the used targets make it difficult to obtain reference points close to the image borders.

A different approach is adopted by Miraldo *et al.* [63], who assume smoothness of the general camera model and describe it using radial basis functions. Their calibration setup requires the observation of points with known world coordinates.

Later methods concentrate on active targets to derive dense image-to-target correspondences, and achieve high accuracy [8, 65], but still require perfect target geometry. Bergamasco *et al.* [10] apply this approach to central cameras, and can show that a ray based, but central, model already improves accuracy, due to better modeling of distortions not adequately described by the parametric model. Our approach goes one step further by keeping the model fully unconstrained and abolishing the requirement for an active target.

All works mentioned above require perfect target geometry, up to the accuracy of the calibration model. The joint estimation of feature locations and camera parameters is common for structure from motion pipelines [82], where camera calibration is often performed without calibration targets. Yet all ray based calibration methods assume a perfectly planar target. However, highly accurate targets are expensive, especially in larger sizes, as well as still limited in accuracy [8, 91]. For central camera calibration, the advantage of estimating target deformations for approximately planar targets has been examined by Albarelli *et al.* [3] as well as by Strobl and Hirzinger [91], who also demonstrate improvements in calibration accuracy even for a highly planar metallic target.

1.3.3 Light Field Depth Estimation

In the following, existing methods for depth estimation are discussed, focusing the description on the occlusion handling.

Where the methods are also included in the quantitative evaluation, the abbreviation is noted in square brackets. Abbreviations are identical to the ones submitted by the respective authors to the 4D Lightfield Benchmark [47, 48] and all method results, including ours, can also be compared on the benchmark website [47].

A very simple method of estimating the depth from light field data is the structure tensor approach introduced to light field imaging by Wanner *et al.* [98],

which estimates local orientations in an epipolar plane image (EPI) from image derivatives. The raw results can then be optimized using various optimization approaches [97, 98, 100] and multi-orientation analysis can be applied to get some estimates for reflective or transparent surfaces [99].

Neri *et al.* [64, RM3DE] perform multi-resolution block matching, adapting the window size with some local gradient measure, and performing matching independently for different viewpoint directions from the center view. Occlusions are handled by using only the best match from the directional EPIs for the final median-filter based post-processing.

Lin *et al.* [53] build a focal stack from the light-field data, and exploit the symmetry around the true depth in the stack to provide depth estimates, which are then optimized in a cost volume. A heuristic is employed to generate a separate occlusion map which is used to switch to an alternate cost for occluded pixels prior to the cost volume optimization.

Strecke *et al.* [90, OFSY_330/DNR] extend on this idea by improving the occlusion handling using four partial focal stacks representing the four viewpoint directions of a cross-hair subset of the light field, and using only the minimal cost from the horizontal and vertical direction, which should be less affected by occlusions. The method is notable for the explicit optimization of surface normals in addition to depth, which improves the surface quality of the reconstruction.

Williem and Park [101] introduce two independent cost functions. Angular entropy, which is a correspondence cost based on the entropy of photo-consistency, and an adaptive defocus cost, both of which show some robustness against occlusion. Reconstruction is then based on cost-volume filtering with graph cut. In a later work they improve this method, [102, CAE] modifying both cost functions to further improve the robustness against occlusion.

The Spinning Parallelogram Operator by Zhang *et al.* [104, SPO] scans the depth volume with a histogram comparison operation, which compares the areas left and right of the EPI line, defined by the respective disparity. This histogram comparison is relatively robust to at least single occlusions, hence no extra occlusion handling is performed in the guided filter based cost volume processing of the local cost estimates. Sheng *et al.* [85, SPO-MO] expand on this approach and add explicit occlusion handling by regarding multi-orientation EPIs and selecting a single unoccluded one for the calculation of the cost volume, according to an occlusion heuristic.

All of these methods make use of some form of cost volume optimization [53, 85, 90, 101, 102, 104], if not using a simple filter based approach [64]. Occlusion handling is always separated from the cost volume optimization and comes in several variants: By using cost functions robust against occlusions [101, 102, 104],

by using the minimal cost from several EPI directions [64, 90] or by switching between separate cost functions for occluded/unoccluded samples [53].

The works focusing on cost functions robust to occlusions show an interesting pattern. While the original publications only use the proposed robust cost functions [101, 104], later works mainly focus on the occlusion handling either by further improving robustness against occlusion or by adding explicit occlusion handling [85, 90]. It seems that even though cost functions exist which show *some* robustness against occlusion, these cost functions do not return optimal results.

On the other hand, methods that handle occlusions by selecting the minimal cost from several, possibly partial EPIs, discard a lot of samples from the input light field. This reduces the number of samples over which the data cost can be calculated and hence reduces accuracy.

Common to all methods is the fact that the used occlusion information is independent of the final optimized depth estimate. The additional scene knowledge available after optimizing the depth model is not reflected by the used cost function, which is limited to the initial occlusion estimates. Our proposed method addresses this point by using the current model to calculate the occlusions inline, during the processing, and therefore improves the utilization of the available light-field data.

Note that there are other methods which optimize the occlusions, like the works by Wanner and Goldlücke [98, 100] where they filter local depth estimates with a model enforcing global consistency with respect to occlusion. However, the accuracy of this approach is limited by the fact that only local estimates are used as priors in a regularization approach, and no updates on the cost are performed for updates in the occlusion model.

Recently, deep learning based methods are also appearing, for example EPINet by Shin et al. [86]. While these approaches show very promising performance, competitive to the results shown in chapter 5, the lack of real world ground truth data mostly limit them to synthetic benchmarks. For evaluations see [47, 48] and the website [46] which is continuously updated.

1.3.4 Polarization Imaging

Polarization imaging makes use of polarization filters or sensors to get additional information about the scene geometry, mainly by regarding surface normal cues from polarization information. Note that polarization information only gives some constraints on the surface normal, hence for a full 3D reconstruction it always has to be combined with some additional estimation method.

Atkinson and Hancock [5] show that polarization information from diffuse illumination can be used to estimation surface normals of specular objects. Kadambi et al. [49] use polarization information, obtained from three images with a linear polarization filter to refine the depth model obtained from a Kinect camera.

Smith et al. [89] show that polarization information can be combined with shading constraints on uniformly colored objects, while Cui et al. [19] demonstrate that multi-view stereo can benefit from polarization constraints.

1.4 Contribution

The following lists the main contributions of this work, divided into the four topics.

Chapter 3 – Fractal Calibration Target A new passive, self-identifying, fractal calibration pattern is presented, which increases the number of calibration points available from passive calibration targets, and reduces the dependency from camera magnification. The target is robust against many image deteriorations, and improves calibration accuracy due to better coverage and the higher number of calibration points, as well as by reducing center bias. The target is also dense enough to provide the basis for the ray based calibration approach. This contribution has also been published [79, 88].

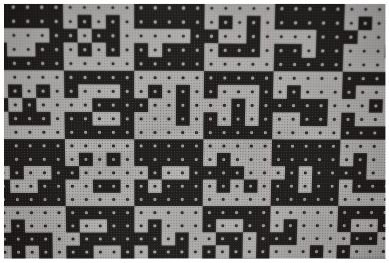


Figure 1.1: Fractal calibration target.

Chapter 4 – Ray Based Camera Calibration from an Imperfect Target To enable ray based calibration from the passive, sparse calibration target, a novel intermediate calibration proxy representation is introduced. It provides the target to image mappings at arbitrary image coordinates that are a requirement for ray based camera calibration. We demonstrate a new calibration method based on this calibration proxy, which estimates a ray based model from an imperfect target, using derivatives of the calibration proxy to derive an error measure in pixel coordinates, analogous to the reprojection error used in classic parametric

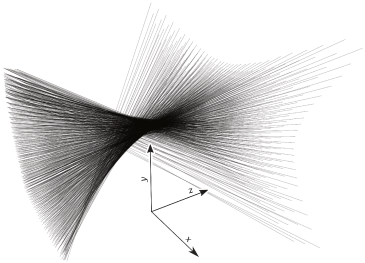


Figure 1.2: Ray visualization of a non-central camera.

calibration methods. In the evaluation we show the advantage of this method over the parametric modeling of the distortion, even for a relatively low distortion, central camera, demonstrating the limitations of the parametric calibration paradigm.

Chapter 5 – Light Field Depth estimation with Inline Occlusion Handling A novel approach to perform occlusion handling for light-field depth estimation is presented, which directly integrates occlusions into the depth model. Compared to all prior methods, this maximizes the use of the available data. Despite the complex occlusion model a PatchMatch [6] based scheme based on local updates is able to give good estimates on this model, and in competitive processing time. Although the method does not guarantee globally optimal solutions, it achieves state-of-the-art results in nine out of twelve error metrics, with a close tie for the remaining three, on the most prominent academic light field benchmark [46]. This contribution has also been published [80].

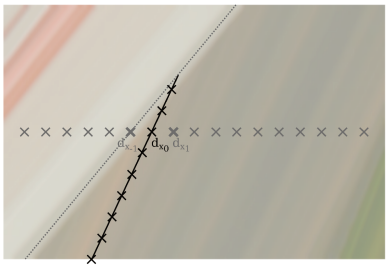


Figure 1.3: Epipolar plane image structure with occlusion.

Section 5.7 – Light Field Polarization Imaging Finally, a combination of polarization imaging with the light field depth estimation approach improves depth estimation for smooth specular surfaces, which violate the color constancy assumption, improving reconstruction accuracy in these areas.

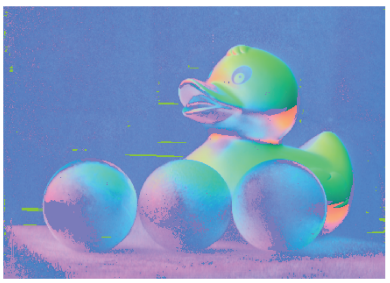


Figure 1.4: Surface normals from polarization imaging.

2

The Camera Calibration Pipeline

Figure 2.1: Example of an industrial computer vision camera with a wide-angle lens.

This chapter gives an overview over the full geometric camera calibration pipeline, introduces common approaches used for individual steps, and explains their properties and key advantages and disadvantages.

This chapter also introduces the definitions and concepts which are the basis for the novel fractal calibration target described in chapter 3 as well as the novel ray based calibration method described in chapter 4.

Geometric camera calibration, also called camera resectioning in literature, describes the geometric properties of a camera-lens system, with respect to how the 3D world is projected onto the 2D camera sensor.

This means we are mainly interested in finding a description of the optical system in question, which provides a mapping from 3D world coordinates two 2D sensor coordinates, but disregarding the actual sensor response, which describes the intensity distribution that a projected 3D point would cause. Those properties

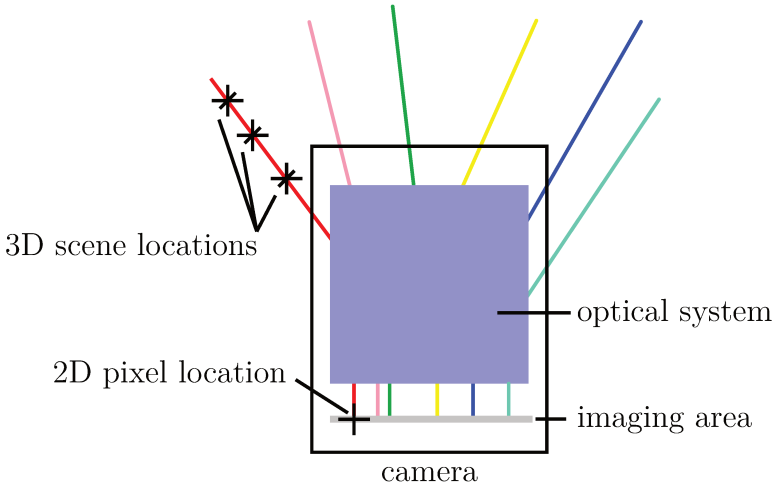


Figure 2.2: The black box camera model, as used in this work. The camera projects 3D scene coordinates into 2D image coordinates and is completely represented by the mapping between 3D rays and 2D pixels.

are described for example by the modulation transfer function [36], and the photometric camera response, and are not under consideration in this thesis.

2.1 Outline

This chapter will be structured as follows: The definition of a camera as it will be used throughout this thesis is introduced in section 2.2, followed by coordinate system conventions in section 2.2.1 and the ideal pinhole camera in section 2.2.2. Section 2.2.3 considers the deviations of real cameras from this ideal camera and models for these aberrations are shown in section 2.3. Finally, the calibration targets are introduced in section 2.4 which are a requirement for the actual derivation of the model parameters in section 2.5.

2.2 Camera

For the scope of this work we regard the camera as a black box system, see figure 2.2, composed of a two-dimensional imaging area and an optical system which projects points from a scene onto the imaging area. Figure 2.1 on the previous page shows a real world example of such a camera system. In this work

only the relation between the 3D world coordinates and the 2D sensor coordinates are relevant, and we regard the camera as a black box system. The mapping from world points to image coordinates is called projection. Camera calibration is both the name for the process of determining parameters of such a mapping, and for the result of this process.

In practice, the above definition is a bit too broad. In reality, we can limit ourselves to the cases where all world points which are projected onto the same image coordinate lie on a line in world space. This *general camera model* was first described by this name by Grossberg and Nayar [40].

2.2.1 Coordinate Systems

A calibrated camera describes a mapping from 3D world space to 2D image space. To allow reasoning about multiple cameras, which is important for light field imaging, it is convenient to split the projection into two parts, the intrinsic and the extrinsic parameters. For this we define the camera space, an additional 3D coordinate system per camera location, with the origin at the center of projection.

Extrinsics describe a transformation from the world space into a camera space, see figure 2.3 on the following page, while the intrinsics describe the actual projection from the camera space to image coordinates, see figure 2.4 on the next page. If the camera pose is changed, then only the extrinsics need to be adjusted, while the intrinsics stay constant. In effect this means that the extrinsics can be reduced to the pose of the camera, made up of the 3D rotation and 3D translation of the camera.

2.2.2 Pinhole Projection – The Ideal Camera

From the perspective of computer vision applications, it is desirable to work with images captured by an idealized pinhole camera. The projection of a pinhole camera model is described by the following equations, and visualized in figure 2.5 on page 17:

$$\hat{x} = C \cdot p \tag{2.1}$$

$$\hat{x} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}, \tag{2.2}$$

Where, p denotes a point in the 3D camera coordinate system, f is the focal length in pixel and c is the location of the camera center. $\hat{x}$ is an image

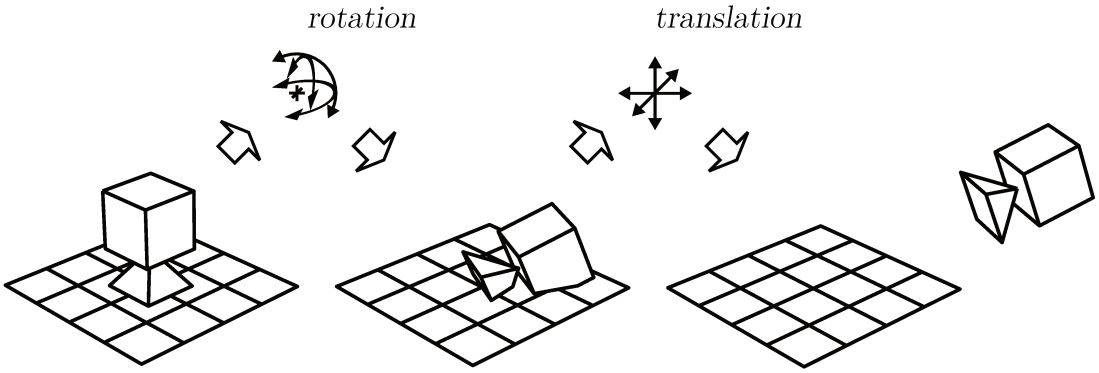


Figure 2.3: Visualization of the extrinsic transform which maps from scene to camera coordinates, applying first rotation then translation to a scene (point) to map it from scene to camera coordinates.

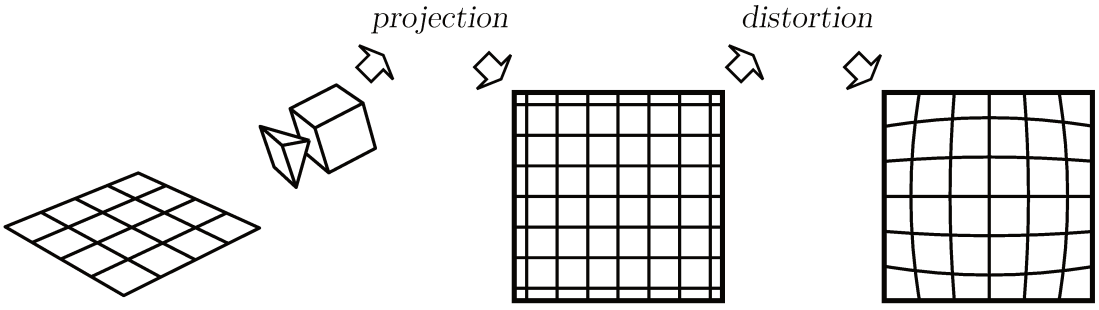


Figure 2.4: The intrinsics define a mapping between camera coordinates and image coordinates and hence both the projection from 3D to 2D, and the distortion of this mapping.

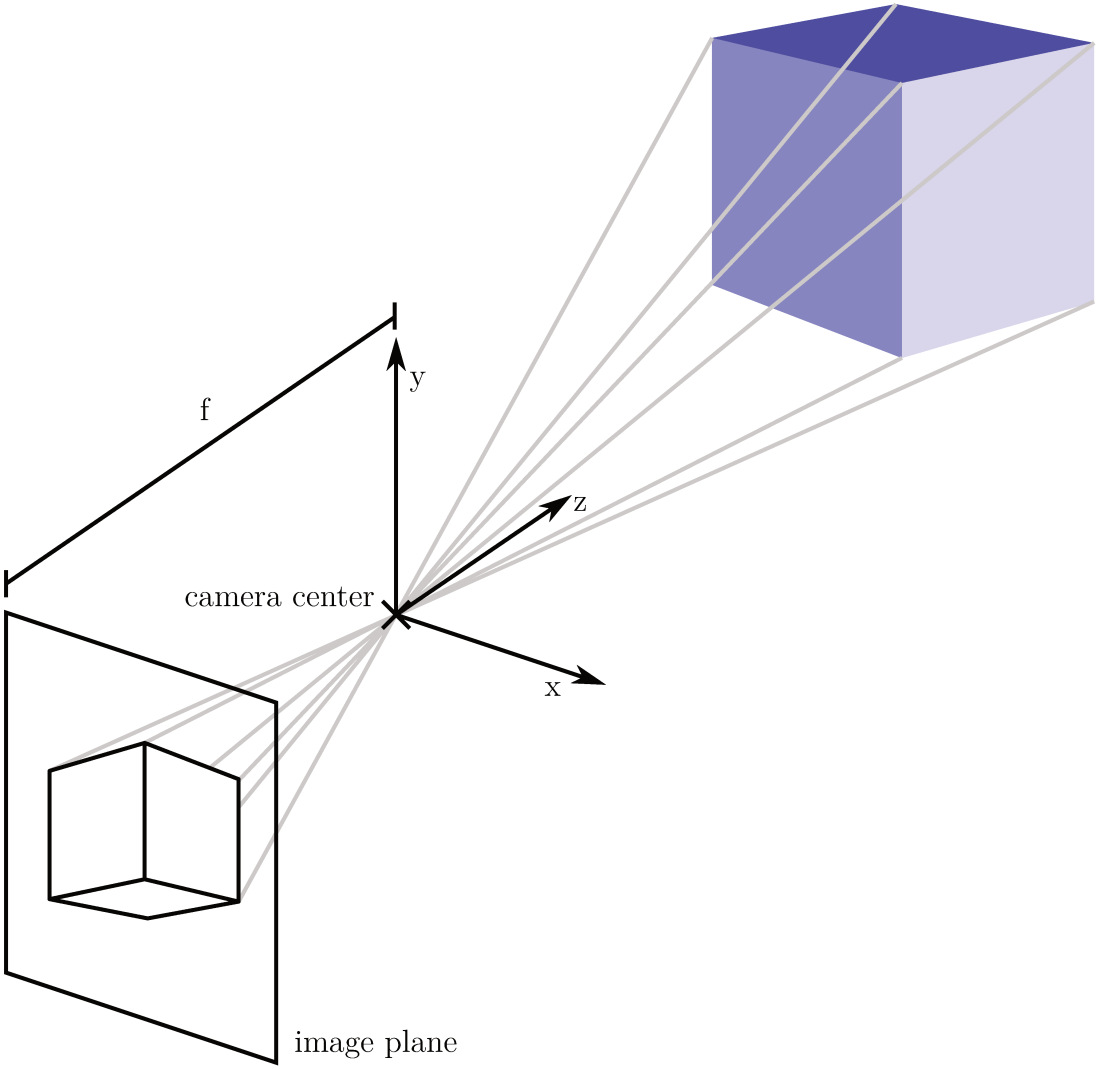


Figure 2.5: Visualization of the projection of the pinhole model. The image is projected by sending lines through the camera center (the pinhole), onto the imaging plane, as described by equation (2.1).

point in *homogeneous coordinates*. h converts from homogeneous to a euclidean coordinates:

$$x = h(\hat{x}) = \begin{pmatrix} \hat{x}_x/\hat{x}_z \\ \hat{x}_x/\hat{x}_z \end{pmatrix}, \tag{2.3}$$

This can be rewritten using *normalized camera coordinates* $\hat{p}$, by putting $\hat{p}$ from equation (2.1) in equation (2.3), as:

$$\hat{p} = h(p) \tag{2.4}$$

$$x = \begin{pmatrix} f_x p_x/p_z + c_x \\ f_y p_y/p_z + c_y \end{pmatrix} = \begin{pmatrix} f_x \hat{p}_x + c_x \\ f_y \hat{p}_y + c_y \end{pmatrix} \tag{2.5}$$

and hence:

$$\hat{x} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{p}_x \\ \hat{p}_y \\ 1 \end{pmatrix}, \tag{2.6}$$

Note that the above equation projects from the 3D camera coordinate system, using the normalized 2D image coordinates to get the final pixel coordinates. This form removes the depth information from p before projection with the camera, and therefor demonstrates that depth information is *not* relevant for the ideal pinhole camera model. For the ideal pinhole model equation (2.1) and equation (2.6) can be used interchangeably, but working in normalized camera coordinates is often preferred as it simplifies for example distortion handling, see section 2.3.

The camera extrinsics, which relate the camera to a common world or scene coordinate system, consist of the camera rotation and translation relative to the world coordinate system. This gives six degrees of freedom for each set of extrinsic parameters. A point w in scene coordinates is transformed to camera coordinates with:

$$p = Rw + T, \tag{2.7}$$

where R is a rotation matrix with three degrees of freedom, while T is a translation vector.

Hence, the full model for the pinhole projection, from scene coordinates to homogeneous image coordinates is:

$$\hat{x} = C(Rw + T). \tag{2.8}$$

Note that notations in literature may differ from this, specifically homogeneous coordinates are used more extensively for certain scenarios.

2.2.3 Aberrations – The Imperfect Camera

In reality, it is impossible to build a camera which perfectly follows the pinhole projection above, due to the physics of optics (and budget). In the following geometric deviations of real cameras from the ideal pinhole projection are described. Only geometric aberrations are listed, aberrations which change *e.g.* the effective resolution or the photometric response are out of the scope of this work.

Radial Distortion Radial distortions are most frequently encountered and most pronounced and appear as a change in apparent image magnification depending on the distance from the camera center. They are often described by their appearance. For example: barrel distortion or pincushion distortion.

Tangential Distortion Tangential distortions are normally much weaker and appear as an effect of decentering or imperfect alignment of lens elements in the optical system. When seen from image space, these distortions influence the projection depending on the angle (in image space) from the optical center.

Lateral Chromatic Aberrations If multiple wavelengths are observed through the same optic, then wavelength dependent characteristics of the optic become relevant as they may change magnification and other distortion characteristics, as a function of the wavelength. The effect is that the recorded color channels may not be aligned and need to be corrected independently. Not that this is only possible with a color camera. A grayscale camera will integrate over a range of wavelengths and lateral chromatic aberrations will be observed as a blur of the image.

Non-Central Cameras Real lenses have non-zero thickness which can lead to non-central characteristic, especially for extreme designs like fisheye lenses. This means that such lenses do not possess a single center of projection, but rather the apparent center of projection shifts depending on the angle of the incoming light. This property leads to distortions that appear depth-dependent, as explained in section 4.3, and hence are more difficult to correct.

Asymmetric Distortions Photographic lenses are made up of between only a few and up to dozens of individual lens elements, which are often difficult to align (or keep aligned) perfectly. Hence, the actual distortions may also possess asymmetric properties which can not be described by radial and tangential distortions. The magnitude of these secondary distortions is normally much lower, compared to the primarily radial and tangential distortion components. However, these secondary distortions may still be relevant for calibration, depending on the required accuracy.

2.3 Distortion Models

A range of models have been proposed in the past, whose power in correcting the aberrations listed above differs.

The most common approach is to regard the aberrations as simple image space distortions, which can be implemented directly in image space, and is commonly formulated using normalized image coordinates:

$$\tilde{p} = L(\hat{p}), \tag{2.9}$$

here $\tilde{p}$ are the distorted coordinates, derived from the ideal normalized image coordinates $\hat{p}$ using a distortion function L. The distorted coordinates are then projected as before (equation (2.1)):

$$\hat{x} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{p}_x \\ \tilde{p}_y \\ 1 \end{pmatrix}. \tag{2.10}$$

Therefore, the full model becomes:

$$\hat{x} = C \cdot L(h(Rw + T)), \tag{2.11}$$

which applies the following operations to a scene point w: First apply extrinsics (map point into camera coordinates), normalize, apply a 2D distortion and finally project using the camera matrix C.

Regarding the distortion model L, a lower number of parameters simplifies calibration because fewer parameters have to be estimated, which also reduces the chance of overfitting. Therefore, the most basic distortion models only correct radial distortions, where:

$$L = \hat{p} + K(r)\hat{p}. \tag{2.12}$$

The simplest distortion functions are low order polynomials, and euclidean distance for r, *e.g.*:

$$L_r = k_1 r + k_2 r^2 \tag{2.13}$$

$$r = \sqrt{\hat{p}_x^2 + \hat{p}_y^2} \tag{2.14}$$

A convenient optimization is to choose polynomials with terms of only even degree, which allows to avoid the calculation of the square root in r.

The next step are models which add secondary terms to correct for tangential distortions which depart from the perfectly radial symmetry of the purely radial models, justified by lens decentering and imperfect alignments of optics [15]:

$$L_t = \begin{pmatrix} L_r(r) \cdot \hat{p} + t_1(r^2 + 2\hat{p}_x^2) + 2t_2\hat{p}_x^2\hat{p}_y^2 \\ L_r(r) \cdot \hat{p} + 2t_1\hat{p}_x^2\hat{p}_y^2 + t_2(r^2 + 2\hat{p}_y^2) \end{pmatrix} \tag{2.15}$$

Indeed, these models (low order polynomial with or without tangential terms) are probably the most widely used in practice [35, 45, 77], as they provide a simple model which handles most distortions adequately. A range of additional models have been proposed, like the radial division and rational polynomial models [57, 72, 92], as well as alternatives for r, to better capture the strong distortion properties of *e.g.* fisheye lenses, see [50, 72, 92] for examples.

One variant extension is the division model implemented in OpenCV [13], which is not directly found in literature, but is a simple extension of the division models, and has wide a adoption due to the easy availability in the open source OpenCV library.

I believe that the main reason why no other models have gained wider usage is that the alternative models are mainly interesting for specific optics, and measuring the actual calibration performance using external means is quite hard, compare section 4.5, and hence often skipped even in calibration literature.

From a distortion function it is straightforward to create an undistortion mapping, as an image with the same dimensions as the (target) image, where each pixel stores the distorted coordinates. This table can directly be filled using the distortion function, as it maps from undistorted to distorted space.

Note that there is an alternative formulation of the distortion pipeline presented above, based on *undistortion* functions, see for example [18]. However, while such a formulation is sometimes convenient when directly working on distorted material, as the undistortion function is directly available, the creation of an undistortion *mapping* is more problematic as it requires the inverse of the distortion function,

which might not be available analytically. Also problematic is that the reprojection error is not directly available and hence other measures must be minimized (*e.g.* Sampson distance), which might not perform as well [27, 43].

All of these approaches above work well if the chosen distortion model is adequate for the actually observed lens distortions. However, in practice the correct model might not be known beforehand. In addition, if very high accuracy is required, then radial and tangential models can be insufficient as they only provide an approximation to the actually observed distortion [8, 10].

Therefore, this approach can be further extended with arbitrary 2D functions, which do not put any prior on symmetries in the distortion. Note that the models proposed up to here will be denoted as *parametric* calibration (models), in contrast, the following models, are conceptually more similar to a lookup-table, with support points between which the actual distortion is interpolated. Of course, in the end they also consist of a number of parameters, but this number is much higher (hundreds to thousands instead of under ten):

$$L(\hat{p}) = U(\hat{p}) \tag{2.16}$$

Where U is a function from $\mathbb{R}^2 \to \mathbb{R}^2$, which implements an arbitrary 2D distortion.

A limited version of this is to assume no specific model but assume purely radial distortion [44], thus calculating a lookup table for the distortion, while estimating the center of distortion and the focal length, however this does ignore tangential effects.

An example for such a mapping is the model by Grompone von Gioi et al. [39] which is based on a sparse mapping of correspondences between distorted and undistorted images, together with affine interpolation based on delaunay triangulation. Other approaches are based on thin plate splines [81], Bézier patches [37] or locally weighted homographies [70]. As shown in the respective works, these approaches can correct some distortions more accurate compared to a parametric approach.

All models described until now only regard deviations in the 2D image space. However, as described by Magill [58] and Fraser and Shortis [33], real cameras also can exhibit non-central characteristics, where the projection does not actually follow the perfect pinhole model, by effectively not possessing a single center of projection. Examples are, among others, fisheye optics and catadioptric and omnidirectional cameras [61, 76, 94]. In these cases the lines in 3D space which describe which points are mapped onto an image coordinate, do not intersect in a single point.

These non-central cameras possess, when compared with a pinhole camera, an effectively depth dependent distortion, please see section 4.3 for a derivation of this property.

Such a camera may be modeled similar to a pinhole camera by replacing the intrinsic camera matrix projection with a more generic function $C : \mathbb{R}^3 \to \mathbb{R}^2$ which maps from 3D camera space to 2D image space using a ray based model, see also section 2.2:

$$x = C(p) \tag{2.17}$$

For specific types of non-central optics, parametric models have been developed in a similar fashion to the polynomial and rational radial models [61, 94].

However, analogous to interpolation based generic 2D distortion models, which only use image space distortion, see section 2.3, it can be beneficial to not constrain the ray based camera model to a specific parametric model, but instead use a less constrained mapping. The most extreme approach is to assign an independent ray to every individual pixel on the imaging plane, which does not constrain the projection in any way, aside from assuming ray properties. Indeed, this is the approach normally chosen for ray based calibration, for example by Barreto et al. [7], Bergamasco et al. [8, 10], Grossberg and Nayar [40] and Nishimura et al. [65]. Note that these approaches all make use of an active target, as otherwise derivation of per pixel ray support points is not possible.

However, such a model is difficult to estimate, therefore the incorporation of additional constrains on the observed distortion, like a smoothness assumption [62], leads to simpler models, which nonetheless retain most of the desirable characteristics of not assuming some specific distortion model and not constraining the projection to a central pinhole projection.

Another possible constrain is the assumption of radial symmetry for the rays, which effectively leads to a 1D radial camera model [17], but this is less generic.

Please see chapter 4 for more in-depth explanations on how to handle and parametrize ray based calibration, as well as the novel calibration approach based on the fractal passive target introduced in chapter 3.

2.4 Calibration Targets

Calibration targets are physical objects with a well-known shape and appearance, which are placed in the scene. The target is observed with the cameras which should be calibrated. Targets are designed so individual feature points of the

target can be localized and identified which results in target coordinate to image coordinate matches. If enough matches are available from several different viewpoints of a target, then estimation of the calibration parameters becomes possible, see section 2.5.

2.4.1 Key Properties

To enable simple and high quality calibration, a calibration target should fulfill several key properties. For a comparison of different target types please see table 2.1 on page 26 as well as table 3.1 on page 40, which also compares the current state-of-the-art with the new target introduced in this work.

Self-Identifying Calibration points on the target must be distinguishable from the recorded images, else no mapping to target coordinates is possible.

High Localization Accuracy The accuracy of the calibration point localization determines the accuracy of the calibration.

Calibration Point Density More calibration points give the calibration method more data points to estimate the calibration parameters, which lead to more accurate and stable results and reduces the chance of overfitting. Note, that localization accuracy and the number of calibration points are of course somewhat exchangeable as more calibration points of a fixed accuracy also increase the calibration accuracy due to the law of big numbers. Interesting is of course not the physical density, but the effective density of calibration points in the recorded image, which is influenced by many factors.

Completeness It should be possible to get calibration points for the whole imaging area. If for example the corners are left out, then the accuracy of the calibration will suffer in the corners.

Low Bias Target localization should not be biased by *e.g.* illumination changes or perspective, as this bias would transfer to the calibrated parameters.

Fabrication and price The target should be relatively easy and cheap to fabricate, to ease adoption.

Usability Finally, a calibration method should be easy to use, to allow fast and reliable calibration under field conditions.

2.4.2 Target Types

The following gives an overview of available target types, and their main properties, a tabular overview of the properties listed in section 2.4.1 is given in table 2.1 on the next page.

Planar 2D Targets Planar 2D target are made up of specific patterns printed on a planar surface. The patterns provide identification and localization. Examples of such targets can be found in [4, 28, 29, 30, 34, 66]. The main advantage of planar targets is the ease of fabrication as well as high localization accuracy. The inherent disadvantage is that all previously available planar targets give a relatively sparse set of calibration points, which is not well suited for *e.g.* ray based calibration.

Active Targets Active targets utilize LCD screens [7, 8, 10, 40, 65], to change to shown calibration pattern over time. The screens show multiple patterns while the camera-screen pose stays unchanged, which allows very accurate localization of the target location of *every* pixel of the camera. Hence, density and accuracy are very good, however fabrication is much more difficult, which is relevant if large targets are to be used, because then a fitting screen might not be available (or very costly). Also, for high resolution cameras it might be difficult to find a correspondingly high resolution screen. Recording the required calibration data is much more involved compared to passive targets, as camera and screen need to be synchronized and more calibration images must be recorded.

3D Targets Instead of self-identification based on 2D patterns, it is also possible to use a 3D target made up of (colored) spheres, arranged within a 3D volume. This is advantageous especially for rapid multi-view/multi-camera calibration, as shown by Shen and Hornsey [84], where the same target can be observed from all directions at the same time. However, the density of calibration points is much lower which limits the maximum accuracy. Non-planar calibration targets are more common in the photogrammetry community [32, 56, 71], and often called test fields. While they are more difficult to construct they can require fewer calibration images, as scale constraints are already embedded in the 3D target.

	passive planar	active planar	3D targets
identification	yes	yes	limited[1]
marker accuracy	high	high	high
effective density	medium	high	low
completeness	high	high	low
bias	low	low	low
fabrication	simple	very involved[2]	medium
price	very cheap	expensive	cheap
usability	high	involved[3]	very high

[1] Colored sphere can be used, but only with color cameras, and only a limited number of colors can be distinguished.
[2] pretty much impossible if no off-the-shelf target (LCD screen) is available with the required specifications.
[3] normally controlled movements with a translation stage as well as multiple shots of the same screen are required

Table 2.1: Comparison of calibration target types from section 2.4.2 according to the key properties listed in section 2.4.1.

2.5 Model Parameter Estimation

Performing a camera calibration means to actually estimate the parameters of a camera model from some input. A calibration method commonly describes the combination of a target type, together with a distortion model and an algorithm which can estimate the model parameters from observations of the calibration target.

Most commonly used are iterative non-linear least-squares approaches, to find a parameter set $\hat{a}$ which minimizes the reprojection error of a model m:

$$\hat{a} = \arg\min_{a} \sum_{j} (i_j - m(t_j, a))^2, \tag{2.18}$$

where a are the parameters of the model (intrinsics and extrinsics) and the t_i are 3D locations on the calibration target, known from the self-identifying patterns on the target, while i_j are the respective image space locations where these calibration points where observed on the target. The camera model m describes the projection from scene coordinates into image space, as described by equation (2.11).

As the reprojection error is a measure in image space, it is directly meaningful, and problems due to heteroscedasticity, because of *e.g.* varying magnification of observed features are avoided. Therefore, the reprojection error is also considered the gold standard for calibration error metrics [27]. Other metrics, which have to be used for example for calibration of undistortion models may be biased for example towards more distant calibration points, which can reduce the accuracy of the calibration [8]. As the internal structure of the minimized least-squares problem is sparse, efficient solvers are available today [2], which make fasts estimation possible. These solvers rely on derivatives of the objective function, which is provided by automatic differentiation [38, 51, 68].

This approach can also be extended to unknown target geometries, where only correspondences between two or more images are known, but not the true position of the correspondence in world space. With enough views and correspondences this problem is still well-defined and the 3D world locations of the correspondences are simply more unknown parameters estimated during the least-squares optimization. Indeed, in such a configuration the challenge is not the estimation of the calibration parameters, but the identification of correct matches from a set which contain false correspondences [82], often using RANSAC [31], as well as finding a good initialization. However, usage of well-defined physical calibration targets can still provide a higher accuracy, as well as more robust matching.

This approach does not directly work for the more complex ray based calibration models. The reason is that for ray based calibration there is no explicit projection available, and hence it is not possible to directly calculate a reprojection error. In addition, use of a passive target requires interpolation or regularization in the ray space, as individual subpixel correspondences from the calibration target are very unlikely to coincide with a ray pixel location (when the model is made up of one ray per image pixel).

Therefore, most ray based calibration approaches make use of an active target [8, 10, 40, 65], like a large LCD screen, see also section 2.4.2, together with a high accuracy translation stage. Active targets provide a very dense pixel to target mapping, but again no projection and therefore no reprojection error is available. Target point to ray distance is therefore often used, which can only approximate the performance of the reprojection error [27].

Another approach is to use a passive target and to interpolate the target location to image mappings, to provide calibration points at arbitrary pixel positions, which again allows independent estimation of individual rays [69, 93]. The novel calibration method introduced in chapter 4, for ray based calibration from passive targets belongs to this last family, but overcomes some limitations mentioned here, by utilizing the intermediate representation of the *calibration proxy*. This

structure implements the interpolation necessary for ray based calibration from a passive (and hence sparse) target, and enables the derivation of a reprojection error for ray based calibration. Please see section 4.4.2 for the details.

3

High Density Passive Fractal Calibration Target

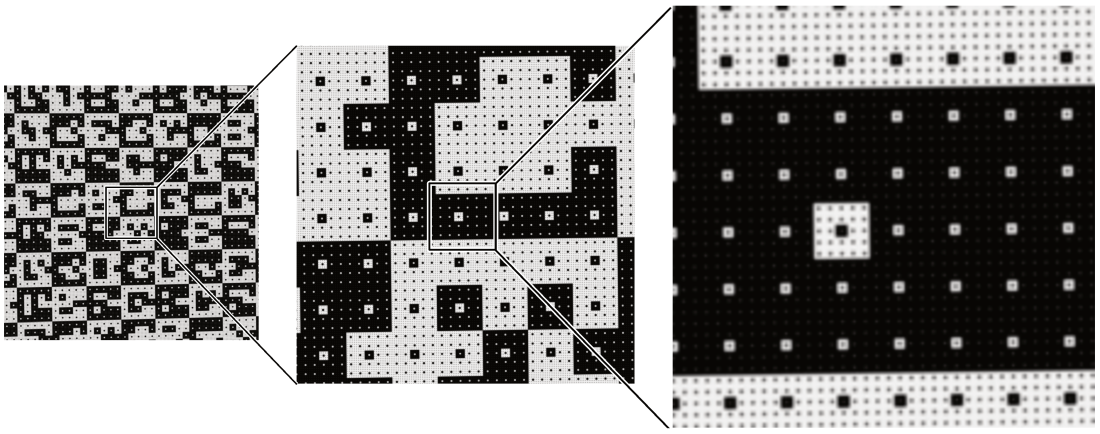


Figure 3.1: A view of the fractal calibration target. From left to right each step reduces the width of the cut-out to a fifth. Note how individual calibration dots resolve to square features when increasing the magnification.

This chapter introduces a novel, passive, fractal calibration target, see figure 3.1. This target increases the density of calibration points, enabling higher accuracy calibration. Thanks to its fractal nature the target can provide a high number of calibration points independent, within bounds, of the camera magnification. When coupled with a classic parametric calibration scheme, the target alone can outperform the accuracy of a calibration derived from the conventional checkerboard target, as shown in section 3.4.4. In addition, the density is high enough to provide the basis for the novel ray based calibration scheme introduced in chapter 4. The target was also published under an open source license [78, 79, 88] to ease adoption.

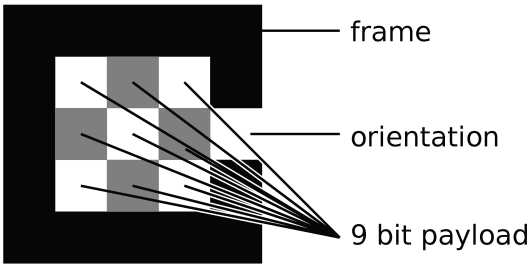


Figure 3.2: Schematic of a single marker. The constant color border with the hole at one side allows detection and fixes the orientation. A single marker contains a payload of 9 bits which is used for identification. .

3.1 Outline

The target introduced in this chapter relies on a self-identifying fiducial marker pattern, combined with a fractal structure, where many calibration points are recursively inserted at several scales. This keeps the number of calibration points within a fixed bound, independent of the magnification factor. The reasoning and design of the fractal structure is discussed in section 3.2. For related work please refer to section 1.3.1

Markers are identified using a 3×3 binary pattern, see figure 3.2, where a black bordered and a white bordered marker are always processed together. This effectively results in 18 bits of payload, which are used to encode the 2D marker position on a regular grid.

Detection starts with the identification of the large scale markers, see section 3.3.1, combining checkerboard corner detection and the payload for identification, as well as error detection by correlating neighboring marker candidates. In a second step the recursive dots are detected using a recursive strategy which fits 2D Gaussian distributions onto dot candidates, compare figure 3.3 on page 32, whose positions are estimated from the previous scale. The recursive refinement scheme is described in section 3.2.3 and section 3.3.2.

As is visible in figure 3.1 on the previous page, the recursion scheme operates with powers of five. The minimum size for successful fitting is around four pixels, which means that for high quality images we get one calibration point for every 4–20 pixels in each dimension, or between 2500 and 62500 calibration points for each megapixel of image data.

While the individual calibration points are not necessarily detected with a higher accuracy compared to other calibration targets, the high number of calibration points leads to better calibration results as errors are averaged out. Compared

to the OpenCV subpixel refinement we can increase the accuracy of the camera calibration by one order of magnitude, when comparing the calibrated projection against the ground truth projection, as shown in section 3.2.2.

3.2 Design

Regarding the marker size, there are two possible directions. Either few, large, individually highly accurate markers, as used by Douxchamps and Chihara [24], or many small markers of individually low accuracy.

3.2.1 Why Marker Size does not Matter

It can be argued that the underlying measurement accuracy for large and small markers is similar, as it depends solely on the quantity and magnitude of image gradients. A large marker could be regarded as a collection of smaller ones, which implies that they should achieve similar accuracy. Indeed, the subpixel marker locations are constrained by the individual image gradients. If the same quantity of gradients is used, where each gradient has the same quality (signal-to-noise ratio), then it does not really matter whether those gradients are spread over few, large markers or several smaller ones. However, the size of the individual markers still has practical implications on the way detection can be performed and where markers can be placed. Note that this underlying accuracy is not reflected by the root mean square error (RMS) scores of calibration methods, as those only give the error between the fitted model and the detected markers. Larger markers have already averaged out more of the gradient noise due to the larger marker sizes, where small markers provide a higher ratio of data samples to parameters in the overconstrained minimization problem of the calibration method, which will also lead to a reduction of the influence of gradient noise.

3.2.2 Why Marker Size Matters – And smaller Markers are Better

The localization of larger markers is dominated by perspective and radial distortion [59], which requires a more complex localization approach for accurate results. While perspective bias is easy to compensate for, even without doing a full calibration, to correct distortion bias, the marker localization has to incorporate the radial distortion [22, 24, 59], which ties marker refinement directly into the calibration. Thus, it is impossible to separate marker detection and camera

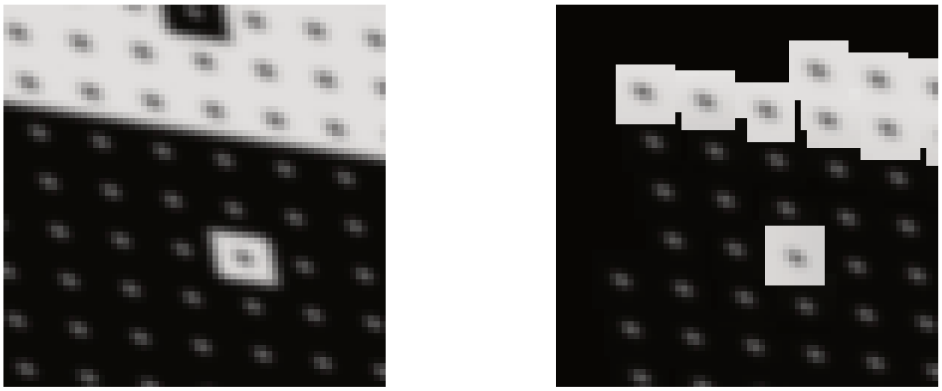


Figure 3.3: An example of the detection performance with small calibration points at the image border. On the left the original input image, on the right the fitted calibration points. Note how the small calibration points can be detected less than 10 pixels from the image border.

calibration, which also makes it difficult to adapt new calibration methods to such a combined calibration scheme. This means that small features are better suited for a generic calibration pattern, as the individual calibration features are not affected much by bias.

In section 3.2.1 we argue that the underlying accuracy for larger and small markers is basically the same. However, there are two ways in which smaller markers can provide accuracy improvements over large markers. A single marker has to be fully visible to provide localization. But the larger the size of an individual marker, the larger the area that is missing, if a part of the marker is obstructed by the image border. Hence, the contribution of edges and corners is reduced for larger markers, which is problematic for camera calibration, as those areas often exhibit the strongest distortion. The result is, that larger markers show a more pronounced center bias, where the quality of the calibration is better in the center of the imaging area than in the corners, while small markers can be detected close to the image borders, see figure 3.3, which reduces center bias.

Lastly, as the number of calibration points decreases due to increased marker size, the probability of overfitting increases due to the reduced ratio of calibration points to calibration parameters.

3.2.3 Fractal Layout

For the reasons exposed in section 3.2, we try to minimize the calibration point size. Of course there is a minimal size for calibration points, before detection becomes

impossible, because individual calibration points blend together. However, a pattern which is optimal in one given situation can become completely useless if the magnification is decreased. As the effective scale at which target features are projected onto the image sensor changes with distance, angle and radial distortion, it would therefore be necessary to capture a range of targets, each optimal at a different scale, and later fuse the calibration information from multiple targets.

Our solution to this tedious multi-target process is the adoption of a fractal scheme, which operates with multiple scales of calibration points. By using a target which includes calibration points at multiple scales, it is possible to get optimal calibration point density at several magnification ratios. Figure 3.1 on page 29 shows the calibration pattern at several magnifications.

Detection is initially performed only for the coarsest layer, then again on the next, higher resolution layer, and so on. When some calibration points cannot be matched at one scale, the distances between fitted calibration points are examined to decide whether detection of the next finer scale should be attempted. While this does not guarantee a, in practice hard to define, optimal density of calibration points, it keeps the density within a fixed bound, depending on the effective image resolution and the deterioration by external sources, like noise, blur and other aberrations.

The used calibration points are individual square dots. Due to the fractal nature of the pattern the finest resolvable calibration points are always at the pixel scale, where several independent degradations, like physical pixel aperture, lens aberrations, diffraction, etc., render a calibration point as a Gaussian-like 2D distribution. When a calibration point becomes large enough that the non-Gaussian characteristics come to bear, it is already possible to resolve the next recursion layer, making more complex refinement methods unnecessary. In addition, this elegantly avoids the bias problems of large markers, as the fractal nature of the calibration pattern always allows the detection of relatively small markers, with a size close to the resolution limit.

A new layer of calibration points is generated from a coarser layer by resizing the pattern with a factor of 5, using nearest-neighbor interpolation. In the scaled pattern, new calibration points are inserted at every fifth pixel by inverting the respective pixel, see figure 3.1 on page 29 for a visualization.

3.3 Implementation

The detection of the fractal pattern is split into two parts, the actual detection using the payload for identification, and the fractal refinement. Detection takes

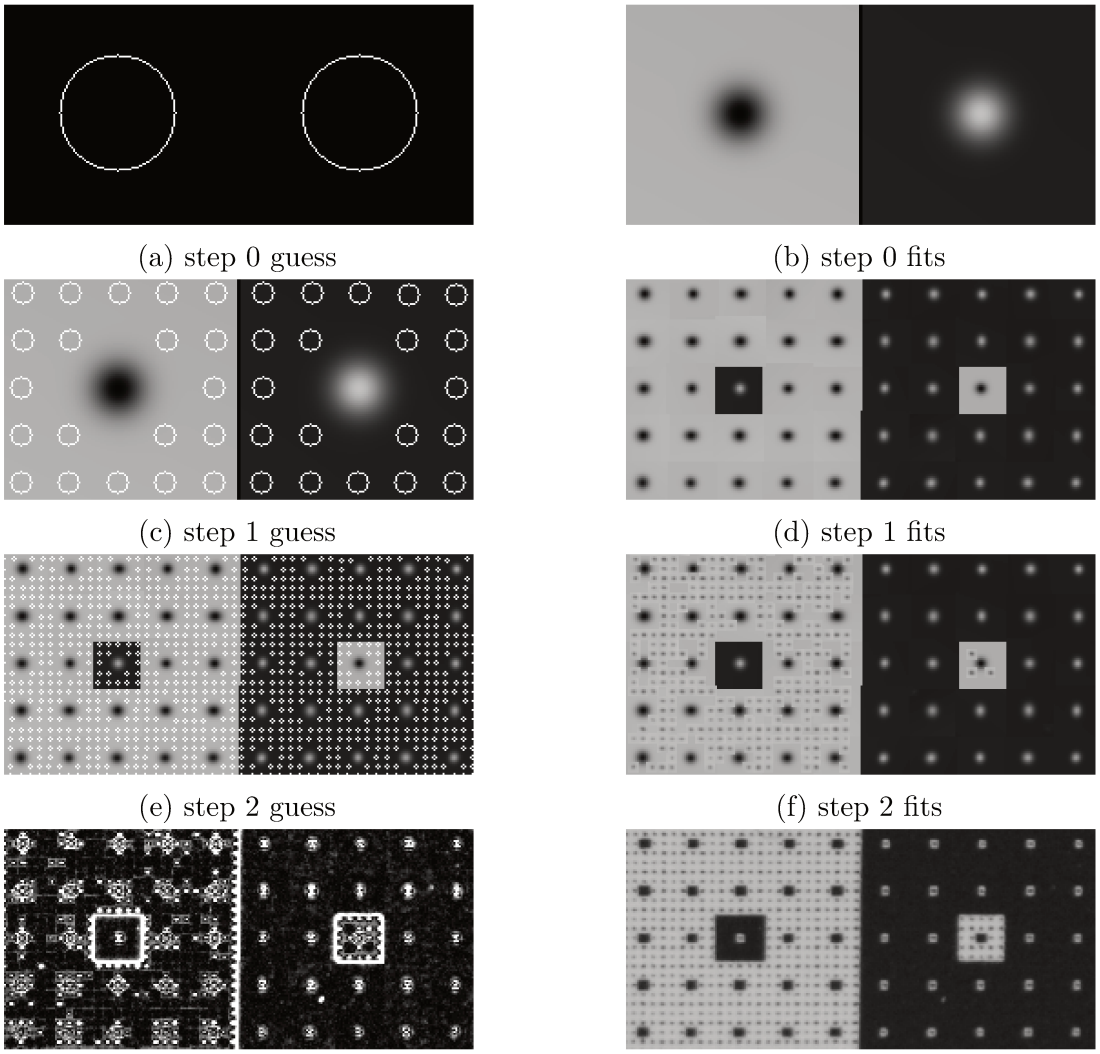
place in scale space, using an x-corner detector followed by a brute force iteration of possible markers using neighboring corner candidates. For each set of four corners, the containing image area is projected onto a marker template using a perspective transform, to assess whether the corner selection represents a marker candidate.

3.3.1 Identification

It is important that parts of the pattern are allowed to be out of view, to give the freedom to cover the whole imaging area with the calibration target, and to place calibration points close to edges and corners. Our method builds on square blocks with uniform borders and a payload of 9 bit in the center, see figure 3.2 on page 30. The orientation is fixed using an opening in the border and markers are arranged in a checkerboard manner, compare figure 3.1 on page 29 and figure 3.5 on page 37. This means that borders are either black or white. The coding scheme combines the payload of black and white markers to provide 18 bits of payload for addressing and always requires the detection of at least two neighboring black and white markers. The address is encoded using an XOR mask to provide a bit of randomization and to keep the overall distribution of bits more uniform. In contrast to other fiducial marker systems, like for example CALTag [4], the payload within the markers does not provide error detection or correction. Instead, the address of neighboring markers is compared to check whether they are consistent, which provides very robust error detection but no error correction. This means the marker grid can have a maximum size of 512 $\times$ 512 markers which results in 262144 markers overall. For more details on the identification scheme please see [87].

3.3.2 Fractal Refinement

The fractal refinement starts after markers have successfully been detected. Expected positions of the first layer of calibration points are estimated from their respective marker using a perspective transform, compare figure 3.4 on the facing page, (a),(c),(e). Those positions are then refined using a least-squares solver which fits rotated 2D anisotropic Gaussian distributions, with a linear gradient as background, compare figure 3.4 on the next page. Individual pixels are used as samples in the fit, with a weighting relative to the initially estimated position. This step is repeated once more to improve accuracy by using the updated center position.



(a) step 0 guess

(b) step 0 fits

(c) step 1 guess

(d) step 1 fits

(e) step 2 guess

(f) step 2 fits

(g) diff between (f) and (h)

(h) input image

Figure 3.4: Visualization of the fractal refinement scheme. From top to bottom, the refinement from coarse to fine is shown, with candidates on the left and the actual fits on the right. As edges are ignored they are simply estimated from the marker size for visualization purposes, while the 2D distributions (right side) are a correct visualization of the estimated fit. Calibration point positions at layer n are estimated by a perspective transform from known calibration points of layer $n-1$. These estimation candidates are shown as circles in (a),(c),(e). Then, individual 2D distributions are fitted to these estimates. The resulting fits are visualized in (b),(d),(f). This process is repeated until calibration point size is too small for a successful estimate. For reference, (h) shows the input image and (g) shows the difference between the final fit and the input, multiplied by 10. Not that all calibration points which were successfully detected in (f) show no discernible error in (g) (black areas).

The recursion works by repeating this procedure, see figure 3.4 on the preceding page, until the size of the calibration points becomes too small for a meaningful fit, which is less than 4 pixels across. Various heuristics are used to verify the validity of the fit, like rejecting calibration points for which either the residual of the fit is too large, or for which the expected accuracy is low due to low contrast, steep background gradient, too small or too large width of the Gaussian, or due to saturation.

3.3.3 Bayer pattern mode

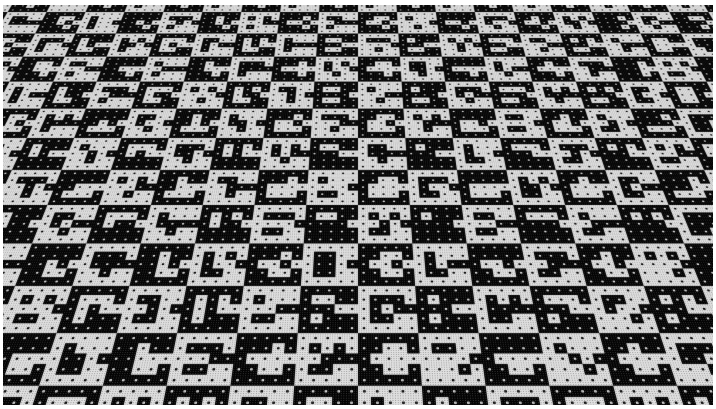
Because individual pixels are directly used as data points in the fit, it is possible to leave out individual samples. For color images acquired using a color filter array, demosaicing can be avoided by processing colors independently of each other, simply leaving out the pixels which belong to a different color channel.

3.4 Evaluation

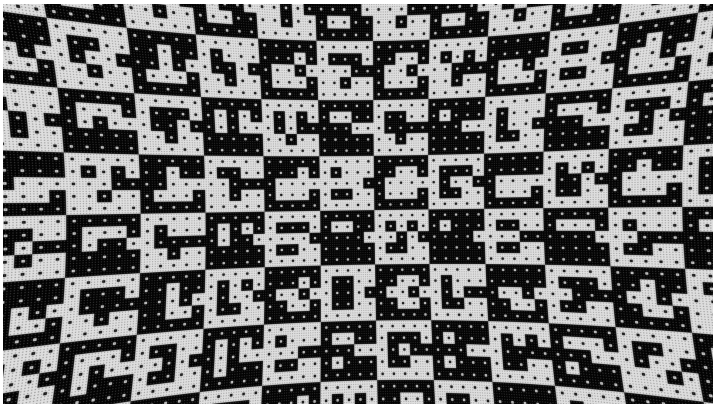
The evaluation is based on rendered images corrupted with Gaussian blur, additive Gaussian noise, uneven illumination, reduced contrast and radial distortion. We compare classic checkerboard detection with subpixel refinement using the implementation from OpenCV, with our fractal calibration pattern. The detected calibration points are then used for either a full camera calibration, again using the implementation from OpenCV, or to solve only for the camera pose under known camera intrinsics (Perspective-n-Point).

3.4.1 Ground Truth Data

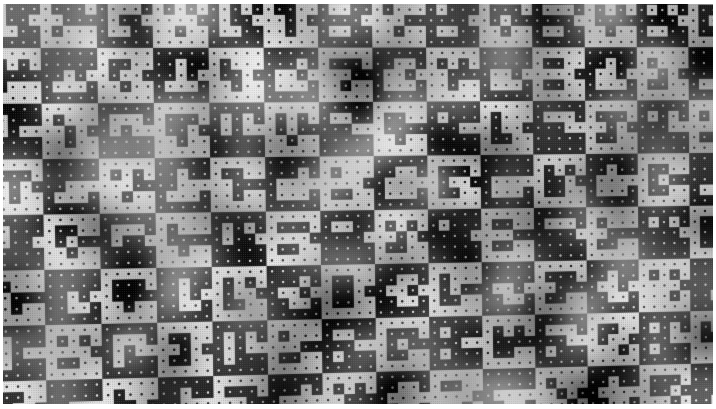
The evaluation images were rendered with the Cycles renderer from Blender [11] using 1024 samples per pixel, with Gaussian anti aliasing at a resolution of 1920 x 1080. For distortion simulation the images were rendered with the resolution increased by a factor of four and scaled down after performing the distortion in order to reduce the influence of the interpolation. Figure 3.5 on the next page shows some examples of the calibration target, filtered with the described degradations.



(a) view angle



(b) radial distortion



(c) uneven illumination

Figure 3.5: Examples of the applied degradations. The view angle images were rendered from different viewpoints, the radial distortion images were interpolated from images with the linear resolution increased by a factor of four. The uneven illumination images were generated by addition and multiplication with very low resolution noise images.

3.4.2 Error Metric

In the literature cited in section 1.3, there are two principal methods of evaluating calibration patterns, the RMS error of the calibration model and the RMS error of calibration points against ground truth, called true pixel error (TPE) by Douxchamps and Chihara [24]. The calibration RMS is a poor choice, as it will not detect bias but simply incorporate it into the model. The TPE works well for example for the evaluation of different refinement methods of corner points, but it cannot be used for a meaningful comparison of calibration targets where the number of calibration points is not roughly the same. The reason is that more calibration points of similar localization performance obviously lead to better calibration results which is not reflected in the TPE.

A further method, which is frequently encountered, is to compare triangulated world coordinates of calibration points or the extrinsics of the calibrated cameras. While these are valid metrics for some applications, they will also miss problematic areas like the corners of the image, if there were no calibration points present for that area. This is especially problematic, because the calibrated model will be bad specifically in those areas where no calibration points were acquired.

However, with ground truth data, it is possible to compare the projection defined by the fitted camera model with the known ground truth projection. This gives an error measure which relates to the actual camera calibration and incorporates the error distribution and bias of the used calibration pattern, as well as averaging from multiple markers. We compare the camera models in image space, by projecting each pixel into the scene, at the depth of the target, using the ground truth model, and then projecting them back into image space with the calibrated model. The root-mean-squared difference between the original pixel coordinate and the projection of the calibrated model defines the error metric, which we call ground truth pixel error (GPE):

$$\mathrm{GPE}(z) = \sqrt{\sum_i \frac{(i - m(m_{GT}^{-1}(i, z)))^2}{\#i}}. \tag{3.1}$$

The GPE is calculated using the root-mean-square error between every pixel coordinate i and the results of projecting this pixel into the world using the depth z, the known ground truth projection m_{GT} and reprojecting with the evaluated model m. Note that R,T are set to zero, because we only want to evaluate the (un-)distortion performance.

For comparison, when (i, t) iterates all target to image correspondences detected by the calibration pattern, the RMS is calculated as:

$$\text{RMS} = \sqrt{\sum_{(i,t)} \frac{(i - m(t)^2}{\#i}}, \tag{3.2}$$

which shows how the RMS is merely the residual between model and (detected) calibration points and hence heavily biased by the distribution and quality of the calibration points, while the TPE

$$\text{TPE} = \sqrt{\sum_{(i,t)} \frac{(m_{GT}(t) - m(t))^2}{\#i}}, \tag{3.3}$$

shows the error between ground truth and model, but again only for the detected image points.

The GPE measures an error in pixel units, stating the localization accuracy of the tested pattern in the context of the tested calibration model. The metric gives a score averaged over the whole image, including the edges and corners, which are the most problematic areas in the context of camera calibration, both because they exhibit the strongest distortion and because it is difficult to place calibration markers on the edges of the image.

For this evaluation the z-coordinates are calculated from the known target position, using the depth of the first view. Alternatively the metric can also be calculated over a depth volume, using a number of depth slices for the reprojection. However, the results then depend more on how good the calibration method can estimate the camera model from a limited number of slices through this volume, than from the marker accuracy. Therefore, in the following the GPE is always measured at the depth of the calibration target. If multiple views were tested, the depth from the first target is used.

3.4.3 Comparison

Table 3.1 on the following page compares the developed fractal target with a range of targets found in literature. A quantitative comparison with the common checkerboard target is performed in section 3.4.4.

As can be seen from table 3.1 on the next page the fractal target is the only one that can keep a high density of calibration points, independent of camera magnification, which leads to a high effective calibration accuracy, see also section 3.4.4. The high marker density enables the use of the fractal target for ray based camera calibration, as shown in chapter 4.

	Accuracy	Density	Completeness
Fourier tag [75]	—	low[1,2]	low
ARTag [28]	low	medium[1]	high
CalTag [4]	medium	medium[1]	high
da Camara Neto et al. [20]	medium	medium[1]	high
AprilTag [66]	—	low[1,2]	low
Daftry et al. [21]	medium	medium[1]	high
Bergamasco et al. [9]	medium	low[1,2]	low
novel fractal pattern	high[3]	high	high

[1] Effective image space marker density depends on magnification
[2] Use case from literature is individual markers with data payload (for *e.g.* VR or interaction)
[3] individually markers not very accurate, but very high number of markers lead to high overall accuracy, see figure 3.11 on page 44 and figure 3.15 on page 46.

Table 3.1: Comparison of calibration target types from section 2.4.2 according to the key properties listed in section 2.4.1.

3.4.4 Evaluation

For reference, a checkerboard with 12 × 6 squares is detected using the checkerboard detection from OpenCV with subpixel refinement over an area of 21 pixels. When evaluating the performance under blur and noise we noticed that under most circumstances the refinement was dramatically improved by performing a Gaussian blur with a sigma of 2 before refinement, which is therefore used in the whole comparison.

In most cases the results of the fractal target surpass the accuracy of the checkerboard pattern by one to two orders of magnitude, for example in figure 3.6 on the facing page and figure 3.7 on the next page. In general the calibration is very robust to noise, reduced contrast, uneven illumination and radial distortion. Two areas where the method is not as robust and where it is eventually surpassed by the checkerboard detection are detection under shallow angles and under strong blur. The reasons and possible countermeasures are discussed in section 3.5.

In figure 3.6 on the facing page the GPE is evaluated under varying radial distortion, as shown in figure 3.5(b) on page 37. Both methods are quite insensitive to radial distortion, with only slight increases in GPE with increased radial distortion. The fractal target has over a magnitude lower GPE for the tested range.

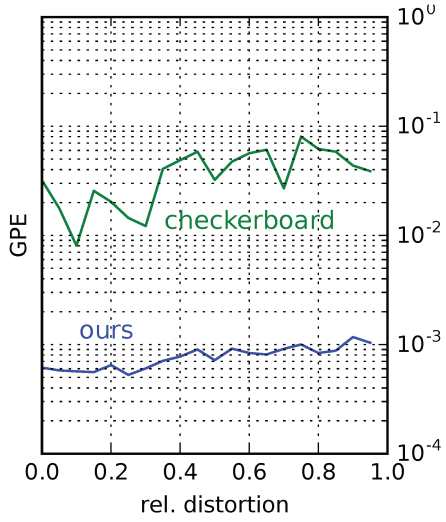


Figure 3.6: A plot of the GPE obtained dependent on the relative radial distortion, compare figure 3.5(b) on page 37. The fractal target provides both better GPE scores, and a lower variability of the results. Both methods achieve slightly worse results with increased distortion.

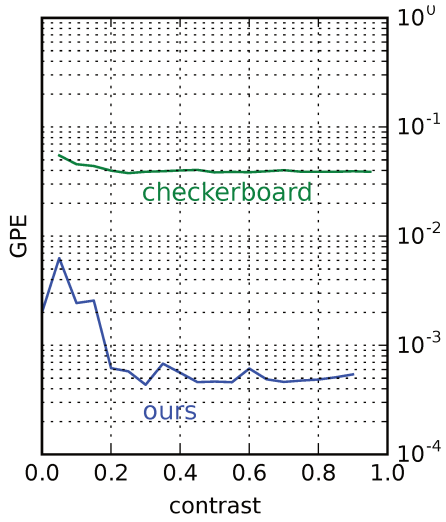


Figure 3.7: With decreasing contrast the GPE stays constant, but there comes a point where the quality starts to decrease. Reaction is stronger for the fractal target, but the results are always better than for the checkerboard target.

Figure 3.7 shows another variable where both methods show strong performance. Under varying contrast the results a very identical for a large range, with an increase in GPE at low contrast values. Here the results are nearly two orders of magnitude better for the fractal target.

Again similar are the results when the camera is rotated around the camera z axis, see figure 3.8 on the next page. In this case both methods show an interesting symmetry, although the checkerboard method has a higher variance.

One area where the fractal target delivers worse results is under strong perspective, *e.g.* when looking at shallow angles, see figure 3.9 on the following page.

Figure 3.8: Plot of the dependency of the GPE on the rotation around the z axis. Both methods show a distinctive symmetry which hints at some underlying systematic dependency on the z-angle. Results for the fractal target have a lower variance and are more stable.

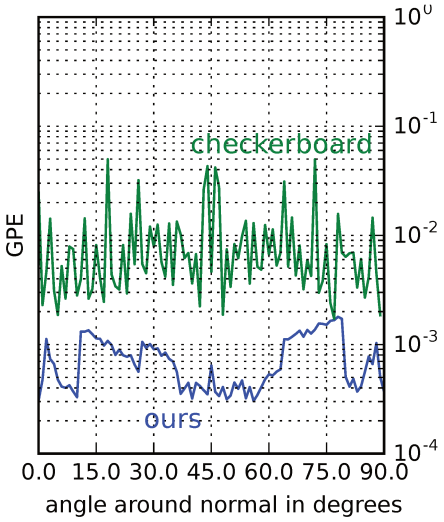
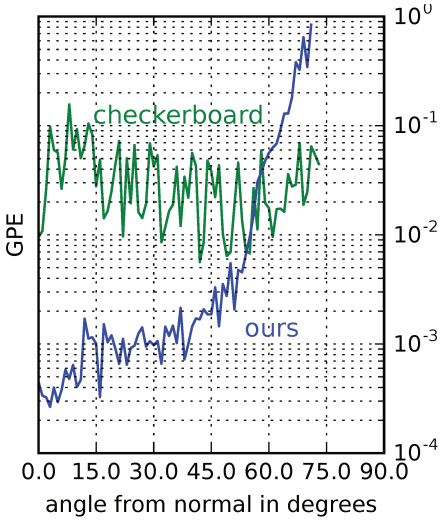


Figure 3.9: Plot of the dependency of the GPE on the view angle. Zero degrees means the camera points directly towards the target, while at 90 degrees it is oriented parallel to the target. The checkerboard target is insensitive against the view angle, but the fractal target shows a strong dependency and eventually is surpassed by the checkerboard target at around 60 degrees from the normal.



In this case the performance drops under the performance of the checkerboard detection from an angle of 60 degrees from the normal.

An area where the fractal target shows a particularly large advantage is with strong background gradients, as present with uneven illumination as shown in figure 3.5(c) on page 37. In figure 3.10 on the facing page we can see that the checkerboard target quickly deteriorates under even mildly uneven illumination, where the fractal target stays quite stable. The reason for this behavior is the localization based on the orthogonal gradient method in OpenCV which is very sensitive to gradiation changes, while our fractal target locates individual dots which are nearly unaffected by image gradients.

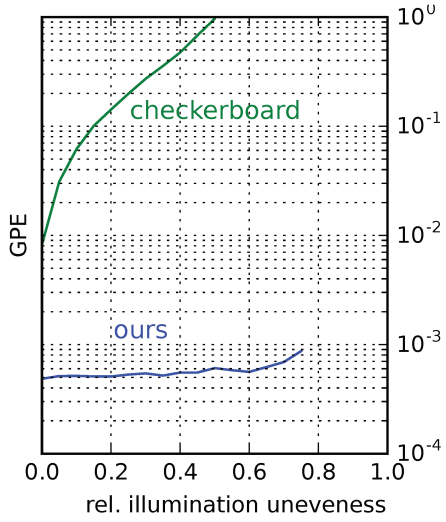


Figure 3.10: This plot shows the influence of an uneven illumination which increases the gradient overlaid on the calibration target. Because the fractal target derives localization information from individual calibration points there is nearly no influence on the accuracy, as each calibration point is composed of gradients in all directions. On the other hand the checkerboard target derives calibration information by observing the orthogonal gradients around a checkerboard corner and is very sensitive to strong image gradients.

The effects of Gaussian blur and noise are shown in figure 3.11 on the next page, which shows that our fractal target is very robust with regard to noise, but looses accuracy under strong blur, while the checkerboard deteriorates under noise, but is more robust to blur.

One reason for the introduction of the new target was the goal of reducing center bias by adding calibration points close to the edges of the target. To evaluate center bias, the GPE is measured at the four corner pixels, and at the center of the images. The center bias B is then defined as:

$$B = \frac{\mathrm{GPE}_{corner} - \mathrm{GPE}_{center}}{\mathrm{GPE}_{center}} \tag{3.4}$$

A perfect result would thus be a B of zero, while a B of one means that the corners only reach half the quality of the center. As is visible in figure 3.13 on page 45 and figure 3.12 on the next page the checkerboard calibration has a very strong center bias between 10 and 100. On the other hand the fractal target achieves values for the center bias close to one, and even outliers are below 10. On average the fractal pattern again has an advantage of around one order of magnitude.

These values also explain the even larger advantage of the fractal target in figure 3.14 on page 45 which shows the GPE results for the corners only.

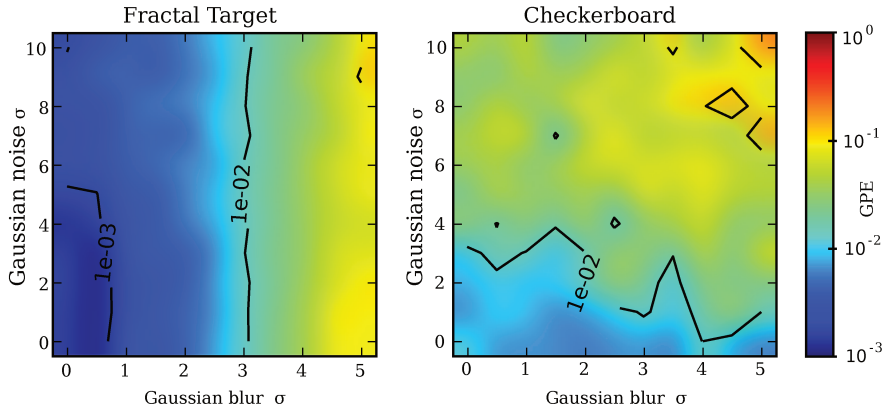


Figure 3.11: The plot shows the quality of the calibration, as measured by the GPE, dependent on both noise and blur. The two targets show quite orthogonal behavior, where the fractal target is robust to noise and the checkerboard target is more robust to blur. Note that the fractal target starts at a much lower error value, hence the checkerboard target only has an advantage in the case of strong blur but very weak noise.

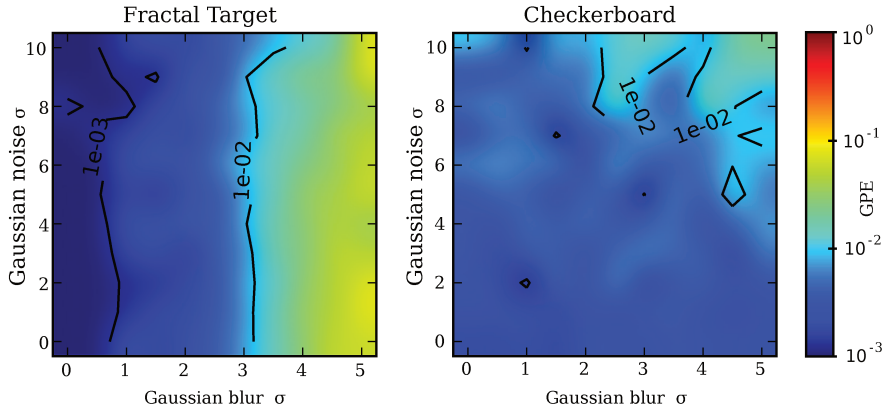


Figure 3.12: Comparison of center and corner quality. The plots show the reduction of center bias by our dense target. The improvements in the corners are much more pronounced compared to the center, with more than two orders of magnitude improvement in some areas. This effect can be attributed to the fact that the recursive target always covers the corners which is not the case for the checkerboard target.

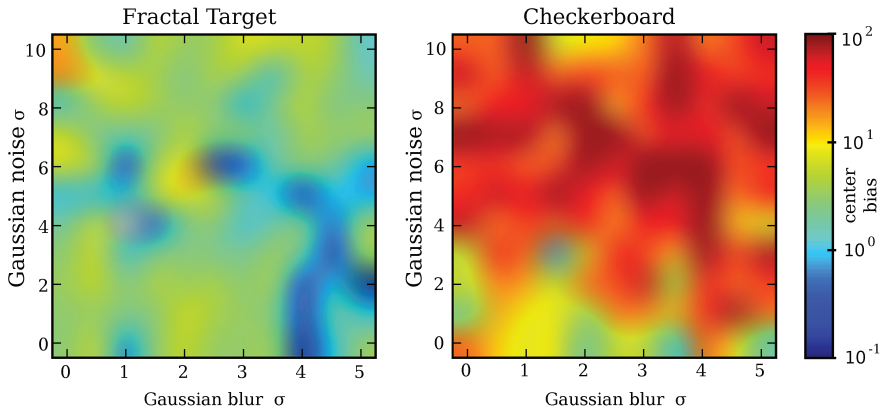


Figure 3.13: Evaluation of the center bias as defined in equation (3.4). The plots show the reduction of center bias by our fractal target. Note that while the overall calibration quality depends on noise and blur, the center bias, which is basically the difference between center and corner results, seems to be independent of those parameters, and is dominated by noise. In addition, the center bias is around an order of magnitude smaller for the fractal target. This can be explained by the ability of the fractal target to place calibration points close to the image border and hence improve calibration results in those areas.

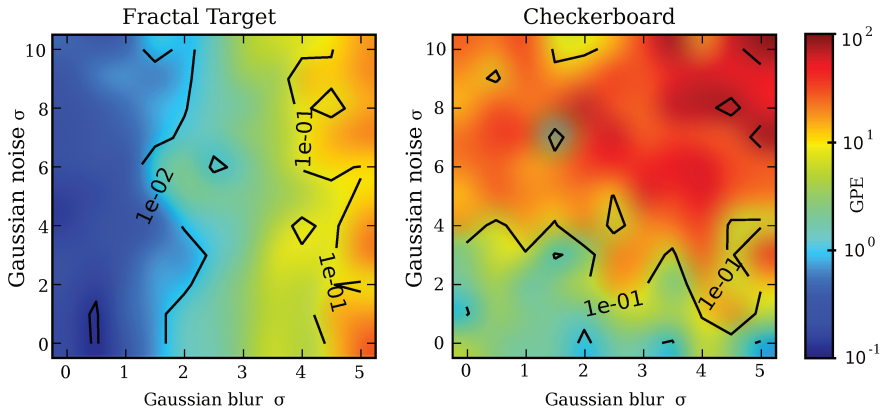


Figure 3.14: Evaluation of the accuracy in the image corners, as measured by the GPE. Compared to figure 3.11 on the preceding page the corners show a more pronounced advantage of the fractal pattern, which also confirms the results of figure 3.13.

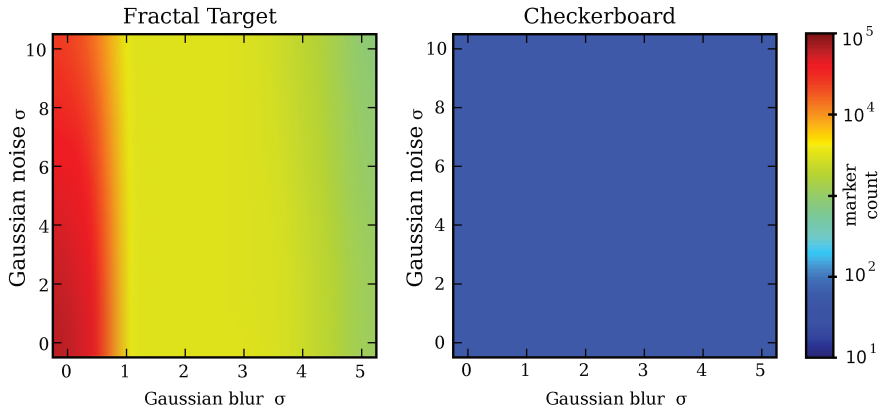


Figure 3.15: This plot shows the number of individual calibration points found for the targets. The checkerboard always returns the same number (else detection would not be possible), while the fractal target finds a much higher number in good conditions, and reduces this number slightly as the noise increases and much more strongly as the blur increases.

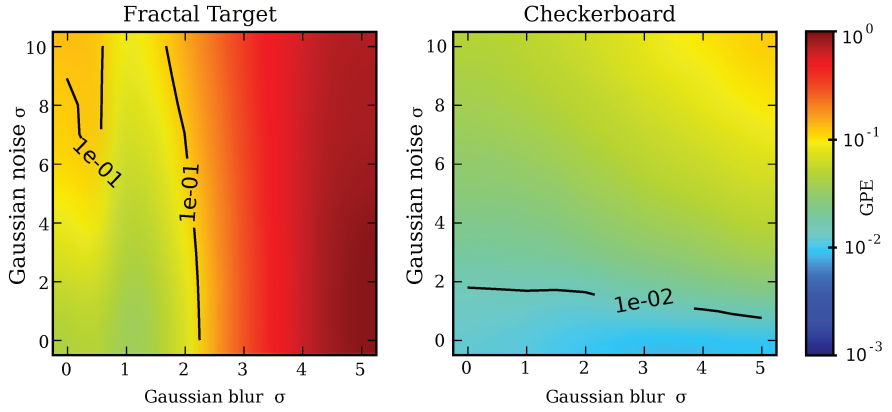


Figure 3.16: The TPE shows a similar behavior to the marker count, compare figure 3.15, for the fractal target, while the quality decreases only with the increase of noise for the checkerboard target. This is in line with the other results, and shows that the checkerboard target performance depends solely on the performance of the checkerboard corner detection.

Furthermore, figure 3.15 on the preceding page shows the number of calibration points that were actually detected, again for different values of blur and noise, while figure 3.16 on the facing page shows the RMS error of the detected individual markers, i.e. not averaged over the calibration (the TPE as defined by Douxchamps and Chihara [24]). From those two plots we can see that, while the RMS error of the individual calibration points is higher in most cases for the fractal target, the number of markers is also much higher which offsets the advantage of the individually higher accuracy markers of the checkerboard pattern, just as argued in section 3.2. Indeed, when comparing figure 3.15 on the preceding page and figure 3.16 on the facing page with figure 3.11 on page 44 it is visible that the quality of the checkerboard calibration is directly linked to the quality of the individual markers as the number of markers is constant, while quality of the calibration with the fractal target is reduced as the number of markers decreases and the individual marker RMS increases due to blur.

3.5 Discussion

While the fractal target delivers most of the anticipated advantages, with highly improved general accuracy, it fails to completely remove center bias and it performs badly under strong blur and shallow view angles.

Regarding the center bias, it seems a dense target is not enough to completely solve the problem. A possible solution could be to introduce a weighting scheme into the calibration method which weights corner samples stronger, however more work is required to estimate good weighting factors and the influence on the overall calibration results.

The second shortcoming, low robustness under shallow view angles and strong blur is a direct result of the desired characteristic of uncoupling the pattern detection from the calibration process. As Mallon and Whelan point out [59], most patterns need to incorporate the calibration within the pattern refinement to remove perspective and distortion bias from the pattern localization. While the checkerboard refinement does not suffer from this drawback, the 2D Gaussian fits used in our target do have this problem. The workaround is to use very small calibration points where this bias is very small, which works for most cases as the fractal structure ensures that the smallest calibration points can be used independently of the magnification. This fails only if the smallest scale cannot be used for another reason than the image scale, which are blur and the perspective distortion, which appears at shallow view angles.

Reliable correction of this bias needs the calibration information, which would tie the pattern detection to the camera calibration. However, this would reduce the universality of the calibration pattern which in the current form can be used completely independent of the camera calibration.

4

Ray-Based Camera Calibration from a Passive Imperfect Target

The most widely used models for camera calibration are based on parametric modeling of lens distortions, often using radial functions, as described in section section 2.3. The reasons are the ease of use provided by calibration methods based on passive planar targets, as well as sufficient accuracy for many vision tasks. However, as the demands for calibration accuracy increase, for example from light field imaging where state-of-the-art methods [80, 86] are now measured with 0.01 pixel error thresholds [48], the absolute performance is increasingly limited by the standard parametric camera models [8, 10], see also section 2.3. The ray based general camera model [40] removes many of the limitations and can increase calibration accuracy even for regular optics [10], but requires complex calibration setups involving active targets. In this chapter we introduce a novel calibration method which can derive a very accurate ray based calibration based on the passive fractal calibration target introduced in chapter 3. The method additionally supports the estimation of the target deformation, easing high quality calibration, as perfectly planar targets are expensive to manufacture. This makes the usage of the ray based calibration model as simple in usage as the standard parametric calibration, but at a much higher quality. The approach scales from non-central wide angle lenses to high quality central lenses.

4.1 Outline

In this chapter a novel, ray based calibration method is introduced. For the related work please refer to section 1.3.2.

First the general camera model and the used parametrization is defined in section 4.2 and with this parametrization, the effectively depth dependent distortion property of non-central cameras is derived in section 4.3. Following this the actual calibration method is described in section 4.4. Section 4.4.9 shows how

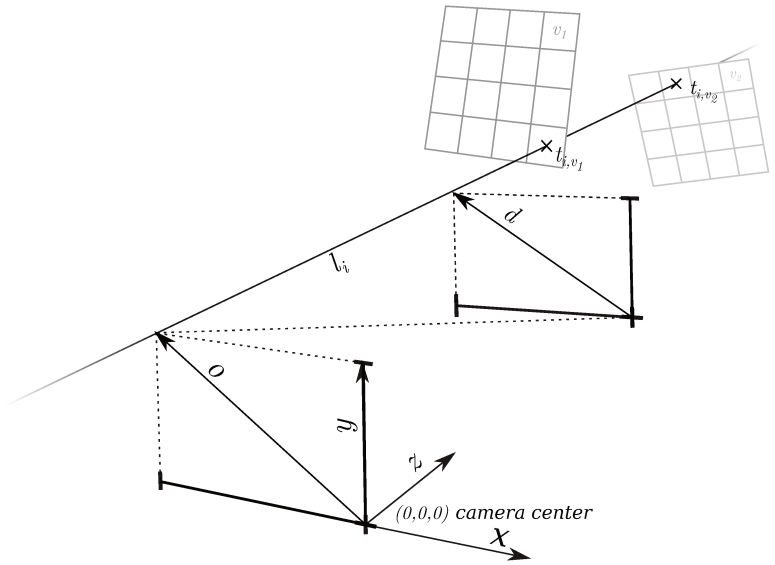


Figure 4.1: Two plane parametrization for the ray based general camera model, in camera coordinates. The ray l_i is defined by the origin o on the camera plane and a direction d. The first plane is at $z = 0$, the second plane at $z = 1$. Also shown are two views v_1, v_2 of the target and the respective intersections t_{i,v_1}, t_{i,v_2} with l_i.

the model can be used to perform undistortion and rectification, on which basis a quantitative evaluation on both synthetic ground truth and a real world stereo validation set is performed in section 4.5.

4.2 The General Camera Model

In the following a parametrization of the general camera model is introduced, which will be used for this work. The general camera model, already briefly introduced in section 2.2, describes an imaging system as a collection of lines in 3D. Each line consists of all world points which are projected onto the same position of the imaging sensor. We restrict the model slightly, by using a two plane parametrization, which limits the calibration to the hemisphere in front of the camera. This simplifies our model by allowing the direct definition of the rays by four parameters, compare figure 4.1. With this restriction we formulate a camera as a collection of pixels, where each image pixel i is associated with a 3D ray l_i which contains all 3D points which are projected onto the pixel:

$$l_i = o_i + z \cdot d_i = \begin{pmatrix} o_{i_1} \\ o_{i_2} \\ 0 \end{pmatrix} + z \begin{pmatrix} d_{i_1} \\ d_{i_2} \\ 1 \end{pmatrix} \tag{4.1}$$

Here, o_{i_1}, o_{i_2} refer to the origin on the camera plane (the xy-plane in camera coordinates). The ray direction is given by d_{i_1} and d_{i_2}. For a point represented with equation (4.1) the depth (the distance from the camera plane) is directly represented as z. Thus, equation (4.1) describes the mapping from 3D camera coordinates onto the pixels of the imaging device. To relate the rays which are given in camera coordinates, with a point in target coordinates, the pose between calibration target and camera is also required. This is given as a rotation R and translation T, from target into camera coordinates:

$$p = R \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} + T \tag{4.2}$$

Here, $t = (t_1, t_2, t_3)$ is a point in target coordinates and p is a point in camera coordinates. The point p must be contained in the ray l_i which represents the projection from world to image coordinates, so we set:

$$l_i = p_i \tag{4.3}$$

$$o_i + zd_i = R \cdot t_i + T \tag{4.4}$$

Which, given multiple target points, can be used to solve for the ray parameters as well as R and T simultaneously. Analogous to the parametric models used for central cameras, R and T describe the extrinsics of the camera, while o and d are the intrinsic parameters, which define how points in camera coordinates are projected onto the imaging plane.

4.2.1 Canonical Form

Lets regard a camera c with the intrinsics defined by the rays l_i, as well as extrinsics R and T. This camera is not unique, because a change in the extrinsics can be compensated by an appropriate change in the ray parameters. The consequence is that the origin of the camera coordinate system is not connected to the rays in any way, and hence does not have any correlation with the real camera pose. To make our model comparable to classic, parametric camera models we want

the z-axis to point forward and the x-axis to point right. Also, we would like the origin to be the point where the rays are closest to each other and therefore most closely resemble a central camera. We therefore impose several constraints on the canonical form $\hat{c}$ of the camera model:

1. The center ray l_c points from the camera center in the positive z-direction: $o_{c_1} = o_{c_2} = d_{c_1} = d_{c_2} = 0$. This prevents rotation and translation of the model around, the x- and y-axis of the camera.

2. A fixed second ray l_r may not point up or down: $d_{r,2} = 0$. This prevents rotation around the z-axis. We choose the ray directly to the right of l_c as l_r.

3. To move the approximate projection center to the origin (in camera coordinates), the canonical form $\hat{c}$ of all equivalent cameras c must satisfy:

$$\hat{c} = \arg\min_{c} \sum_{i} {o_{c,i}}^2 \tag{4.5}$$

4.3 Depth Dependent Distortion

A non-central camera in the form of the general camera model can be regarded through a regular pinhole camera, as introduced in section 2.2.2, by projecting individual rays with the camera matrix C, which is made up of the focal length f and a projection center c:

$$\bar{p} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} o_1 + z \cdot d_1 \\ o_2 + z \cdot d_2 \\ z \end{pmatrix} \tag{4.6}$$

$$\bar{p} = \begin{pmatrix} f_x(o_1 + zd_1) + c_x z \\ f_y(o_2 + zd_2) + c_y z \\ z \end{pmatrix} \tag{4.7}$$

which results in the pixel coordinates:

$$\tilde{p} = \begin{pmatrix} \frac{f_x o_1}{z} + f_x d_1 + c_x \\ \frac{f_y o_2}{z} + f_y d_2 + c_y \end{pmatrix}. \tag{4.8}$$

This directly shows that the direction of the ray specifies the constant part of the projected pixel position, while the origin of the rays appears as a depth dependent term. In other words the depth dependent distortion is directly caused

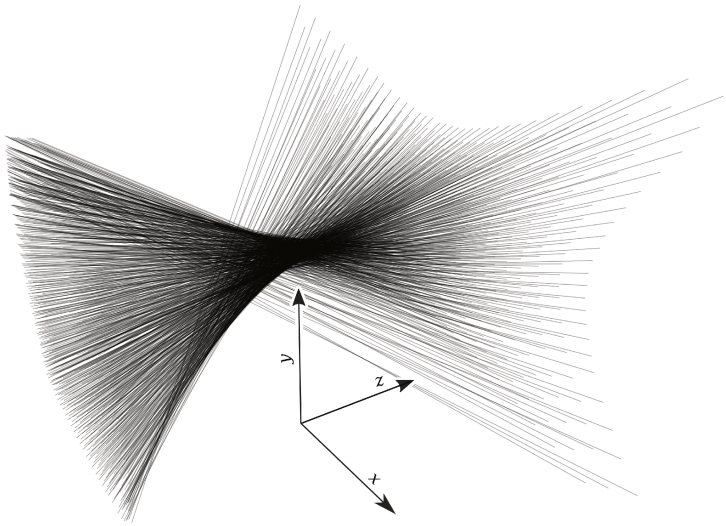


Figure 4.2: Visualization of the non-central nature of a fisheye lens. The image shows a 10 mm depth slice of the central area. For comparison, the vectors of the axis indicator have a length of 1 mm. The standard deviation for the ray origins, estimated from the covariance of the solution, were around 50 µm.

by the non-central characteristics of the imaging system, while image space distortions depend on the direction of the rays. This dependency is also visualized in figure 4.2 and figure 4.3 on the following page, which visualize depth dependent distortion from an actual fisheye lens.

4.4 Calibration Method

The goal is to derive the calibration parameters using observations of the passive fractal calibration target, introduced in chapter 3. To this end, we first derive sparse image-to-target correspondences from the calibration images, see section 4.4.1. We assume smoothness of the calibrated projection, and interpolate the missing correspondences for individual rays from these sparse correspondences, see section 4.4.2. To enable the calculation of model errors in image space, the derivative of the interpolation is exploited. The actual calibration is then mostly a question of correctly formulating the error metric and adopting a sensible initialization, as detailed in section 4.4.5 and section 4.4.5. Finally, the actual

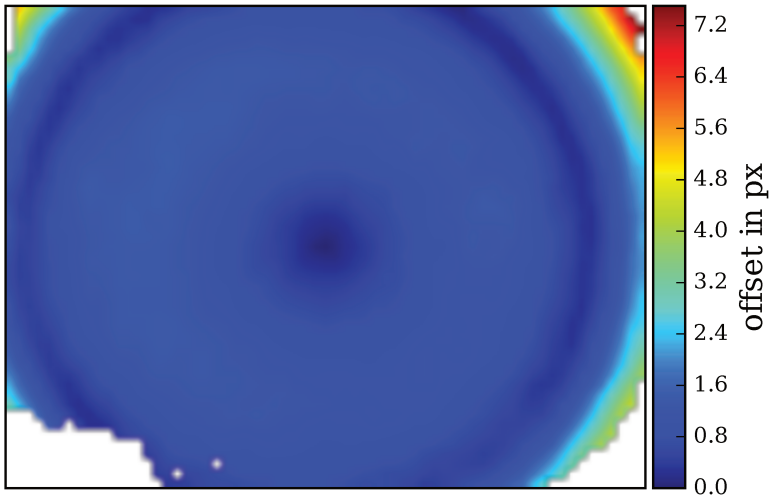


Figure 4.3: Depth dependent lens distortion of the fish-eye lens, according to equation (4.8), at a distance of 1 m. Note that the depth dependend distortion component is around one pixel for a large part of the image, highlighting the need for a non-central camera model for this type of lens.

calibration is implemented using a standard non-linear least-squares solver, in this case *Ceres Solver* [2], see section 4.4.6.

4.4.1 Input

For the calibration target we use the fractal calibration target described in chapter 3. It provides many calibration points, independent of image magnification, and hence gives a relatively dense sampling, for a passive target, of correspondences between image points j and the respective target points t_j. For color images acquired using a Bayer pattern sensor, the pattern detection allows the selection of the relevant pixels of the image, allowing the separate detection of the different color channels without the possible bias induced by demosaicing.

4.4.2 Calibration Proxy

Recall that the general camera model consists of a collection of rays. Each ray l_i maps a single image location i, to a collection of target locations $t_{i,v}$, one for each view, see figure 4.1 on page 50. For successful calibration it is necessary to obtain multiple corresponding target points for each ray, to provide the support

points which define the ray in space. However, given the sparse collection of correspondences from the passive calibration target, only few pixels, if any, will have multiple corresponding target matches.

Similar to Ramalingam *et al.* [69] we assume that the projection of our camera is smooth in image space, and derive the missing target correspondences by fitting a function $\Theta_{i,v}$ which maps from image to target coordinates. This fit is done independently for each ray i and view v.

Because of the assumed smoothness of the derived mappings we do not require one ray per pixel, but instead use a subset of $M \times N$ pixels. We call this collection of target locations, calculated by the function fits, the *calibration proxy*:

$$\{t_{i,v}\}_{\substack{i \in M \times N \\ v \in \text{views}}} \tag{4.9}$$

We use $M = 65$ and set N accordingly, depending on the aspect ratio. Compared to Ramalingam *et al.* [69], who derive the image to target mappings by fitting a local 4-point homography, we extend the model by adding a generic 2D polynomial. This allows us to derive a very accurate image to target mapping around the pixel coordinate i using a locally weighted collection of all correspondences $C_v = \{(j, t_j)\}$ between pixels coordinates j and target locations t_j of view v:

$$\Psi_{i,v}, \Upsilon_{22_{i,v}} = \underset{\Psi, \Upsilon_{22}}{\arg\min} \sum_{(j,t_j) \in C_v} (t_j - \Theta(j))^2 \cdot G(j - i) \tag{4.10}$$

$$\Theta(x) = \Psi(x) + \Upsilon_{22}(x) \tag{4.11}$$

$$G(x) = \exp\left(\frac{x^2}{2\sigma^2}\right) \tag{4.12}$$

Here Θ estimates target coordinates from an image coordinate x using a perspective warp Ψ and the 2D polynomial Υ_{22}. The quadratic error is weighted according to the Gaussian distribution G, to give more weight to samples which are close to the desired image coordinates. The σ is a constant, expressing the smoothness of the mapping. Ψ is a simple perspective transform using 8 parameters, while the 2D polynomial Υ_{22} has the constant and linear terms removed, as those can be modeled by Ψ:

$$\begin{aligned} \Upsilon_{22}(x) = \quad & \upsilon_{13} \quad y^2 \\ & + \upsilon_{22}\, x \;\; y + \upsilon_{23}\, x \;\; y^2 \\ & + \upsilon_{31}\, x^2 + \upsilon_{32}\, x^2\, y + \upsilon_{33}\, x^2 y^2 \end{aligned} \tag{4.13}$$

Our implementation also allows, if desired, the usage of polynomials of higher degrees, with higher order terms added according to the scheme in equation (4.13).

The result of equation (4.10) is used to calculate the corresponding target locations for all rays i in all views v, which are then stored in the calibration proxy:

$$t_{i,v} = \Theta_{i,v}(i). \tag{4.14}$$

4.4.3 Calibration

In the following, a range of error terms will be introduced, whose minimization can solve the calibration problem, starting with E_1 which is basically just a rewrite of equation (4.3):

$$E_1(o, d, R, T; t) = p - (o + p_3 \cdot d) \tag{4.15}$$

with p as in equation (4.2). The constant target location t is provided by the calibration proxy, while the ray origin o and direction d, as well as transformations from target to camera coordinates given by R and T are the parameters we would like to estimate. In E_1 the error is calculated as the difference, in camera coordinates, between the measured target location, and the intersection between the ray and the plane defined by p_3. The desired calibration is then estimated using a non-linear least-squares approach:

$$\underset{o,d,R,T}{\arg\min} \sum_{\substack{i \in \text{rays} \\ v \in \text{views}}} E_1(o_i, d_i, R_v, R_v; t_{i,v})^2 \tag{4.16}$$

This formulation already gives usable results when initialized correctly, see section 4.5, but there are possibilities for improvement.

Because E_1 calculates the residuals in camera coordinates, the different scales and orientations under which the target is observed is a source of heteroscedasticity in E_1, as errors arise from the marker localization in image space, and not in camera space. With parametric central cameras this is easily accounted for, by using the reprojection error as the error metric. With ray based models these effects are difficult to account for, because we are missing a projection. Indeed, Ramalingam *et al.* [69] do not consider these effects, while Bergamasco *et al.* [8] implement only a partial compensation, and assume scale effects to be negligible.

However, in this case the calibration proxy not only derives correspondences, but functions $\Theta_{i,v}$, which map from image to target coordinates, see equation (4.11). To calculate the error in image coordinates we regard the error propagation using first order Taylor series expansion of $\Theta_{i,v}$ about the image point i, given as the Jacobian matrix $J_{i,v}$, which can be calculated numerically on proxy generation.

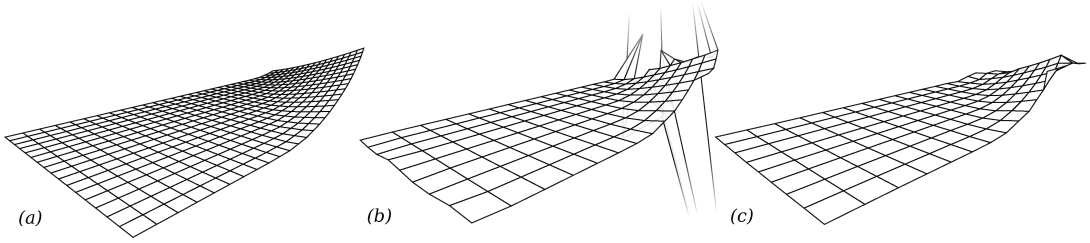


Figure 4.4: Example of a calibrated deformed target. *(a)* The mesh used to render the calibration images. *(b)* The fitted target deformation. *(c)* Added border constraints, see section 4.4.4.1.

The step from camera to image coordinates can then be implemented using R^{-1} to move from camera to target coordinates, and the inverse of the Jacobian, to get from target to image coordinates. The error in image coordinates, given constant target coordinates t and Jacobian $J_{i,v}$ is then:

$$E_2(o, d, R, T; J_{i,v}, t) = {J_{i,v}}^{-1} \cdot R^{-1} \cdot E_1(o, d, R, T; t). \tag{4.17}$$

4.4.4 Target Geometry

The above error metrics still assume perfect target geometry. However, it is quite difficult to fabricate sufficiently accurate targets [3, 91]. Normally, when calibrating using a planar target, the same target points are observed at different image locations in multiple images. It is then possible to add the 3D locations of the observed target points as additional parameters to the calibration model, and jointly optimize the target geometry and the camera model [91].

In our calibration model this is not possible, as the image points are defined by the rays, and the associated target points vary. To enable the determination of target geometry, the target is therefore modeled as a fixed resolution mesh using a regular grid, where each grid point g stores the offset from the perfect target. These offsets are transferred to the observed target locations in the proxy using bilinear interpolation. This influence of grid points to the observed target points is only determined by the observed target coordinates, and therefore independent of the optimization process. For the implementation this means that the additional parameters can be passed as four 3D parameters with associated constant weights ν determined by the bilinear interpolation in target space:

$$E_3(g, o, d, R, T; \nu, J_{i,v}, t) = {J_{i,v}}^{-1} \cdot R^{-1} \cdot E_1(o, d, R, T; t_m(g; \nu, t)) \tag{4.18}$$

$$t_m(g;\nu,t) = t + \sum_n \nu_n \cdot g_n \tag{4.19}$$

Here the ν_n are the additional constant weights, determined using the known target coordinates and the four neighboring mesh points g_n.

Leaving the whole target geometry variable introduces ambiguities, because target scale and orientation are not fixed, which means that changes in the target orientation and scale can be countered by corresponding changes of the extrinsics between target and camera. The solution, as described by Strobl and Hirzinger [91], is to fixate three points of the mesh.

4.4.4.1 Border Constraints

At the borders of the visible area the deformation problem is not well posed, because it is possible that a grid point influences only a single residual. To avoid artifacts on the border of the mesh, we additionally add an explicit smoothness term only to grid points which fall below a certain number of samples. See figure 4.4 on the preceding page for an example of a calibrated target.

4.4.5 Initialization

Due to the very generic formulation of the general camera model, the problem is non-convex and requires sensible initialization. To this end we initialize under the assumption that we have, in a small part of the image, an approximately rectilinear projection, and add an error term which conforms to the traditional projection model, using an additional unknown parameter for the focal length f plus a camera center which is fixed at half the image size. This term is inversely weighted with the distance to the image center, so only the center area is forced to conform to the regular central camera model.

The solver is then started with a small subset of all observed views and a set of sensible initial parameters. The used parameters, which worked for all cameras that were tested, are a set of extrinsics which place all targets in front of the camera at a distance of 10 meters, a focal length of 10000 pixels and initial rays that all point towards the target in the same direction.

4.4.6 Implementation

To derive the calibration we use a standard non-linear least-squares solver [2], minimizing E_1, but with all ray origins fixed at $(0, 0)$ and the target mesh fixed at the perfect geometry. After this initial problem has converged we relax the

problem by removing the projection constraints, replacing E_1 with E_3, and adding the target geometry as parameters to the solver. In the last step the fixed ray origins are also free to vary, so the model jointly optimizes rays and geometry.

For the whole calibration procedure the canonical constraints 1 and 2 from section 4.2.1 are enforced by setting the respective parameters to zero. The resultant calibration is not necessarily in the canonical form, as equation (4.5) is not enforced. Therefore, a post-processing step is performed after successful calibration, adjusting T_3 and the ray origins o_i, so equation (4.5) is fulfilled.

Note that the implementation is a direct, very straightforward implementation of the equations presented in this work, all the magic required to perform least-squares optimization, like auto-diff, is provided by the Ceres Solver library [2].

4.4.7 Outliers

The input target/image correspondences, and the calibration proxy sometimes contains a few outliers. This happens nearly exclusively when input images exhibit strong blur or noise. For this reason we filter out outliers when fitting the proxy, as well as on final calibration, by examining the sample residuals and removing samples above a certain threshold (we used 5× the median). In all experiments the number of outliers was below 3% of the total sample count.

4.4.8 Stereo and Multi-View

The calibration model is trivially extended to calibration scenarios with multiple cameras, by adding additional constraints where relative camera poses should stay the same. For example if the relative extrinsics between the two cameras A, B of a stereo setup should stay the same (otherwise stereo imaging would not make much sense), then an additional error term is added as:

$$E_{A,B} = \sum_v \kappa(R_{v_A} - R_{v_B} - R_{AB})^2 + \eta(T_{v_A} - T_{v_N} - T_{AB})^2. \tag{4.20}$$

Here v indexes different views of the calibration target, taken with both cameras at the same time.

The new parameters R_{AB} and T_{AB} give the relative rotation (as an angle-axis vector) and the relative translation between the two cameras which are new unknowns which will be estimated by the least-squares optimization. This additional constrain enforces that this relative pose stays the same over all

calibration views. When the whole setup (or the calibration target) is moved, and as a result the relative pose also changes, then $E_{A,B} > 0$.

Note that it is not necessary to estimate these jointly, they can also be estimated after running the full calibration by fixing all other parameters, which allows an estimation the stability of the stereo or multi-view rig, according to the calibration data. The parameters κ and η allow adjusting the weight of the constraints relative to each other and to the main calibration residual. Low weights will not influence the calibration result much, while large weights will lead to a large overall residual if the setup is not rigid, thus allowing the evaluation of the rig rigidity.

This extension to multi-view is a significant improvement on the method of Bergamasco *et al.* [8], which calibrates multiple cameras separately and then calculates the transforms in a second step.

4.4.9 Correction Tasks

To calculate the corrected images from distorted inputs, we use the calibrated ray model of a camera and intersect the model rays with the depth plane for which we want to derive the correction. The points resulting from the intersection are then projected with the desired virtual camera (*e.g.* a pinhole camera). The result is a sparse pixel to pixel mapping between the distorted and the desired undistorted images. To derive a dense interpolation we chose thin plate splines [25], due to their excellent extrapolation properties, which is important at the image borders. Note that the depth chosen for the intersection only matters for non-central cameras, or when performing rectification. If the calibrated camera is central and the virtual camera is placed at the same position, then the depth has no effect on the calculated mapping.

For undistortion we simply place a virtual camera exactly at the position of the calibrated camera.

For rectification, we have an additional constraint for the reference camera, whose x-axis must be parallel to the line given by the extrinsics between the two (or multiple) cameras, which will result in an undistortion which fulfills the rectification constraints on the remapped images.

4.5 Experiments

This sections evaluates the calibration performance with a range of experiments, both on synthetic data, see figure 4.5 on the next page, and on real images, see figures 4.2 and 4.6 and table 4.1 on page 53, on page 62 and on page 63.

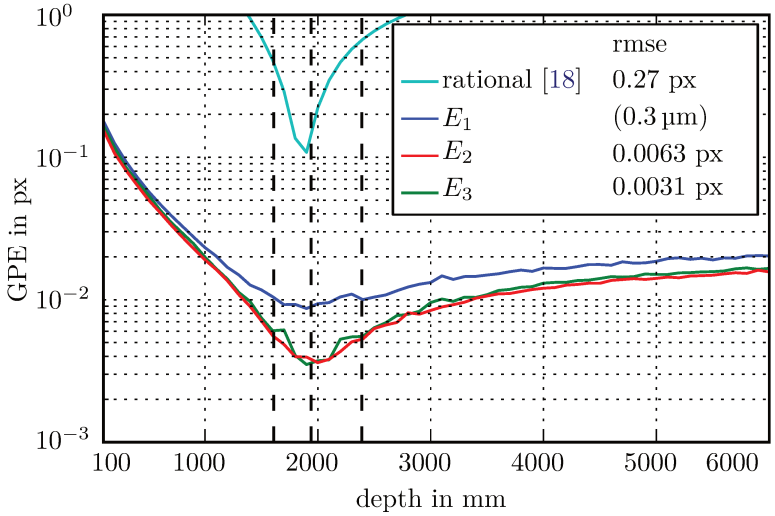


Figure 4.5: Evaluation with a non-central camera on ground truth data, evaluating the *GPE*, see text, against depth. The dotted vertical lines give the minimum, average and maximum distances at which the target was observed in the calibration images. The central rational camera model [18] is unable to cope with the depth dependent effects and optimizes for the average of the observed distortions. Our non-central model achieves much better accuracy and can extrapolate reasonable well to unobserved depths. Note how the full model (E_3) gives slightly worse results as it may deform the target, although in this evaluation the target is perfectly planar. E_1 gives worse results than either E_2 or E_3 due to the heteroscedasticity in the input data, which leads to increased weights for samples observed from a larger distance.

The method is evaluated against the division model implemented by OpenCV [13], which has a very high adoption throughout computer vision community. In addition, the method is compared to the parametric calibration with target geometry estimation [91], using the same distortion model. This evaluation does not seek the best parametric model for the tested lens. In practice, it is not feasible to perform evaluation of distortion models every time a camera calibration is required. The method is also not evaluated against other generic, ray-based calibration approaches, as those require an active target. Where an active setup is feasible, it can be expected that active methods surpass this approach, due to the much higher number of data points which can be collected from an active target.

Model	max	avg max	max rmse	avg rmse	fit rmse (px)
division [13] (checkerboard)	3.0566	0.0308	0.8753	0.2276	0.120
division [13] (calibration proxy)	2.8728	0.0298	0.9517	0.2198	0.096
imperfect target ([91]+[13]))	2.7276	0.0275	0.7561	0.1949	0.023
ours (fully unconstrained)	1.3883	0.0176	**0.3289**	0.1080	**0.0067**
ours (planar)	1.5800	0.0170	0.4343	0.1103	0.0290
ours (central)	1.6156	0.0159	0.3546	0.0965	0.0072
ours (planar & central)	**1.3605**	**0.0158**	0.3492	**0.0963**	0.0320

Table 4.1: Results of the stereo evaluation. For all stereo pairs of the verification set, the checkerboard corners are triangulated and the distances of all directly neighboring corners are compared. We report the highest change between maximum and minimum distance (max), the average over the distance between the max and min over all pairs (avg max), the maximum root-mean-squared deviation from the respective average over all corner pairs (max rms) and the average (avg rms). All results are reported as percentage of the respective average distance. We also list the calibration residual (fit rms). The best values are denoted in bold.

The only ray based method which operates on passive targets is the method by Ramalingam *et al.* [69, 93]. However, the method produces a sparse irregular collection of rays, where the ray placement depends on the target placement and might not extend to the image borders. As the evaluation is based on the correction performance of the calibration, it is not possible to evaluate [69, 93], due to the missing undistortion.

When evaluating the quality of a calibration method, the residual is a poor proxy for the actual quality of the calibration, as detailed section 3.4.2. For example problematic areas, like image corners, may not be part of the calibration data. Also, more data points increase the residual as well as the quality, and more parameters can lead to overfitting. Some of these effects are visible in table 4.1 and figure 4.6 on the current page and on the facing page, compare also section 3.4.4. For this reason we look at two different quality metrics. For experiments on synthetic data we can compare with the known ground truth projection. We use the GPE introduced in section 3.4.2, see also [79, 88], which is defined as the root-mean-squared difference between the ground truth and the calibrated projection over all pixels.

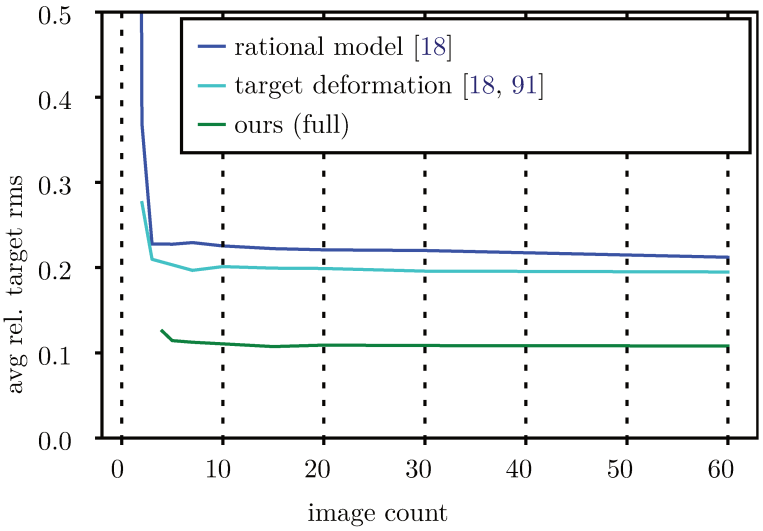


Figure 4.6: Plot of the calibration accuracy, as measured by the avg RMS error of the stereo setup, see table 4.1 on the preceding page, depending on the image count of the calibration set. The number of calibration images only has a small influence on the calibration accuracy.

In figure 4.5 on page 61 we compare a division model as implemented by OpenCV [13], against our unconstrained model on synthetic data, rendered with a non central camera. While the central model fits reasonably well to the average depth observed in the calibration images, our unconstrained model performs much better at all depths. Thanks to the non-central model the performance also decreases much slower at unobserved depths. Results for the calibration of a real non-central imaging system are shown in figure 4.2 on page 53, where we show the results of our method on a wide-angle fish-eye lens, which exhibits strong non-centrality.

The main focus of this work are not the more uncommon non-central cameras, but the usage of the ray based calibration to improve the accuracy of regular lenses, as those might be used for example in stereo or light field setups. To assess the performance from real data, the evaluation roughly follows Bergamasco *et al.* [8, 10] and utilizes a stereo setup. For the evaluation, a set of 60 calibration and 60 verification images are recorded from a stereo rig. The two cameras are calibrated from the calibration set and the resultant models are used to undistort the images in the verification set. The verification set consists of images of a checkerboard target. From the undistorted stereo data sets we triangulate the corners of the checkerboard pattern and observe the distance between neighboring

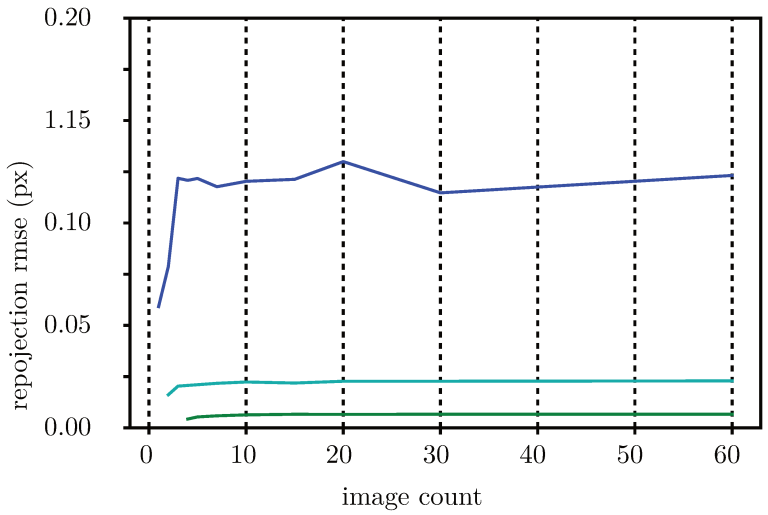


Figure 4.7: Plot of the reprojection error of the calibration against the number of calibration images. Once again, the reprojection error demonstrates its independence from the accuracy of the calibration, as the RMS error actually *increases* with the increase in calibration accuracy, as shown in figure 4.6 on the preceding page.

checkerboard corners as the target is moved around the scene. The consistency of these distances gives us a good external estimate of the quality of the calibration. For a high quality calibration the triangulated points should keep the same distance to each other, independent of the target distance or orientation. The results in table 4.1 on page 62 show that our method is much more accurate even for this highly central, low distortion lens.

We also investigate the target planarity, by performing two additional tests: First we disable deformation estimation in our method. Secondly, we add the estimation of the target deformation by also optimizing the target geometry for the parametric model [91]. We found that while this improves the RMS error a lot, the accuracy is only slightly improved. To check whether the calibrated camera is actually central, we constrain our calibration to only estimate ray directions, with the ray origins fixed at the camera center. As calibration with a fixed planar target and central ray model appears to provide the highest accuracy, we assume that the tested camera is, to a very high degree, central and the target planar.

Note that our findings are similar to the ones reported by Bergamasco *et al.* [10], but while they evaluate their method using an only nearly central camera, our evaluation is based on a highly central camera, as indicated by the slightly

increased evaluation error when using the unconstrained model. Hence, the improvement shown in table 4.1 on page 62 stems solely from the improved distortion modeling capability of the ray based model, and not from the non-central nature of the calibrated camera.

We also evaluate the influence of the number of calibration images on the calibration quality, see figure 4.6 on page 63. Interestingly, the number of calibration images does not have a large influence after some threshold, which is seven images for our approach and just three for the division model. We also show the reprojection error depending on the number of images in figure 4.7 on the facing page, which illustrates once more why the reprojection error of the calibration is a bad measure of the absolute calibration accuracy.

4.6 Discussion

In its current formulation the method is limited to imaging systems with a field of view below 180 degrees. Note that this limitation is a direct result of the two-plane parametrization of the ray based model, see section 4.2. While this formulation is convenient for expressing our calibration method, it should directly transfer to a more generic ray parametrization.

The method possesses several tuneable constants, mostly concerned with the generation of the calibration proxy. While the resultant calibration model seems to change little for different values, we evaluated only few combinations.

As shown in section 4.5 our approach allows high accuracy calibration of central and non-central cameras from the passive target, without the requirement of a perfectly manufactured target. Compared to previous approaches this simplifies camera calibration and avoids the necessity to select the right calibration model beforehand, which should make calibration easier to use and more robust, especially for specialist optics, like extreme wide-angle lenses.

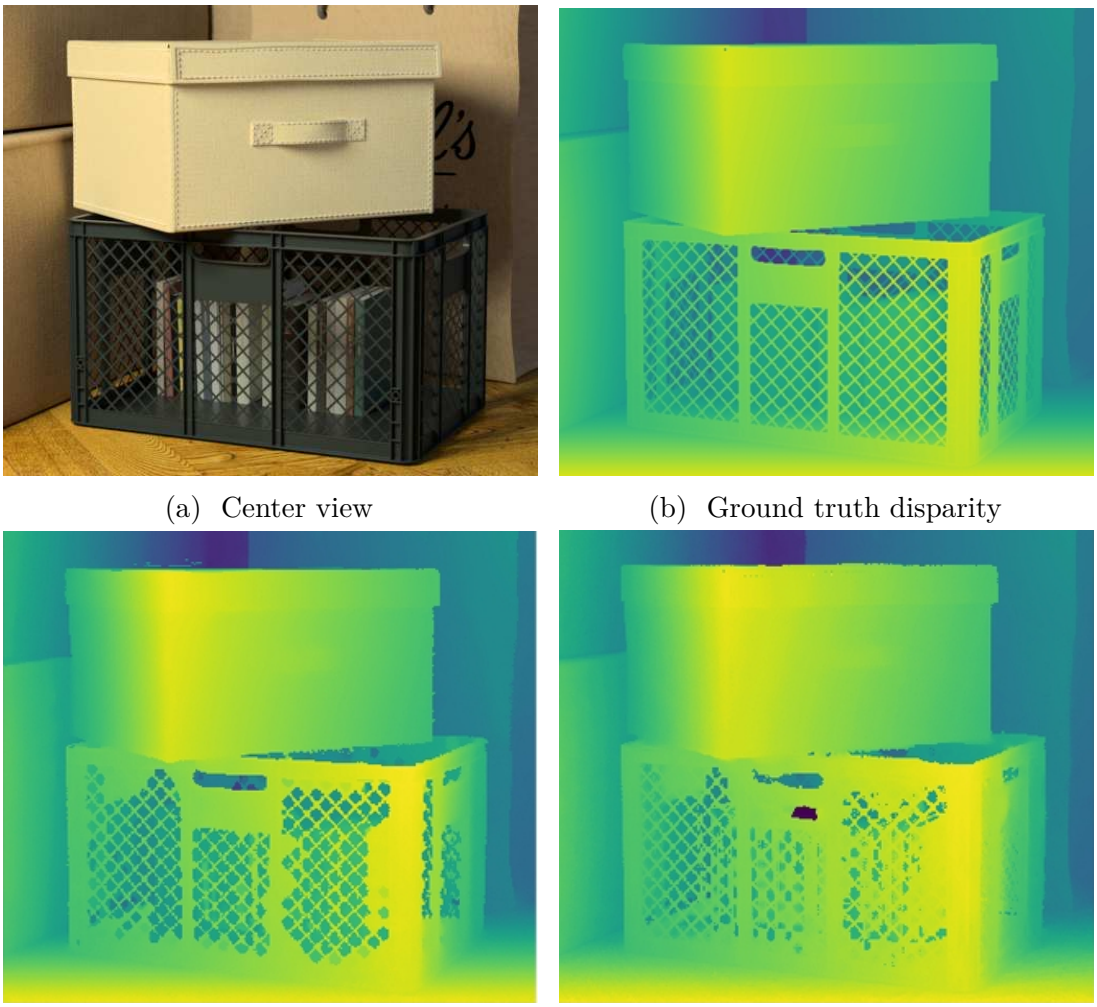
5 Light Field Depth Estimation

Depth estimation from multiple images is a central task in computer vision, with a long-standing history. Depending on the application area, different types of depth sensors are utilized, ranging from passive and active stereo cameras, over active depth cameras, to light field cameras. If depth accuracy is the most important factor, compared to *e.g.* financial budget or portability, then light field cameras might be the best choice. Light Field depth may be used for a broad range of scenarios, from special effects to industrial inspection, or autonomous driving scenarios. Alternative active sensors like time-of-flight cameras or Lidar may not always be viable due their active nature which leads to interference and limited range.

In this chapter a novel depth estimation approach for light field imaging is introduced, which is build around proper occlusion reasoning which improves both discontinuities in the estimated disparity map, and allows better regularization.

5.1 Outline

For related work please see section 1.3.3. This chapter is structured as follows. Section 5.2 explains core concepts of light field imaging and section 5.3 shows the occlusion problem and the importance of occlusion handling. Section 5.4 introduces the novel depth estimation method, with further details split into the modeling in section 5.4.1, occlusion reasoning in section 5.4.2, the data term in section 5.4.3, smoothness term in section 5.4.4, optimization in section 5.4.5 and the initialization in section 5.4.6. Finally, the method is evaluated on real and synthetic data, showing that this method outperforms the previous state-of-the-art, see section 5.5. In addition, two extensions are presented, implementing BRDF surface normal enhancement in section 5.6 and light field polarization imaging in section 5.7.



(a) Center view

(b) Ground truth disparity

(c) **OURS**
$BadPix_{0.07} = 10.8$

(d) SPO-MO, Sheng *et al.* [85]

Figure 5.1: **Improved reconstruction** through our inline occlusion handling approach, in comparison with Sheng *et al.* [85, SPO-MO]. Note the considerably improved reconstruction of the partially occluded content within the box and on the right side of the box. The improvement can also be measured quantitatively by the percentage of bad pixels (error > 0.07 px), here 10.8 for ours and 15.5 for Sheng *et al.*

5.2 Light Field Imaging

The term light field imaging is commonly applied to approaches based on the multi-dimensional sampling of a scene, using recording setups which can capture more dimensions than the two-dimensional projection of a perspective camera. This data driven approach is inspired by the plenoptic function [1], which describes a scene by the light passing through every point in space:

$$L(x, y, z, \theta, \phi, t, \lambda), \tag{5.1}$$

which gives the light intensity (radiance), where (x, y, z) is the 3D scene point, (θ, ϕ) are the angles giving the direction of the light, t is the time and λ is the wavelength. Note that this could still be extended with, for example, polarization information, to get an even more complete description of the scene. In computer vision, time is often regarded in discrete time steps, and wavelength is simplified to a few color channels, therefore it is not necessary to regard those as continuous dimensions, and they are normally dropped from the function, to reduce dimensionality to five dimensions. See figure 5.2 on the next page for a visualization:

$$L(x, y, z, \theta, \phi). \tag{5.2}$$

Furthermore, if we constrain the problem solely to the light leaving the scene through a defined window. This constrain allows us to simplify this to 4 dimensions, sampled at an intersection plane, see figure 5.3 on page 71:

$$L(u, v, s, t), \tag{5.3}$$

where (u, v) is the intersection points at the first plane and (s, t) the intersection at the second plane.

Note that while this is a common parametrization of the plenoptic function, in this work we will instead regard perspective 2D projections of the scene, acquired with an idealized pinhole camera, by translating the viewpoint along the 2D plane, see figure 5.4 on page 72. This also gives a sampling of the 4D plenoptic function. In this viewpoint (or *subaperture view*) parametrization the projection from scene to image space follows the form described in equation (2.8) from section 2.2:

$$\hat{x} = C(Rw + T). \tag{2.8 revisited}$$

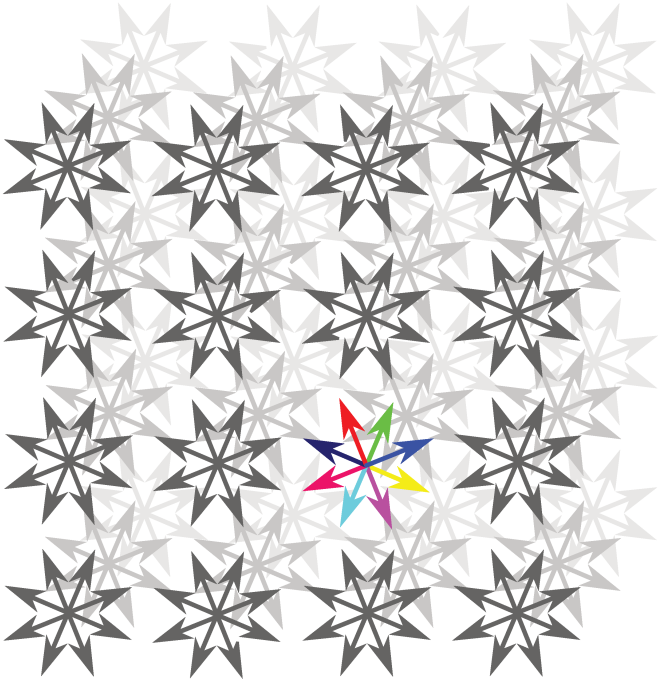


Figure 5.2: Visualization of the five-dimensional plenoptic function where the "star" locations represent the three dimensional spatial components and the arrows the two-dimensional angular components.

Assuming a scene coordinate system aligned with the camera where R is the identity and $T = (T_x, T_y, 0)^T$, then movements of the camera via T directly translate to image space movements as:

$$\hat{x} = C(w + T) \tag{5.4}$$

$$\hat{x} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} w_x + T_x \\ w_y + T_y \\ w_z \end{pmatrix}, \tag{5.5}$$

And hence:

$$\hat{x} = \begin{pmatrix} f_x w_x + f_x T_x + w_z c_x \\ f_y w_y + f_y T_y + w_z c_y \\ w_z \end{pmatrix}, \tag{5.6}$$

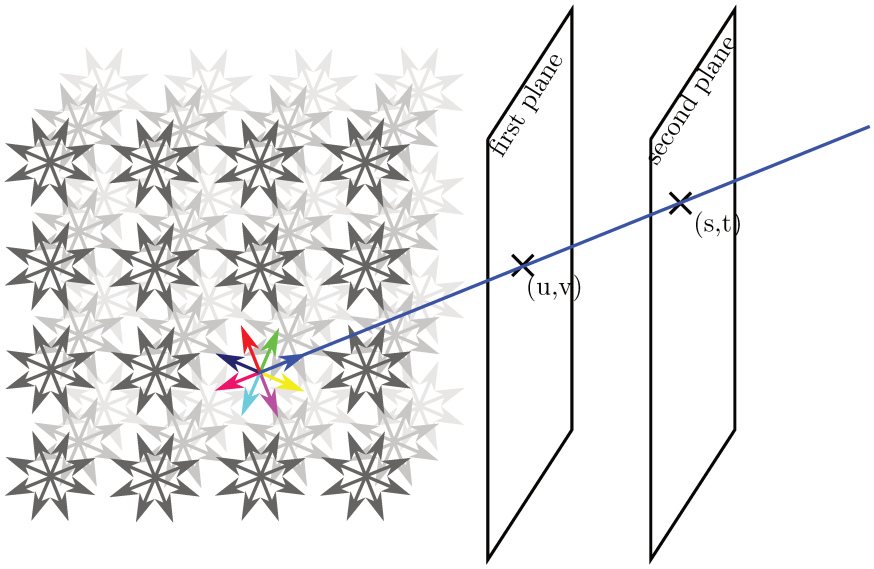


Figure 5.3: By assuming that the radiance stays constant outside of the scene, the plenoptic function can be parametrized as a 4D function, for example using a two plane intersection model as shown here.

applying h then gives:

$$x = \begin{pmatrix} f_x \frac{w_x}{w_z} + c_x + f_x \frac{T_x}{w_z} \\ f_y \frac{w_y}{w_z} + c_y + f_y \frac{T_y}{w_z} \end{pmatrix}. \tag{5.7}$$

This is convenient, as we can compare two viewpoints with different translations A, B:

$$x_A - x_B = \begin{pmatrix} f_x \frac{A_x - B_x}{w_z} \\ f_y \frac{A_y - B_y}{w_z} \end{pmatrix}. \tag{5.8}$$

If the spacing between cameras along a direction (here x) is expressed by the baseline $b = A_x - B_x$. Then the image feature movement can be used to estimate the distance z by rewriting equation (5.8) as:

$$z = \frac{f_x b}{d}. \tag{5.9}$$

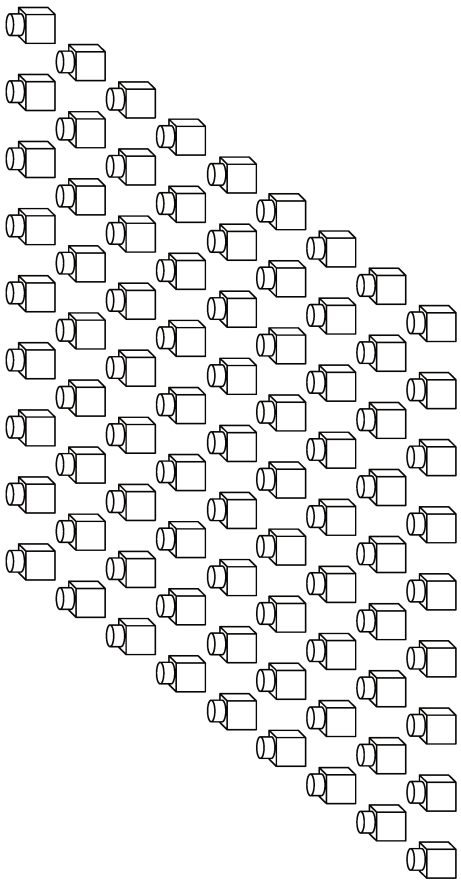


Figure 5.4: Viewpoint parametrization of a 4D light field using a 2D camera array. Each camera captures a 2D perspective projection of the scene from a single viewpoint, multiple viewpoints capture a range of samples from the angular and spatial domains of the full light field.

Here d is the disparity, the distance (in pixels) of the apparent motion of object points in image space, caused by the viewpoint shift.

This is well known from stereo imaging, but with a light field setup more than two viewpoints are available, compare figure 5.4, which allows the re-parametrization of the light field as an EPI, see section 5.4.1. EPIs are a convenient structure for light field depth estimation, as the orientation of lines in an EPI allows the measurement of the disparities, while at the same time the EPI enables reasoning about occlusions in the scene, where scene content occludes parts of the scene for some viewpoints, see also section 5.4.2. Occlusion reasoning

is not possible to this extent in stereo imaging, as occluded are only visible from one view, whereas in light field imaging they might be observable from multiple viewpoints.

5.3 The Importance of Occlusion Handling

Light-field imaging allows for highly accurate depth estimation, by sampling a scene from many viewpoints, compare section 5.2. Compared to stereo imaging, the oversampling increases depth accuracy and the large number of viewpoints reduces the chance of encountering a sample which is occluded in all other views. As for related tasks, such as stereo depth and optical flow, proper occlusion handling is essential for obtaining high-quality depth reconstructions. An inaccurate occlusion model will immediately reduce the reconstruction quality, since foreground and background samples are confused within the data-term around object boundaries. This is a well-known problem and virtually all state-of-the-art methods for light-field depth estimation implement some form of occlusion handling. However, they differ in the way how they perform this. Please see section 1.3.3 for the approaches used in related works.

We can classify occlusion handling approaches into three different paradigms to handle occlusion, each with a different level of complexity. At one end of the spectrum there are approaches which formulate an elaborate model for jointly estimating depth and occlusions, ideally for all views jointly. This explicit joint optimization has been formulated by Kolmogorov and Zabih [52], however their approach is prohibitively slow with existing solvers, even when restricting the problem to stereo and single pixel accuracy [95]. Hence, there are currently no practical realizations of such an approach for light-field imaging. At the other end of the spectrum, there are all the existing approaches to light-field depth estimation. In a nutshell, they employ a pre-processing step to filter out all potentially occluded pixels in each view. However, the way to achieve this differs. After this pre-processing step one, or sometimes multiple, cost volumes are derived explicitly or implicitly from the image data. The cost volumes are then used to derive the depth for (normally) the center view of the camera. The hope is that the cost volume is free of the influence of occlusion. Obviously, such a two stage procedure is sub-optimal for various reasons. One major problem is that wrongly discarded non-occluded pixels are lost for the remaining computation steps.

This chapter introduces a new way to incorporate occlusion reasoning in a more integrated fashion than existing approaches, and in this way makes the most use of the available data. To achieve this goal while keeping computation costs

down, we borrow from PatchMatch [6], which allows very fast optimization using a randomized approach and local optimization, and can be extended to make use of a data term and a smoothness term [73]. In our case both the data term and the smoothness term is subject to the occlusion information of neighboring pixels.

In effect, we continuously update the occlusion information during the processing, which means that it is always consistent with the estimated depth, and by virtue of this synchronization the occlusion information is implicitly improved with every iteration.

In the original PatchMatch the local errors directly sum up to a global energy which is implicitly minimized, as there are no local interactions. However, while we also perform only local evaluations and updates, because of the interaction between depth model and occlusion, these local updates do not give any guarantees with respect to the global error. By using PatchMatch we are able to achieve our goal of efficiently estimating a depth model where occlusion information does not have to be pre-computed. By doing so, we observe a substantial improvement in reconstruction quality, both qualitatively and quantitatively. Interestingly, our improvements are not only located at object boundaries, but also the quality of interior surface reconstruction improves. This stems from the fact that we can make better use of the available data than other methods, even those methods with a strong focus on regularization.

5.4 Method

Given the fact that the depth model which is being reconstructed implicitly contains the information required for proper occlusion handling, we formulate the cost function in a way that makes explicit use of the occlusion information encoded within the model, see section 5.4.2 for more details. This makes occlusion a first class citizen of the model.

This cost could in principle be optimized with some global optimization method. However, as the resultant optimization problem is highly ill-posed, this approach would probably be extremely slow, compare [52, 95]. Therefore, we base our approach on PatchMatch [6] to perform only local optimization, and introduce extra constraints into the cost term to avert suboptimal solutions arising from this fast, but globally suboptimal optimization.

Apart from the implications of the occlusion handling, the approach is formulated as a standard minimization problem with a cost based on a regularization term and a data term, where both are influenced by the occlusion handling.

5.4.1 Model and Data

The model used here is the disparity map of the central view. To simplify occlusion handling we confine the data to the subset of viewpoints shifted only horizontally or only vertically from the central viewpoint (cross-hair configuration), see figure 5.6 on page 77. The volume of the horizontal 3D subset can be sliced row-wise to obtain a set of epipolar plane images (EPIs, compare figure 5.5 on the following page), which represent the full information content of the subset. The central row of an EPI corresponds to a row of the center view, which directly maps to the same row in the disparity map. The same applies to columns in the vertical 3D subset. A single sample from the disparity map corresponds to a 2D line in the respective EPIs, where the slope of the line represents the disparity and hence encodes the depth, compare figure 5.7 on page 78. The cost function $E_i(d)$ for a single sample i of our model (a pixel of the center view disparity map D), based on the data term $\xi_i(d)$ and the regularization term $\zeta_i(d)$ is formulated as the cost associated with a disparity d, where the disparity map D is held constant for the evaluation of the sample:

$$E_i(d) = \rho \cdot \zeta_i(d) + \xi_i(d), \tag{5.10}$$

where ρ is a regularization weight.

5.4.2 Occlusion Handling

Compared to the methods in section 1.3.3, we obtain occlusion information from our depth model, and not via some heuristic external to the optimization. This simplifies our occlusion metric to a simple threshold θ_d. We consider a disparity sample d in the disparity map to be potentially occluded by any other sample d_i if $d_i - d > \theta_d$.

The actual decision whether a sample is occluded or not is performed during the evaluation of the cost terms, which means that updates to the model performed during an iteration of the optimization, directly affect the costs of all future evaluations, which speeds up the propagation of locally good solutions, as advocated by the PatchMatch algorithm [6].

5.4.3 Data Term

Because we only consider either horizontal or vertical camera movement, relative to the central view, only samples from the same row (or column, respectively),

Figure 5.5: **Epipolar Plane Images (EPIs)** are extracted from a linear 3D subset of the 4D light field, by extracting all rows (for a horizontal subset) and stacking them together, shown at the bottom. For the vertical stack the same is done with columns. Because the apparent motion of scene points between the different viewpoints depends on the depth of the point within the scene, the orientation of features in the EPI encodes the depth of the respective points. Note that the EPI shown here is pre-shifted so a disparity of 0 is not at infinity but rather within the scene, hence disparities may also be negative.

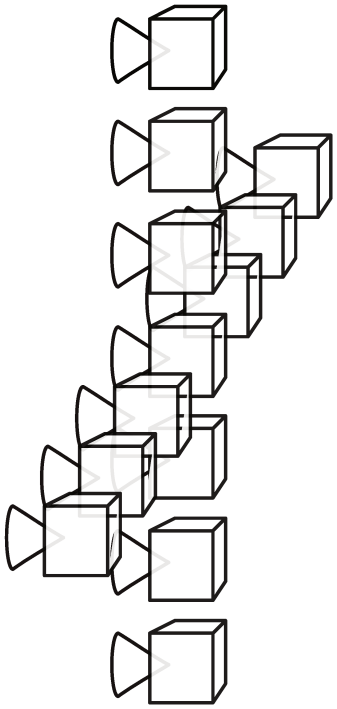


Figure 5.6: Multi-Camera cross-style light field setup. The cross configuration samples two 3D subsets of the full 4D light field in orthogonal directions, which allows sampling of most of the interesting effects of a scene, including observation of isotropic and anisotropic BRDF characteristics. It also gives good constraints for depth estimations as both purely vertical and purely horizontal image space features can be detected, something whic is not possible for stereo imaging.

can occlude any given sample in an EPI, compare figure 5.5 on the facing page and figure 5.7 on the next page. In the following we will always assume that we are looking at horizontal EPIs, but all statements apply to vertical EPIs via a corresponding 90° rotation of EPI, view and disparity map.

To evaluate the data error for some disparity d at location i in the disparity map, we sample along the corresponding line $\Gamma_{d,i}(s)$, see figure 5.7 on the following page, by evaluating $\Gamma_{d,i}$ for all rows s of the EPI. A sample $\Gamma_{d,i}(s) = x$ corresponds to a pixel position at the image coordinate (x, i_y) of view s. While i_y is an integer, x is a fraction, hence the actual pixel value $C_s(x, i_y)$ is derived by interpolation in the horizontal direction. To actually calculate the data error we generate all

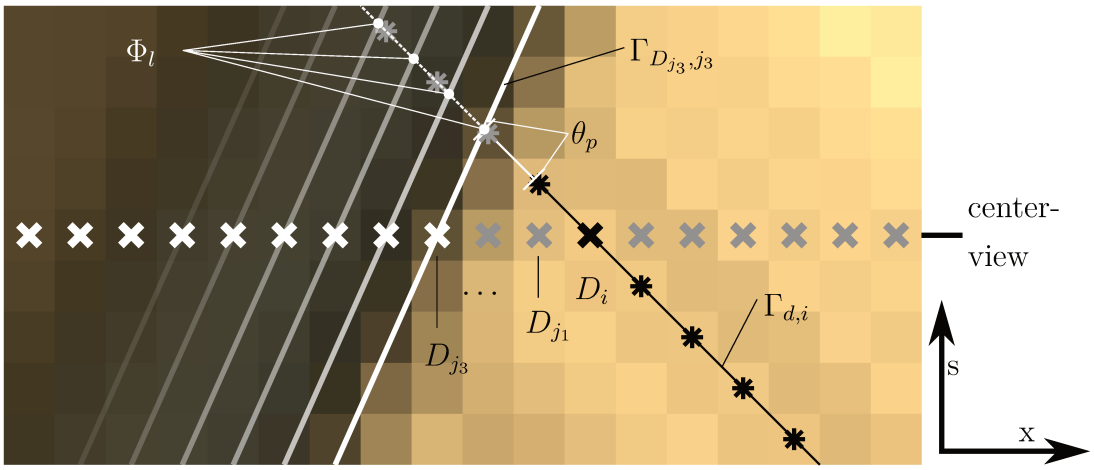


Figure 5.7: **Occlusion handling in an EPI:** The lines Γ are defined by the respective disparities D_j in the center view, represented by a cross ($\times$), while the EPI samples on $\Gamma_{d,i}$ are shown as star ($\ast$). From the intersections Φ_l (white dots), the one closest to the center view is obtained with Γ_{d,j_3}, hence all samples behind this point minus a safety distance if one pixel are disabled (grayed out).

intersections between $\Gamma_{d,i}$ and all other lines $\Gamma_{d,j}$ of the EPI which fulfill the occlusion condition in equation (5.11). Note that lines from samples to the left of i can only intersect above the center view, while samples to the right can intersect below. Given these left/right intersections as Φ_l and Φ_r, respectively, the occlusion term $nocc(s, \Phi_l, \Phi_r)$ is set to zero or one:

$$\text{nocc}(s, \Phi_l, \Phi_r) = \begin{cases} 1 & \text{if} \quad s < \phi - \theta_p \quad \forall \phi \in \Phi_l \\ & \text{and} \quad s > \phi + \theta_p \quad \forall \phi \in \Phi_r \\ 0 & \text{otherwise,} \end{cases} \tag{5.11}$$

The occlusion area is extended by one pixel from the intersection point, to avoid mixing of foreground and background when deriving the actual color sample $C_s(x, i_y)$ from the input view s via linear interpolation. Given the occlusion terms, the data error is simply the variance of all visible samples. We extend the previous definitions by the subscripts h and v to denote the horizontal and vertical EPI variants respectively (following terms with respect to a fixed sample i and a fixed disparity d):

$$\xi'_i(d) = \frac{\begin{array}{c}\sum_s(\mu - C(\Gamma_h, s))^2 \cdot nocc_h(s, \Phi_{h,l}\Phi_{h,r}) + \\ \sum_t(\mu - C(\Gamma_v, t))^2 \cdot nocc_v(t, \Phi_{v,l}\Phi_{v,r})\end{array}}{\begin{array}{c}\sum_s nocc_h(s, \Phi_{h,l}\Phi_{h,r}) + \\ \sum_t nocc_v(t, \Phi_{v,l}\Phi_{v,r})\end{array}}, \tag{5.12}$$

where μ is the mean of all unoccluded samples for (i, d).

To avoid failures due to the local nature of our approach, we also threshold the data term on the number of unoccluded samples, and set the error to infinity if less than θ_o samples are unoccluded, because otherwise, moving individual samples (incorrectly) towards the background can reduce the variance in flat areas, by reducing the number of unoccluded samples.

Even with this occlusion constraint there is a second case where the local solution can substantially deviate from the correct depth. This can be observed on purely horizontal or vertical structures in the scene. For such structures the data error is zero for one direction, hence, if *e.g.* for a vertical structure, the vertical component of the data term is zero, then if a large connected block of the vertical structure is moved into the background, the remaining horizontal component also becomes zero because we observe only a single sample in that direction. We protect against this by checking, for each candidate, whether the chosen disparity leads to a single pixel wide background structure, as measured by θ_d over a range of 10 pixels. If such a case is detected, the error is set to infinity.

5.4.4 Smoothness Term

For a disparity sample d at location i in the disparity map, the smoothness error is defined by:

$$\zeta_i(d) = (d - \Omega_i(d))^2 \tag{5.13}$$

Where Ω is a smoothing filter based on the bilateral filter. This filter smoothes the disparity map using a weighted mean, with weights derived from the color and disparity difference against a central sample. The filter uses hard thresholds θ_d and θ_c to determine which samples are allowed to influence the smoothing, which gives well-defined borders without disparity bleeding. Given the color values of the center view as C, and the current disparity map as D, the smoothing filter Ω is given by:

$$\Omega_i(d) = \frac{\sum_j \lambda_{i,j}(d) \cdot D_j}{\sum_j \lambda_{i,j}(d)}, \tag{5.14}$$

where j indexes a 7×7 window around i.

The relative weight $\lambda_{i,j}(d)$ of the disparity map sample D_j is calculated depending on the color difference $\Delta_{i,j} = \alpha|C_i - C_j|$ and the disparity difference $\delta_j(d) = \beta|d - D_j|$ between the sample j and the central sample i, with α and β as parameters which steer the relative weighting of color and disparity differences. The weights are calculated as

$$\lambda_{i,j}(d) = \max\{\epsilon_d, \sqrt{\Delta_{i,j}^2 + \Delta_{i,j} \cdot \delta_j(d)}\}^{-1}, \tag{5.15}$$

if $\Delta_{i,j} \leq \theta_d$ and $\delta_j \leq \theta_c$, and

$$\lambda_{i,j}(d) = \max\{\epsilon_c, \sqrt{\Delta_{i,j}^2 + \delta_j^2(d)}\}^{-1}, \tag{5.16}$$

if $\frac{\Delta_{i,j}}{\beta} > \theta_d$ and $\delta_j \leq \theta_c$. Otherwise $\lambda_{i,j}(d)$ is set to zero. The thresholds θ_d and θ_c set the maximum difference for disparity based weighting (if $\frac{\Delta_{i,j}}{\beta} \leq \theta_d$ and $\delta_j \leq \theta_c$) or color based weighting (if $\frac{\Delta_{i,j}}{\beta} > \theta_d$ and $\delta_j \leq \theta_c$).

The ϵ are used to provide damping against zero differences, and ϵ_c also provides some adaption to noise in the input images, using $\epsilon_c = \epsilon_d + \theta_e \cdot E'_i(d_0)$, where E'_i is identical to E_i, aside from changing ϵ_c to $\epsilon_c = \epsilon_d$. Hence, $E'_i(d_0)$ is the initial error at this iteration, using the initial disparity d_0. This increases the minimal blurring of the smoothing filter, when no good candidates where found in the previous iteration – which after a few iterations is mostly due to noise in the input images.

The crucial part is the usage of the current disparity candidate d within the filter, which lets the smoothing filter adapt to the value of the candidate. The current disparity at i from the model, D_i is not used during the evaluation. This means that the smoothness term can switch, for example at an object border, from averaging over the foreground to averaging over the background, depending on the evaluated disparity candidate, as shown in figure 5.8 on the next page.

The thresholds encourage the smoothing according to the model (*i.e.* disparity map) by making the disparity difference the dominating weight term for small disparity differences ($\frac{\Delta_{i,j}}{\beta} \leq \theta_d$). The color differences play a secondary role and encourage smoothing along similar colors. At the same time the hard thresholds mean that the weight is quickly set to zero if the differences in color and/or disparity become too large, ensuring that only those samples are taken into account for which it is likely that they belong to the same object, both from the color and the disparity similarities.

The simple smoothness term as described above limits the estimation accuracy in two ways. Firstly, the method tends to over-smooth at object edges when both

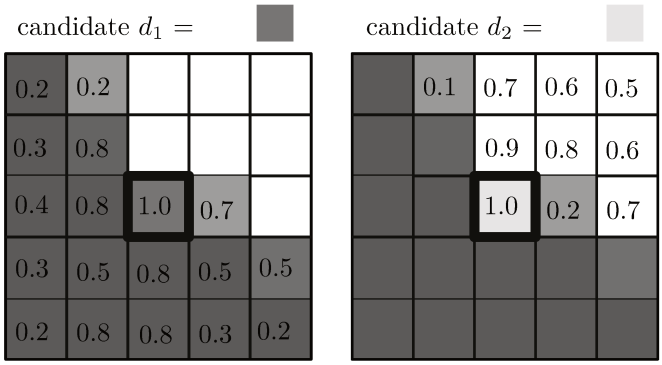


Figure 5.8: **Switching behavior of the smoothness term.** The two grids represent the identical neighborhood around a central disparity sample d, indicated by the brightness of the cells. Depending on the value of a candidate d_i, the weights, given as numbers within the cells, change according to equation (5.15) and equation (5.16), which by design leads to a distribution which generates a smoothing of those samples most similar to the central candidate in both color and disparity.

sides of the object are visible, because the edge of the object will be averaged with the neighbors from both sides. Secondly, planes with a steep inclination tend to show staircase artifacts, as the thresholding in the filter encourages areas to be piecewise planar.

We extend the filter to preserve normals and planes separately. In the smoothing filter, consistent normals between the central sample i and some other sample j are detected by comparing the local gradients in D. If the gradient difference is below θ_g, then D_j is corrected by this normal when it is used in equation (5.14).

For planar surfaces we add a metric which detects purely planar surfaces, by taking four samples around the central sample, located at the corners of a square with a size of 11×11, and fitting a plane through these four corners. If the residual from the fit is below θ_f and the distance between the plane and disparity candidate are below θ_d we evaluate the plane at i and use this result instead of Ω.

Both of these metrics are applied with a damping factor, where the correction with normal and plane is weighted with the original smoothing filter with a weight of 0.5 to prevent overshooting.

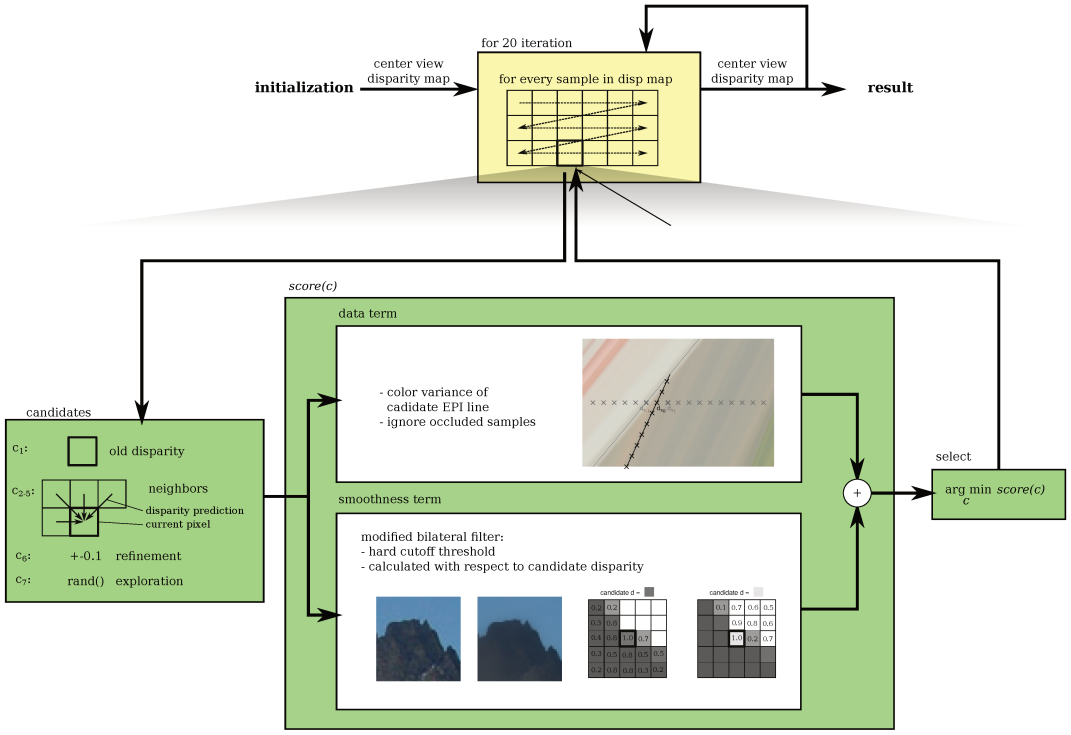


Figure 5.9: Flow chart visualizing the data flow of our method. After initialization all pixels of the disparity map are iterated. For every pixel in the disparity map several candidate disparities are evaluated, and the one with the lowest sum of smoothness and data error is chosen to update the disparity map. This process is repeated 20 times, changing the direction of processing between each iteration.

5.4.5 Local Optimization

Both the data term and the smoothness term are formulated with a strong focus on correct occlusion handling with hard thresholds in disparity and color differences. While this encourages well-defined borders in the model, it makes the problem harder to optimize, owing both to the sudden onset of the influence of samples, and to the complex interaction between samples due to occlusion. Pre-calculating the error terms for a number of discrete disparity labels and building a cost volume is also not possible, as both terms deliberately depend on the current state of the model. Therefore, we base our method on PatchMatch [6]. The method iterates the disparity map and, at each sample, calculates the local

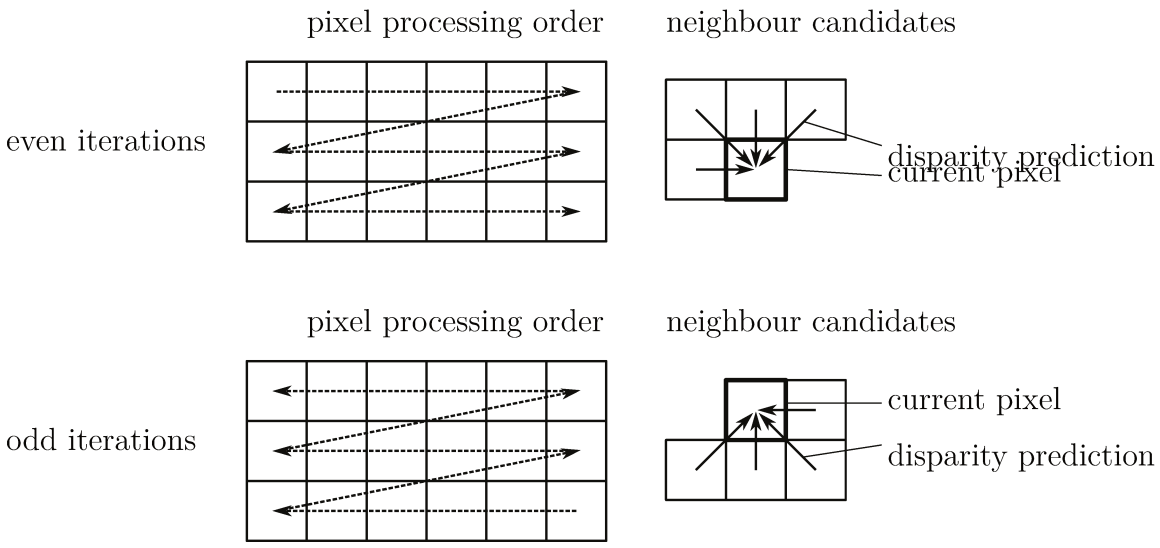


Figure 5.10: Candidate disparities are predicted from the direct neighbors already processed in this iteration. Processing reverses direction after each iteration, so good solution can propagate in all directions.

error E_i for the current disparity d_0, as well as for several disparity candidates. If any of the candidates has a lower error over the previous solution, the model is immediately updated, which allows propagation of locally good solution.

We use four predictors to provide the disparity candidates which are evaluated with the local error term, also visualized in figure 5.9 on the facing page.

Propagation: Depending on the iteration number, the solver iterates over the disparity map either left-to-right and top-to-bottom, or the reverse, see figure 5.10. The disparities of all neighbors (either direct or over the corner) which were already processed in the current iteration are used as candidates for evaluation. As the model is always directly updated when a lower error is found, an improved estimate at one sample will directly be used in the data and smoothness term of the next sample, within the same iteration. Hence, as the improved disparity at a sample is provided as a candidate to the solver for the next sample, good solutions can quickly spread over the whole disparity map.

Random Improvement: At each iteration, candidates d_i are generated by sampling u from a uniform distribution between -1 and 1 as:

$$d_i = d_0 + \tau \operatorname{sign}(u) u^2 \tag{5.17}$$

where τ is the parameter which steers the max range of the refinement. The quadratic term ensures that smaller changes are sampled with a much higher frequency than larger ones.

The following two predictors are only activated if the error of the current model is above an activation threshold θ_a.

Random Neighbor: For some scenes a feasible candidate might be not directly adjacent but further away, *e.g.* when a surface is partly occluded by some detailed foreground object, like a smooth background behind the branches of some plant. For this reason we also use distant neighbors, by sampling uniformly within a range of $\pm 15\,\mathrm{px}$.

Random Guess: Finally, we also sample randomly from the valid disparity range.

5.4.6 Initialization

If the solver is simply initialized, either with a uniform or random disparity, the solver is not able to converge to a good solution in most cases, because it will tend to preserve clumps of uniform disparities as long as the data term doesn't force a change. This means that a wrong initialization on low structured regions will be preserved. To resolve these ambiguous, low textured areas we use an initialization which is based on line fits through zero crossings.

For each EPI we calculate the zero-crossings of the second derivative in the horizontal direction to collect a number of feature points, and store their location. In a second step we regard all these feature points on the bottom row of the EPI and iterate over all feature points within the disparity range of the top row. Using these two points we fit a line RANSAC-style, and compare the two points to the points on the rows in between, using configurable thresholds for the maximum horizontal distance and the maximum gradient difference.

The resulting lines are mapped into the center disparity map to provide a sparse but highly accurate initialization of disparities. This map contains now most of the foreground structures of the light field, *e.g.* features that are visible from

θ_d	$0.05K$	θ_g	$0.025K$	θ_f	$0.01K$
θ_c	3	θ_o	$0.25V$	θ_a	0.01
α	0.15	β	20	ϵ_d	0.5
ρ	$0.0375I$	τ	$0.2K$	θ_e	400

Table 5.1: List of parameters used for all results but figure 5.11 on the next page, where V is the total number of views, K the disparity range of the scene and I the current iteration number.

either all horizontal or all vertical views. To retrieve a dense initialization of our model – the center view disparity map – we now interpolate the missing samples from the sparse disparity map. For each missing sample we check for the next samples in the sparse map in horizontal and vertical direction, and weight them with respect to the color difference between the sample which is to be calculated and the color of the respective sample from the sparse map. Finally, a median filter is applied on the dense map.

The result of the initialization is a disparity map where unstructured areas are interpolated from their borders, while structured areas are smoothed. Because the solver can easily fill in areas where structure exists, but will tend to leave smooth areas relatively untouched, this initialization already defines the performance for unstructured areas in the input data.

5.5 Experiments

We have tested our method on several light-field data sets, including real and synthetic data. In the following we describe the results in more detail and demonstrate the improved occlusion handling, see figures 5.1, 5.11 and 5.12 on page 68, on the next page and on page 87, but also the excellent surface regularization, see figures 5.11 to 5.13 on pages 86–88. These improvements owe in part to the improved utilization of data from the input light field, as we discard less information due overzealous occlusion handling, as well as to the improved detection of object borders. More results of our method are available on the website of the 4D Lightfield Benchmark [47]. All results presented here use 20 iterations and, apart from figure 5.11 on the next page use the parameters shown table 5.1.

(a) center view (b) **OURS** (c) OFSY_330/DNR, Strecke *et al.* [90]

Figure 5.11: **Disparity estimates on the truck data set [96]**, which is challenging due to the large amount of noise, therefore (b) was computed with a version of the data set scaled down to half size in the spatial domain. Note that although our method uses half size images, the reconstruction is much more detailed, see for example the rope at the top left, or the structure below the driver cab. Smoothing is also improved, although some artifacts remain, like the rough ground before and behind the truck, or the "fireflies" around some object edges. The hole at the back of the cargo area is wrong with both methods because there is a specular reflection visible from several viewpoints.

5.5.1 Qualitative Results

In figure 5.11 we show our results on the truck scene from the (new) Stanford Light Field Archive [96]. For comparison, we also show the result of Strecke *et al.* [90] (OFSY). While the results leave room for improvement, the detail reconstruction shows the effectiveness of the occlusion handling. At the same time the regularization is also improved, which is otherwise a strength of OFSY, compare figure 5.12 on the next page.

5.5.2 Quantitative Results

The quantitative evaluation is based on the public 4D Lightfield Benchmark by Honauer *et al.* [47]. The benchmark does not report a single score, but instead calculates 12 different error metrics, compare figure 5.12 on the facing page, which consider a range of different failure cases, using well-known global metrics like *BadPix* and *MSE*, but also surface quality metrics, and more specific error metrics, like fine thinning/fattening. For details please refer to their paper [47] and the

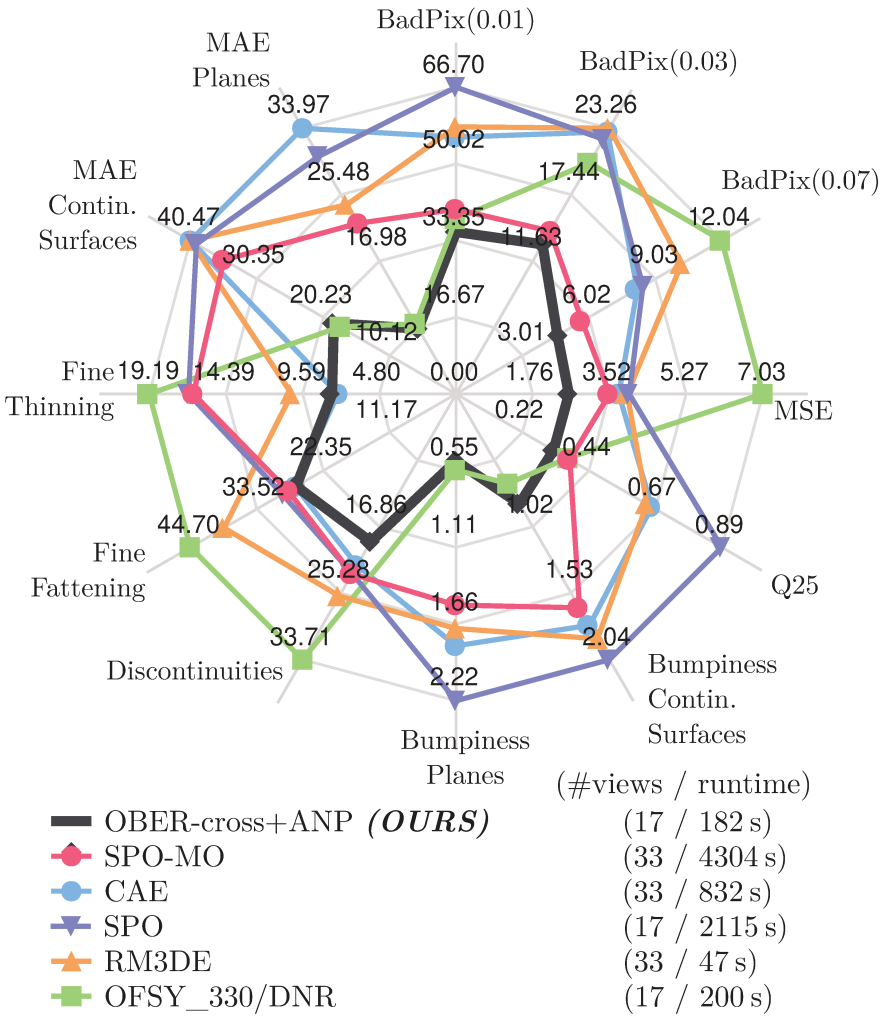


Figure 5.12: **Mean errors over all twelve benchmark scenes**, evaluated with the twelve error metrics of the 4D Lightfield Benchmark [47] and visualized on a radar chart. The legend gives the number of viewpoints and the (approximate) runtime. All metrics are expressed as an average error over twelve data sets. Lower values are better, and located closer to the center. As we can see our method (OBER-cross+ANP) is located closest to the center on average, and manages an improvement over the previous state-of-the-art on most metrics, without exposing a specific weakness. The main challengers which surpass our method in some metric (CAE and OFSY) manage so only by accepting subpar performance on other metrics.

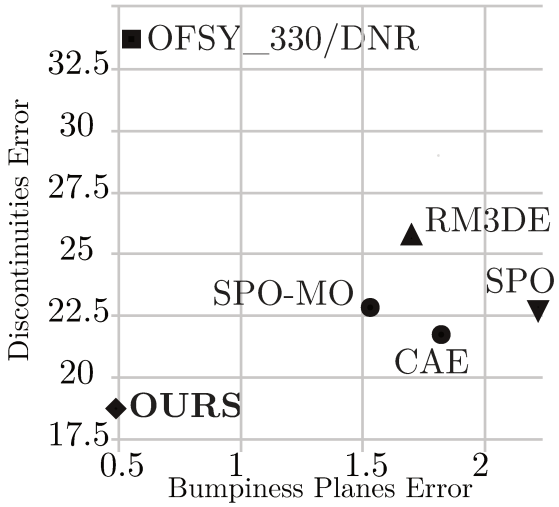


Figure 5.13: **Trade-off between smoothing and object border preservation**, comparing the *Discontinuities* metric with the *Bumpiness Planes* metric [47]. Results are averages over all 12 benchmark scenes. Note how the good smoothness score for OFSY reflects the focus on the regularization, while the other methods are optimized towards correct object borders. Our method leads both metrics, making the trade-off obsolete.

benchmark survey [48]. The benchmark is performed by generating disparity maps for 12 scenes, 8 of which have publicly available ground truth disparity, while for 4 scenes the ground truth is kept secret. Algorithm results are uploaded to a web-service and all results, including ours, are available on the benchmark website [47] – our method is abbreviated *OBER-cross+ANP*.

We report our results in comparison to the state-of-the-art, as represented by the top five published methods, when sorted by the average $BadPix_{0.07}$ score, as of 2017/11/11. The averaged errors over all 12 scenes are shown in figure 5.12 on the previous page. Note that our method takes the lead for 9 of the 12 error metrics, and is close behind for the remaining 3.

This is even more remarkable if we consider that several of the error metrics are often traded in against each other, as is the case for bumpiness versus discontinuities and for fine fattening versus fine thinning, which have a strong tendency to revert the order of the methods between the respective error metrics.

Indeed, by plotting the *Discontinuities* metric, which gives the errors around depth discontinuities, and one of the smoothness metrics, like *Bumpiness Planes*,

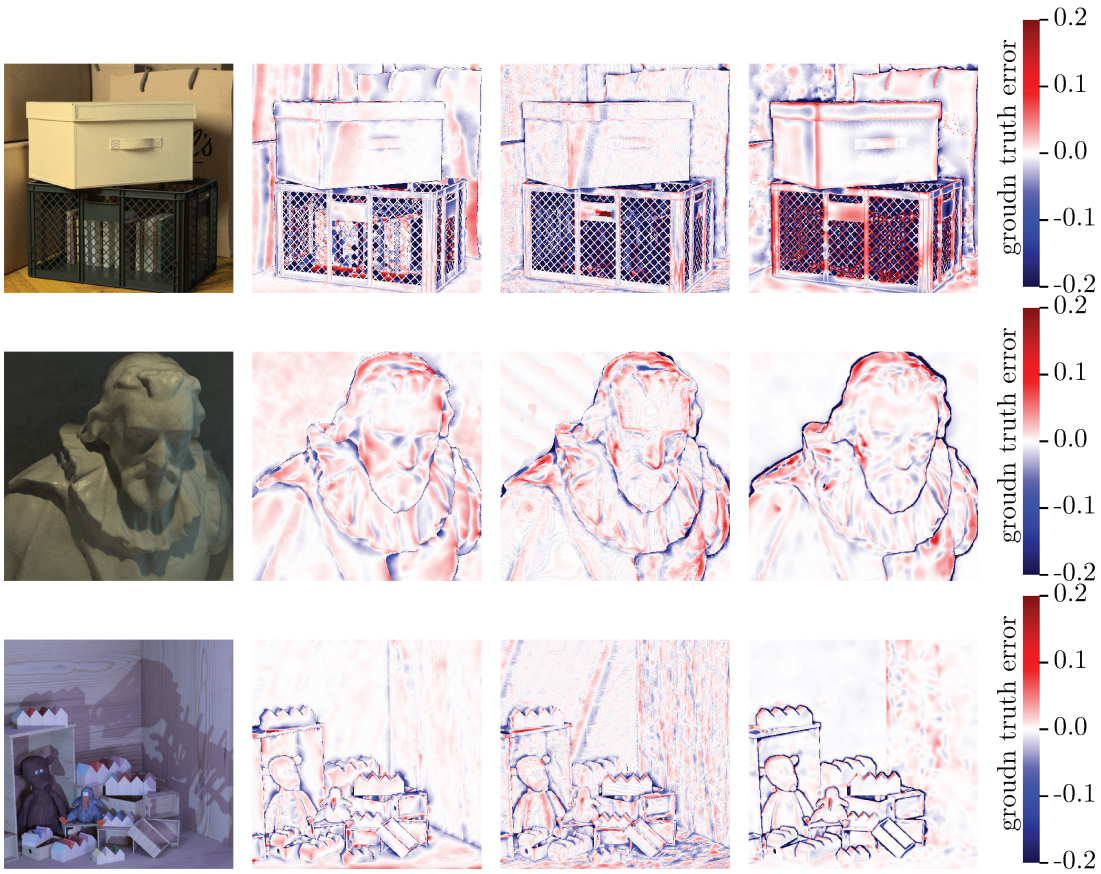


Figure 5.14: **Results from four scenes** of the 4D Lightfield Benchmark [47]. On the left is a view from the scene, followed by the visualization of the deviation against the ground truth of our method against three state-of-the-art methods. Our results show improvements especially around occlusion areas (*e.g.* content of boxes and borders of objects), but also displays excellent surface regularization.

we can directly evaluate the trade-off between smoothing and preservation of object boundaries, see figure 5.13 on page 88. As we can see, all tested methods fall into one extreme, favoring either border handling over smoothing. However, our method manages to not only find a favorable trade-off, but instead completely dominate the other methods on both of these metrics.

5.6 BRDF Based Surface Normal Estimation

Compared to previous approaches, see section 1.3.3, the method presented in this work is based on improving the modeling capabilities by improving occlusion handling. However, the approach is still formulated under the constraint of color constancy, which means that object points are assumed to appear with the same color independent of the view direction, aside from occlusion. However, most real world materials deviate from this assumption and their appearance changes depending on the orientation of camera, surface and light source.

One approach is to explicitly model the characteristics of light transport at the surface. Physically accurate models for reflection, as well as approximations of these models are well known from computer graphics. Fitting such a model from light field data is possible and can be combined with the depth estimation framework introduced in this work, see [42]. However, this modeling requires a known light source positions, which might be difficult to achieve in reality, especially in outdoor scene where there might not be one primary illumination source. This extension was researched by Marcel Gutsche and a detailed description can be found in [41, 42].

5.7 Light Field Polarization Imaging

Depth from polarization exploits polarization cues to improve scene reconstruction. For related work see section 1.3.4. Such an approach requires polarization information, either by recording a static scene multiple times using a linear polarization filter in front of the optics, or by using a Bayer-pattern style polarization sensor, see figure 5.15 on the next page and [103]. Compared to the previously required multiple cameras or multiple exposures, the Bayer-style polarization image sensor provides a huge increase in usability of polarization imaging, because polarization information can be recorded from a single exposure of a single camera, where previously at least three exposures were required. These kinds of sensors have only recently become available [103], and cameras are now available from

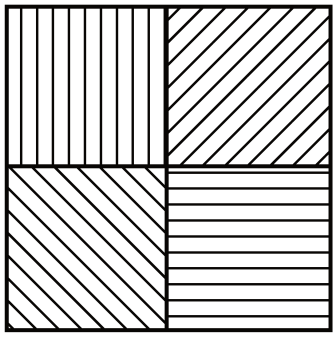


Figure 5.15: Visualization of a Bayer style polarization image sensor. The lines show the direction of the polarization filters in front of the individual pixels. The image shows the 2×2 base pattern, which is tiled to achieve the desired resolution.

several distributors. The Bayer-style polarization image sensor consist of a regular CMOS image sensor where a repeating 2×2 pattern of linear polarization filters are placed in front of the individual pixels, compare figure 5.15. This layout enables spatial multiplexing of polarization information, and multiple polarization channels are handled in the same manner as multiple color channels for a Bayer RGB image sensor.

5.7.0.1 Surface Orientation from Polarization Cues

The core observation is that incoming non-polarized (diffuse) lighting will be linearly polarized with a phase and magnitude depending on the surface orientation and the camera position, compare [49].

When a surface is illuminated by unpolarized ambient light, the brightness I recorded from an image sensor follows:

$$I(\varphi_{pol}) = \psi + \omega \cos(2(\varphi_{pol} - \varphi)) \tag{5.18}$$

and can be recovered from three independent measurements of the intensity of a pixel under three different orientations φ_{pol} of the polarization filter. Here ψ gives the average signal and ω the amplitude, while φ_{pol} is the known polarizer angle and φ is the unknown azimuth angle. This means the estimation of the azimuth angle does not require any further knowledge of *e.g.* material properties. However, the azimuth angle is still ambiguous as, from equation (5.18) it follows that φ and $\varphi + \pi$ give the same signal.

Given the estimated polarization curve there are two ways to estimate the degree of polarization, first from the actual signal:

$$\varrho = \frac{\omega}{\psi} \tag{5.19}$$

and secondly from the surface properties:

$$\varrho = \frac{(n - \frac{1}{n})^2 \sin^2 \vartheta}{2 + 2n^2 - (n + \frac{1}{n})^2 + \sin^2 \vartheta + 4 \cos \vartheta \sqrt{n^2 - \sin^2 \vartheta}}. \tag{5.20}$$

Note that this formula is only valid for dielectric surfaces. n is the material specific refractive index which has to be known beforehand, while ϑ is the zenith angle.

These constraints cannot be directly used to estimate depth, as only surface orientations can be calculated, which must be integrated to generate a closed surface. However, direct integration leads to large errors from three sources. Firstly, local errors will accumulate over the integration, secondly, the surface orientation contains the φ ambiguity, and thirdly, discontinuities cannot be reliably detected. Hence, polarization information is always combined with additional measurement modes, for example from a Kinect structured light sensor as shown by Kadambi et al. [49], or by combining multi-view stereo constraints [19] or assuming a uniform color to exploit shading constraints [89].

5.7.0.2 Approach

In this section we will show a proof-of-concept extension to our depth estimation framework which incorporates polarization constraints into the error terms. The idea is that explicit handling of the φ ambiguity is not necessary when implemented as an additional data error in the optimization, as the optimizer will simply pick the solution which minimizes the sum of all errors, hence resolving the ambiguity according to other cues from the data error or smoothness term. Note that in the following we will always refer to the parameters of single pixels, but the operations are normally applied for all pixels independently.

5.7.0.3 Input Data

In this approach we will assume per pixel polarization information in the form of intensity values stored as four color channels, which correspond to four orientations of a linear polarizer $(0, \frac{1}{4}\pi, \frac{1}{2}\pi, \frac{3}{4}\pi)$. For how this data is acquired in reality see section 5.7.0.8.

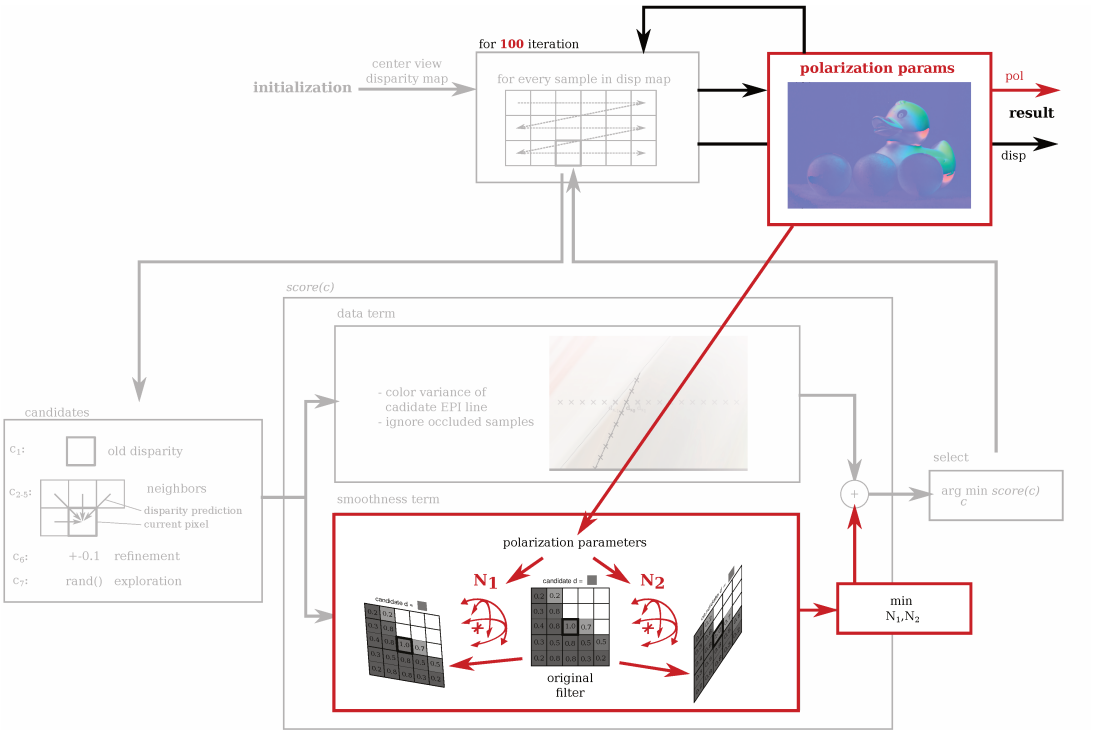


Figure 5.16: Flow chart visualizing the processing chain when using the depth estimation framework with polarization images. In addition, to the normal iteration scheme, compare figure 5.9 on page 82, polarization information is obtained between iterations, and is used to correct the smoothness term with the calculated normals.

5.7.0.4 Polarization Estimation

Polarization information is gathered between the regular iterations, compare section 5.4.5 and figure 5.16. After each regular iteration all samples of the disparity map are iterated again, but this time instead of calculating an error according to equation (5.12) the samples are collected and used to estimate the parameters of equation (5.18). Because this problem is overconstrained, a least-squares solver is used which greatly reduces the noise of the estimation, compare figure 5.19 on page 99. After estimating ω, ψ and φ, the degree of polarization is calculated according to equation (5.19) and using this the zenith angle ϑ is calculated numerically using equation (5.20). Note that this requires the refractive

index to be known, at least approximately. In the current implementation a constant value is used, which has to be supplied by the user before processing.

5.7.0.5 Modified Error Terms

Figure 5.16 on the preceding page visualizes the data of the processing with polarization information. Please compare with figure 5.9 on page 82 to see the differences to the original processing. The polarization information supplies cues about the surface normal, hence the data term is left as is, and only the smoothness term is modified. The regular smoothness term calculates an error against a smoothed version of the surface. As the polarization information supplies an (ambiguous) surface normal, the original smoothness term is used in nearly the same way as before, however the surface normal is converted to a disparity gradient following equation (2.1) and this gradient is applied to all samples before being evaluated by the smoothness term.

This means the smoothness term does not calculate an error against a smoothed version of the local neighborhood (as before), but an error against the average of the local surface estimated from a neighboring sample using the surface normal from the polarization estimation. In a way this can be interpreted as tilting the plane of the smoothing term from perpendicular to the camera z axis to perpendicular to the estimated surface normal. The φ ambiguity is resolved by using the minimum between the two possible estimates (φ and $\varphi + \pi$).

5.7.0.6 Normal Based Extrapolation

Because the normal is only used in the smoothness term, it can only extrapolate the surface structure from neighboring samples. However, for smooth, untextured surfaces the initialization is already quite wrong (due to missing structure), hence the normal guided smoothing is still not able to recover the correct shape. Hence, integration into the estimation framework requires several additional processing steps, to perform a normal guided extrapolation from the approximately known borders which will be detailed in the following.

Detecting Polarization Areas First, one regular iteration of the depth estimation is performed, see section 5.4.5. Second, using the estimated disparities, polarization parameters are estimated per pixel, as described in section 5.7.0.4.

Then the image space gradients of all images are calculated in x and y direction using central differences ($(-1\ 0\ 1)$ and $(-1\ 0\ 1)^T$).

Using the disparity map after the first iteration, the image space gradients are averaged along the corresponding EPI lines, taking occlusion into account as defined by equation (5.11). The result is the center view gradient map G. Problematic areas are detected using the degree of polarization ϱ and the image gradient and stored as valid disparities map W as:

$$W = \begin{cases} 1 \text{ if } \frac{G+10}{\varrho} > 20 \\ 0 \text{ otherwise} \end{cases} \tag{5.21}$$

Extrapolation Map Calculation At the beginning of each iteration the valid processing samples map D is calculated from W using a dilation dil_1 with a radius one box kernel as:

$$D_n = \begin{cases} 1 \text{ if } W_{n-1} = 1 \\ 1 \text{ if } \text{dil}_1(W_{n-1}) = 1 \text{ and } \varrho_{n-1} < 0.02n \\ 0 \text{ otherwise} \end{cases} \tag{5.22}$$

Where n is the current iteration. This approach extends the processed area from the border and from low polarization areas towards the center and towards high polarization areas.

At the end of each iteration W is set equal to D.

Extrapolation Map Usage During estimation, the valid processing samples map D defines which samples are processed at all (if D is zero a sample is completely skipped). The valid disparities map W defines whether a disparity sample is used during calculation of the smoothness term, see equation (5.13). If W is zero for a sample this sample will not contribute at all to the smoothness term calculations. This scheme allows the extrapolation of a new border of disparities from the valid disparities of the last iteration, completely discarding the noisy initialization. Because masked out disparity values are only used after they have been estimated from their neighbors, wrong initializations on smooth surfaces cannot influence the final estimation, and are instead slowly filled in from the borders. In addition, the threshold in equation (5.22) prioritizes processing of low inclination areas over steep inclinations, where errors in the polarization processing have more influence. Hence, more frontal regions are processed first which gives better results as in the end the areas are filled in from the borders and errors accumulate over the infilling.

Iterations Because the polarization normal based infilling converges much slower compared to the regular processing, the iteration count is increased to 100 iterations. This gives enough time to fill areas with up to 200 pixel diameter, which was enough to process the tested data sets.

5.7.0.7 Limitations

In the current implementation the samples added to the least-squares solver which estimates φ from equation (5.18) are used equally, as if they were originating from the central camera. However, camera pose is known at this step, hence a correction using the actual camera pose would be possible, which should lead to improved estimates.

Also, the current approach assumes a fixed refractive index. As in reality the refractive index depends on the material, this will lead to some errors in the estimate.

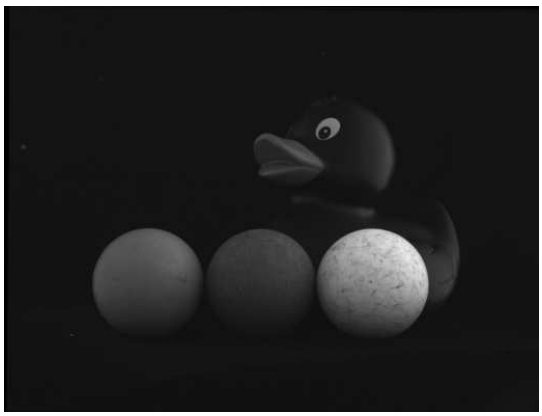
5.7.0.8 Results

The following results were obtained using a four direction Bayer style polarization sensor. The test scenes were recorded using very diffuse unpolarized illumination using a light box. 4D Light fields were recorded using a 2D translation stage and calibrated and rectified using the approach described in chapter 4.

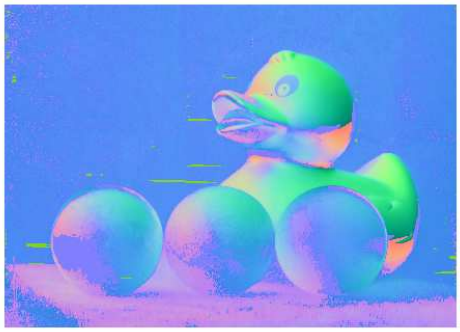
Because of the high amount of noise, the rectified images were scaled to half the resolution and then processed with the approach described above, which results in the meshes shown in figure 5.17 on the facing page. Results for more scenes are shown in figure 5.18 on page 98. From the results it can be seen the polarization processing works fine for some scenes, like figures 5.17 and 5.18(f) on the facing page and on page 98 but fails for others, see figures 5.17 and 5.18(d) on the facing page and on page 98.

There are several possible reasons for the failures. The refractive index n was not known and simply fixed at 1.5 for all scenes. In addition, equation (5.20) is only valid for dielectric materials, which might explain the failure in figure 5.18(d) on page 98.

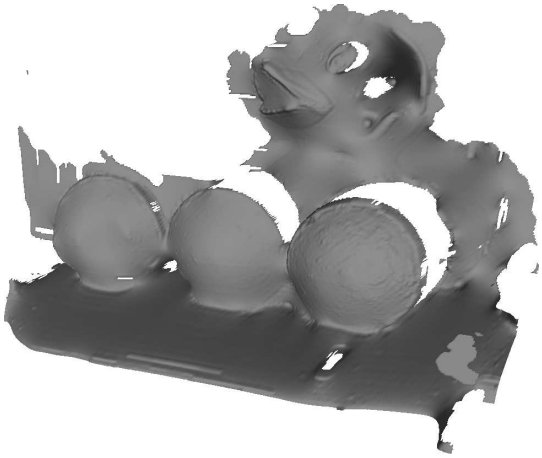
Although depth estimation performance is not very impressive, the approach allows extraction of very clean polarization images, see figure 5.19 on page 99. The reason for this is that polarization normals are not enough for this approach to disambiguate smooth areas. However, the estimated disparities will only be wrong in areas where this causes similarly colored areas to be mixed, resulting in still correct surface normals when performing polarization calculations. These



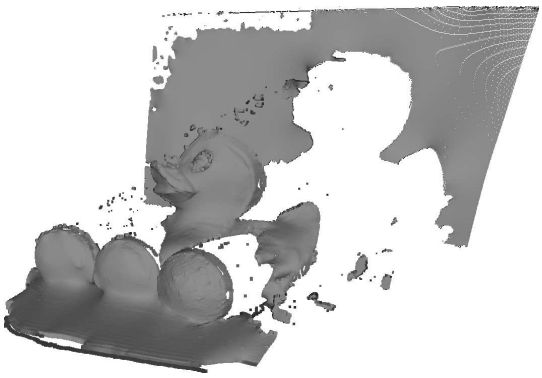
(a) center view image



(b) rgb visualization of estimated polarization information

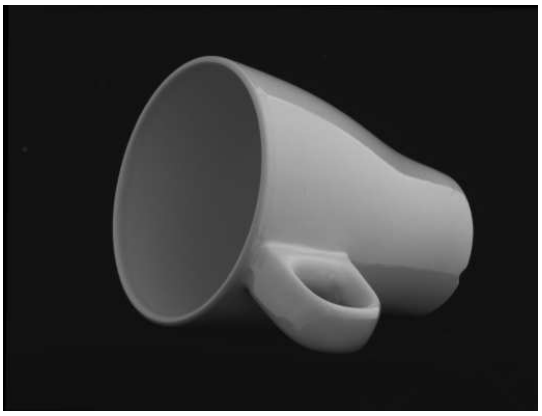


(c) regular processing

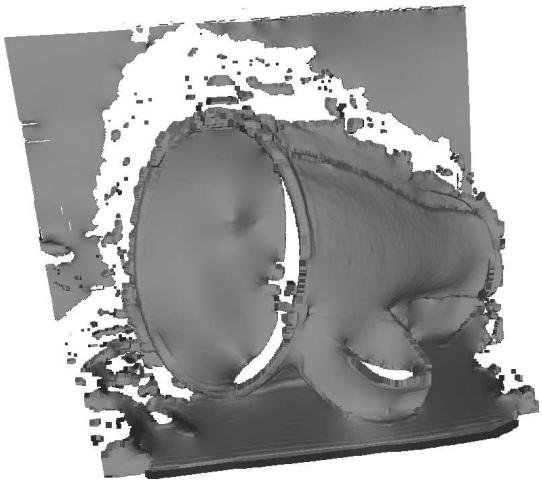


(d) processing with polarization normals

Figure 5.17: Examples of processed meshes with and without polarization imaging. The top row shows the center view of the data set (a) and the estimated polarization normals (b), visualized in RGB. The regular processing (c) shows artifacts in smooth glossy areas. The processing with polarization information (d) did successfully close gaps in the smooth areas and was also able to reconstruct most of the background, but shows strong oversmoothing.

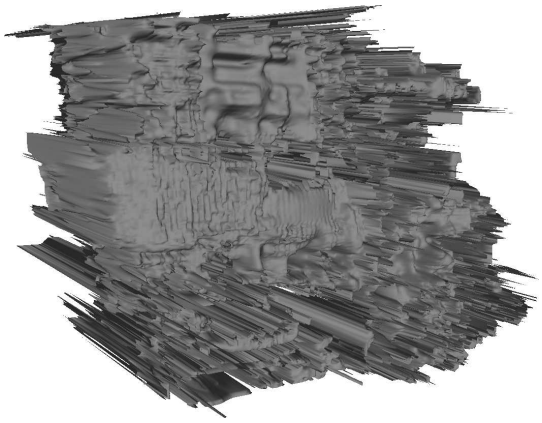


(a) center view image



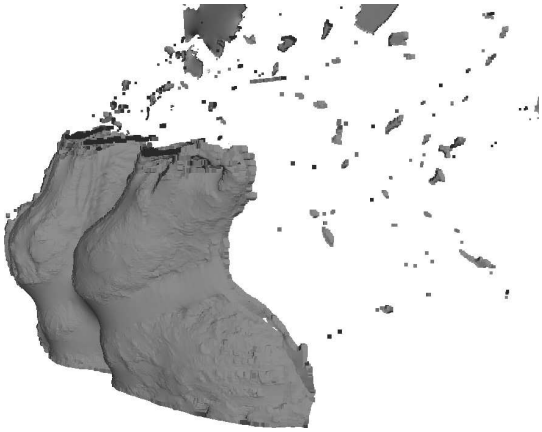
(b) processing with polarization normals

(c) center view image



(d) processing with polarization normals

(e) center view image



(f) processing with polarization normals

Figure 5.18: More examples of polarization based depth estimation.

(a) regular processing (b) polarization normals from merged center view (c) inline estimation with polarization normals

Figure 5.19: Comparison of polarization estimation on the duck data set. Using only the four polarization samples of the center view polarization image gives very noisy results (a). Creating a cleaner center view image by average samples along the EPI end then estimation the depth results in a cleaner estimate (b). The best result is achieved with the inline polarization estimate, as used internally during depth estimation (c). For this all polarization samples are used to estimate the polarization parameters, as described in section 5.7.0.4.

polarization normals are the result of all samples which are visible according to equation (5.11), which gives very clean estimates, see figure 5.19, which can be useful for more advanced methods building on polarization imaging.

5.7.0.9 Discussion

We have demonstrated the feasibility of integrating polarization processing into the depth estimation framework. While there are still some issues, this already improves depth estimation for several difficult cases. Also, the merged polarization images give very clean polarization cues, which might be useful for other analyses, like diffraction index estimation. Future work could improve on the results shown here, by implementing normal correction depending on view angle, or by explicit estimation of the refractive index from the redundant measurements under different viewing direction.

6 Conclusion

The following chapter summarizes the results of this thesis, followed by an outlook which points out possible future research directions and a final conclusion section.

6.1 Summary

To summarize, this thesis was concerned with two main topics: High accuracy camera calibration and improved light field depth estimation.

Calibration Towards the goal of high accuracy camera calibration, a novel fractal calibration target was proposed in chapter 3, which improves calibration point density, improves overall accuracy, and reduces center bias, as demonstrated with synthetic benchmark data. Based on this target a new approach for generic high accuracy ray based calibration was introduced in chapter 4, which allows high quality calibration from passive imperfectly manufactured calibration targets. This simplifies calibration of central and non-central cameras compared to the previous, active target based methods. The accuracy of the calibration approach was tested on real and synthetic data and verified with a stereo based evaluation.

Light Field Depth Estimation With respect to light field depth estimation, a new approach was introduced in chapter 5, which focuses on improved occlusion handling. This results in state-of-the-art accuracy with very good background/foreground separation and good regularization. The extensibility of this approach was demonstrated with surface normal estimation based on polarization normals, which improves results on difficult smooth and glossy surfaces. The performance was evaluated on a synthetic light field data set.

6.2 Outlook

Regarding the calibration target, although the quality is very high, there is a slight perspective bias, which could be corrected by taking perspective into account when fitting the local calibration points. Also, speed could be improved, for example by implementing a scale space approach.

For the ray based calibration, the main limitation is currently the parametrization of the rays, which limits the field of view to 180°. An angle/offset based parametrization should correct this, although for it to be useful also different undistortion projections should be implemented.

Another issue for the practical usability is fast re-calibration. In the field, cameras from a fixed setup might be moved, or extrinsics might change due to thermal expansion. A simple non-target extrinsics re-calibration could be possible, based on features extracted from the recorded scenes, for example SIFT [54] or ORB [74], to re-estimate the extrinsic parameters, while keeping the intrinsics constant.

With respect to the light field depth estimation, a wide range of future research is possible. The results in this thesis show that focusing on occlusion handling is very useful for depth estimation. A future, deep learning based implementation of the approach could get rid of most hand-tuned parameters, and could be based on a recurrent neural network architecture. Regarding the extensibility, new measurements or output modes could be added beyond polarization imaging, for example estimation of reflectance properties, or explicit optimization of material properties, with or without polarization imaging. For the special case of polarization imaging, it would be useful to actually estimate the material constants given either hand-labeled ground truth depth, or by jointly estimating the material properties, depth and normals given the redundant views from the light field imaging.

6.3 Conclusion

In conclusion, this thesis has advanced the state-of-the-art in camera calibration and light field depth estimation.

For camera calibration a highly accurate, yet easy to use calibration method was developed, which makes high accuracy camera calibration more accessible for researchers. This will benefit future research efforts in areas where high accuracy is paramount, like for example light field imaging.

For light field depth estimation a new approach with state-of-the-art performance was presented which allows highly accurate depth estimation, especially with respect to small structures and complex occlusions. This can benefit all applications which make use of 3D information, like visual effects or robotics applications. A range of future research directions can be based on the proposed depth estimation framework, and two possibilities were already explored based on light field BRDF estimation as well as light field polarization imaging. Both of these benefit smooth specular surfaces which are otherwise difficult to estimate with passive imaging methods.

List of Symbols

B center bias - measures the relative accuracy difference of a calibration between center and corner of the imaging area. 53

C camera matrix (3×3), respectively intrinsics function ($\mathbb{R}^3 \to \mathbb{R}^2$). 23, 26, 28, 30, 31, 60, 77, 78

D Center view disparity map. A 2D map of disparities where each disparity sample coincides with a pixel in the center view image of the light field. 83, 86, 88, 89, 124

D valid processing samples map. 103

E cost function (error function/objective function), denotes an error term minimized to achieve calibration or depth estimation. 64–69, 83, 88, 91, 123

G Gaussian distribution with standard deviation σ, to locally weight samples in the proxy fit. 63

G gradient map. 103

K radial distortion function ($\mathbb{R} \to \mathbb{R}$). 28

L a generic image space distortion function ($\mathbb{R}^2 \to \mathbb{R}^2$). 28–30

L the plenoptic function states the radiance given some parametrization of space and angle. 77

R rotation matrix (3×3), which rotates from the world (or calibration target) coordinate system into the camera (or view) coordinate system. For most purposes actually implemented using angle-axis vector representation. Full transform from world to camera coordinates for a world point w is $Rw + T$. 26, 28, 46, 59, 64, 65, 77, 78

T translation vector, which translates from the *R-rotated* world (or calibration target) coordinate system into the camera (or view) coordinate system. Full transform from world to camera coordinates for a world point p is $Rp + T$. 26, 28, 46, 59, 64, 65, 77–79

U arbitrary (*e.g.* lookup-table based) image space distortion function ($\mathbb{R}^2 \to \mathbb{R}^2$). 30

W valid disparities map. 103

Δ Color difference of a sample in the modified bilateral filter. 88

Γ EPI line - a 2D line on an epi, which is defined by the center view x coordinate and the slope defined by a disparity value. 85–87, 114, 124

Ω Modified bilateral filter, used in the smoothness term of the light field (LF) depth estimation. 87, 89

Φ EPI line intersectino, the point resulting from the intersection of two EPI lines Γ. 86, 87, 124

Ψ Perspective warp in image space ($\mathbb{R}^2 \to \mathbb{R}^2$, eight parameters). 63

Θ Proxy fit function. Maps from image to target coordinates ($\mathbb{R}^3 \to \mathbb{R}^2$). 63, 64

Υ 2D polynomial for proxy fit. 63, 115

α Weight of color differences in the modified bilateral filter. 88, 93

β Weight of disparity differences in the modified bilateral filter. 88, 93

δ Disparity difference of a sample in the modified bilateral filter. 88

ϵ Bilateral filter damping. 88, 93

η weight of the translation error in the multi-view constrain. 67, 68

$\hat{a}$ a set of calibration parameters (intrinsics and extrinsics) that minimizes an error measure with respect to a set of calibration points / images. 34

$\hat{p}$ point in normalized camera coordinates. 24–26, 28–30

$\hat{x}$ image point in homogeneous coordinates. 23, 24, 26, 28, 77, 78

κ weight of the rotation error in the multi-view constrain. 67, 68

λ Calculated weight of a sample in the modified bilateral filter. 87, 88

μ Mean of all unoccluded samples along an EPI line Γ. 87

ν Mesh point weight, from bilinear interpolation in constant target space. 65, 66

ω amplitude of polarization response function. 99, 100, 102

ψ offset of polarization response function. 99, 100, 102

ρ regularization weight in the depth estimation. 83, 93

σ standard deviation of Gaussian distribution used for proxy sample weights. 63, 113

τ Max range of random refinement. 92, 93

θ occlusion threshold. 83, 86–89, 92, 93

$\tilde{p}$ distored 3D point in distorted camera coordinates. 28, 60

υ Coefficient of the 2D polynomial Υ. 63

φ_{pol} polarization filter azimuth angle. 99

φ Surface normal azimuth angle. 99, 100, 102, 104

ϱ degree of polarization. 100, 103

ϑ Surface normal zenith angle. 100, 102

ξ data error. 83, 87

ζ smoothness error. 83, 87

a a set of calibration parameters (intrinsics and extrinsics). 34

b baseline - distance between two viewpoint. 79

c camera index. 60

c camera center (2D). 23, 25, 26, 28, 59, 60, 78, 79

d Direction vector for a ray of the ray based model. In principle 3D, but we assume $d_z = 1$ for simplicity. 59, 64, 65

d disparity - the distance (in pixels) of the apparent motion of object points in image space, caused by the viewpoint shift. 79, 80, 83, 85–89, 91, 92, 124

f focal lenght (2D, in pixel). 23, 25, 26, 28, 60, 66, 78, 79

g 3D target mesh support point. 65, 66

h converts from homogeneous to an euclidean coordinates. 24, 25, 28, 79

i image point and/or index. Image points are also directly used to index *e.g.* camera rays, which are defined via the corresponding image coordinate. 34, 46, 47, 116

k distortion parameter. 28

l A ray of the generic ray based camera model. Rays map 3D locations defined by the ray onto a 2D pixel coordinate, hence rays can be index by pixel coordinates i. 58–60, 62, 122

m distortion model. 34, 46, 47

n refractive index. 100, 104

o Origin of a ray of the ray based model. In principle 3D, but we assume $o_z = 0$ for simplicity. 58–60, 64, 65, 67, 122

p 3D point in local camera coordinates. 23, 25, 26, 31

r radius metric, calculates a radial distance depending metric from normalized camera coordinates (*e.g.* $\sqrt{(x^2 + y^2)}$ but can also be some other mapping). 28, 29

t Target point, in the target coordinate system defined by the location of the calibration target in the scene. In principle a 3D coordinate, but for the commonly used planar targets the z component is always zero. 34, 46, 47, 58, 64–66, 122

v view index. 63, 64

w 3D point in world (scene/target) coordinates. 26, 28, 77–79

x image point. 31, 79

Glossary

baseline baseline - distance between two viewpoint. 79

BRDF bidirectional reflectance distribution function. It describes how light is distributed when reflected at a surface. 12, 75, 85, 111, 124

calibration proxy Intermediate representation consisting of a multidimensional regular grid of image points and associated fitted target points. The calibration proxy maps from image points to target points, where the image points are located at fixed positions to enable ray fitting, and the target points are calculated to a very high accuracy, as they are the results of fitting a 2D function to thousands of image point/target point pairs.. 35, 63

camera resectioning geometric camera calibration. 21

center bias a measure of how biased a calibration is towards a high quality center of the imaging area, at the cost of inaccurate borders. 50, 52, 53, 55

disparity disparity – the distance (in pixels) of the apparent motion of object points in image space, caused by the viewpoint shift. 80, 83, 85

EPI epipolar plane image. 15, 80, 114

general camera model Very generic ray based camera model. A camera and lens is modeled as a black box, where the model consists of the mapping from 3D rays to 2D pixel coordinates. Can model non-central, omnidirectional, asymmetric and uneven cameras.. 23

GPE ground truth pixel error. 46, 47, 49–51, 53, 55, 70, 120, 121

homogeneous coordinates projective coordinates, useful in projective calculations, for example in camera projection. 24

LF light field. 114

Lidar Light detection and Ranging, an active method to measure the depth using laser based time-of-flight measurements.. 75

pinhole camera the ideal pinhole camera describes the perfect, central projection that an a mathematically perfect (and physically impossible) pinhole camera would possess, free of any geometric distortion. 23

plenoptic function the higher dimensional function which describes all light that travels within a scene, or relative to some object. 77–79, 113, 123

PMBP PatchMatch Belief Propagation. 11

radiance radiant flux emitted, reflected, transmitted or received by a given surface/area, per unit solid angle. 79, 113, 123

RANSAC RAndom SAmple Consensus, identifies outliers using an iterated sequence of: random sampling, model estimation, inlier detection [31].. 35

RMS root mean square error. 39, 44, 46, 53, 71–73

subaperture view a 2D perspective projection with a finite aperture from the full light field, i.e. a single viewpoint from a light field array. 77

test fields large scale calibration target, term from the photogrammetry community. 33

TPE true pixel error. 44, 46, 47, 53, 56, 122

List of Figures

List of Tables

Bibliography

[1] Edward H. Adelson and James R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, pages 3–20. MIT Press, 1991.

[2] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. http://ceres-solver.org.

[3] Andrea Albarelli, Emanuele Rodolà, and Andrea Torsello. Robust camera calibration using inaccurate targets. In Frédéric Labrosse, Reyer Zwiggelaar, Yonghuai Liu, and Bernie Tiddeman, editors, *BMVC*, pages 1–10. British Machine Vision Association, 2010. ISBN 1-901725-40-5. URL http://dblp.uni-trier.de/db/conf/bmvc/bmvc2010.html#AlbarelliRT10;http://dx.doi.org/10.5244/C.24.16;http://www.bibsonomy.org/bibtex/2561cd11236da10b876680b1495ebf1bb/dblp.

[4] Bradley Atcheson, Felix Heide, and Wolfgang Heidrich. Caltag: High precision fiducial markers for camera calibration. In Reinhard Koch, Andreas Kolb, and Christof Rezk-Salama, editors, *Vision, Modeling, and Visualization (2010)*. The Eurographics Association, 2010. ISBN 978-3-905673-79-1. doi: 10.2312/PE/VMV/VMV10/041-048.

[5] Gary A Atkinson and Edwin R Hancock. Shape from diffuse polarisation. In *BMVC*, pages 1–10, 2004.

[6] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24:1–24:11, July 2009. ISSN 0730-0301. doi: 10.1145/1531326.1531330.

[7] Joao P Barreto, Rahul Swaminathan, and Jose Roquette. Non parametric distortion correction in endoscopic medical images. In *3DTV Conference, 2007*, pages 1–4. IEEE, 2007.

[8] Filippo Bergamasco, Andrea Albarelli, Emanuele Rodolà, and Andrea Torsello. Can a fully unconstrained imaging model be applied effectively to central cameras? In *CVPR*, pages 1391–1398. IEEE Computer Society, 2013. ISBN 978-0-7695-4989-7. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2013.html#BergamascoART13;http://doi.

ieeecomputersociety.org/10.1109/CVPR.2013.183;http://www.bibsonomy.org/bibtex/258f4d060c797dcb493d35edf614220df/dblp.

[9] Filippo Bergamasco, Andrea Albarelli, Luca Cosmo, Emanuele Rodola, and Andrea Torsello. An accurate and robust artificial marker based on cyclic codes. 2016. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7384749.

[10] Filippo Bergamasco, Luca Cosmo, Andrea Gasparetto, Andrea Albarelli, and Andrea Torsello. Parameter-free lens distortion calibration of central cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3847–3855, 2017.

[11] Blender Online Community. Blender - a 3d modelling and rendering package, 2016. URL http://www.blender.org.

[12] Jean-Yves Bouguet. Camera calibration tool-box for matlab. *http://www.vision. caltech. edu/bouguetj/calib_doc/*, 2002.

[13] G. Bradski. The opencv library. *Dr. Dobb's Journal of Software Tools*, 2000.

[14] D.C. Brown. The simultaneous determination of the orientation and lens distortion of a photogrammetric camera. Technical report, AFMTC-TN-56-20, ASTIA Document, 1956.

[15] D.C. Brown. Decentering distortion of lenses. *Photogrammetric Engineering*, 32:444–462, 1966.

[16] Duane C Brown. Close-range camera calibration. *Photogramm. Eng*, 37(8): 855–866, 1971.

[17] Federico Camposeco, Torsten Sattler, and Marc Pollefeys. Non-parametric structure-based calibration of radially symmetric cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2192–2200, 2015.

[18] David Claus and Andrew W Fitzgibbon. A rational function lens distortion model for general cameras. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 213–219. IEEE, 2005.

[19] Zhaopeng Cui, Jinwei Gu, Boxin Shi, Ping Tan, and Jan Kautz. Polarimetric multi-view stereo. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] V.F. da Camara Neto, D. Balbino de Mesquita, R.F. Garcia, and M.F.M. Campos. On the design and evaluation of a precise scalable fiducial marker framework. In *Graphics, Patterns and Images (SIBGRAPI), 2010 23rd SIBGRAPI Conference on*, pages 216–223, Aug 2010. doi: 10.1109/SIBGRAPI.2010.37.

[21] Shreyansh Daftry, Michael Maurer, Andreas Wendel, and Horst Bischof. Flexible and user-centric camera calibration using planar fiducial markers. In *BMVC*, 2013.

[22] Ankur Datta, Jun-Sik Kim, and Takeo Kanade. Accurate camera calibration using iterative refinement of control points. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1201–1208. IEEE, 2009. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5457474;https://www.ri.cmu.edu/pub_files/2009/10/VS.ICCV.2009.pdf.

[23] M Diebold, O Blum, M Gutsche, S Wanner, C Garbe, H Baker, and B Jähne. Light-field camera design for high-accuracy depth estimation. In *SPIE Optical Metrology*, pages 952803–952803. International Society for Optics and Photonics, 2015.

[24] Damien Douxchamps and Kunihiro Chihara. High-accuracy and robust localization of large control markers for geometric camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):376–383, 2009. URL http://dblp.uni-trier.de/db/journals/pami/pami31.html#DouxchampsC09;http://doi.ieeecomputersociety.org/10.1109/TPAMI.2008.214;http://www.bibsonomy.org/bibtex/2af89069404722635a687de17d9a1257b/dblp.

[25] Jean Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*, pages 85–100. Springer, 1977.

[26] Wolfgang Faig. Calibration of close-range photogrammetric systems: Mathematical formulation. *Photogrammetric engineering and remote sensing*, 41 (12), 1975.

[27] Mohammed E Fathy, Ashraf S Hussein, and Mohammed F Tolba. Fundamental matrix estimation: A study of error criteria. *Pattern Recognition Letters*, 32(2):383–391, 2011.

[28] M. Fiala. Designing highly reliable fiducial markers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(7):1317–1324, July 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2009.146.

[29] Mark Fiala. Artag, a fiducial marker system using digital techniques. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 590–596. IEEE, 2005.

[30] Mark Fiala and Chang Shu. Self-identifying patterns for plane-based camera calibration. *Mach. Vis. Appl.*, 19(4):209–216, 2008. URL http://dblp.uni-trier.de/db/journals/mva/mva19.html#FialaS08;http://dx.doi.org/10.1007/s00138-007-0093-z;http://www.bibsonomy.org/bibtex/2d8e46fc42a36c863b178e0db28d8ca85/dblp.

[31] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[32] Clive S Fraser. Digital camera self-calibration. *ISPRS Journal of Photogrammetry and Remote sensing*, 52(4):149–159, 1997.

[33] CS Fraser and MR Shortis. Variation of distortion within the photographic field. *Photogrammetric Engineering and Remote Sensing*, 58(6):851–855, 1992.

[34] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recogn.*, 47(6):2280–2292, June 2014. ISSN 0031-3203. doi: 10.1016/j.patcog.2014.01.005.

[35] Andreas Geiger, Frank Moosmann, Ömer Car, and Bernhard Schuster. Automatic camera and range sensor calibration using a single shot. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3936–3943. IEEE, 2012.

[36] J.W. Goodman. *Introduction to Fourier Optics*. MaGraw-Hill, 2nd edition, 1996.

[37] Ardeshir Goshtasby. Correction of image deformation from lens distortion using bezier patches. *Computer Vision, Graphics, and Image Processing*, 47(3):385–394, 1989.

[38] Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*, volume 105. Siam, 2008.

[39] Rafael Grompone von Gioi, Pascal Monasse, Jean-Michel Morel, and Zhongwei Tang. Towards high-precision lens distortion correction. In *ICIP*, pages 4237–4240. IEEE, 2010. ISBN 978-1-4244-7994-8. URL http://dblp.uni-trier.de/db/conf/icip/icip2010.html#GioiMMT10;http://dx.doi.org/10.1109/ICIP.2010.5651928;http://www.bibsonomy.org/bibtex/20b74d629c38ad7ceb2e9ad892e8887b4/dblp.

[40] Michael D. Grossberg and Shree K. Nayar. A general imaging model and a method for finding its parameters. In *ICCV*, pages 108–115, 2001. URL http://dblp.uni-trier.de/db/conf/iccv/iccv2001-2.html#GrossbergN01;http://doi.ieeecomputersociety.org/10.1109/ICCV.2001.937611;http://www.bibsonomy.org/bibtex/244bb053a99cb57432389aa93f4677f97/dblp.

[41] Marcel Gutsche. *Light Fields Reconstructing Geometry and Reflectance Properties*. PhD thesis, 2018.

[42] Marcel Gutsche, Hendrik Schilling, Maximilian Diebold, and Christoph Garbe. Surface normal reconstruction from specular information in light field data. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1735–1742. IEEE, 2017.

[43] Matthew Harker and Paul O'Leary. First order geometric distance (the myth of sampsonus). In *BMVC*, pages 87–96, 2006.

[44] Richard Hartley and Sing Bing Kang. Parameter-free radial distortion correction with center of distortion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1309–1321, 2007.

[45] Janne Heikkila and Olli Silven. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106–1112. IEEE, 1997.

[46] Katrin Honauer and Ole Johannsen. 4d light field dataset. http://hci-lightfield.iwr.uni-heidelberg.de/, 2016.

[47] Katrin Honauer, Ole Johannsen, Daniel Kondermann, and Bastian Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. In *Asian Conference on Computer Vision*. Springer, 2016. *http://lightfield-analysis.net*.

[48] Ole Johannsen, Katrin Honauer, Bastian Goldluecke, Anna Alperovich, Federica Battisti, Yunsu Bok, Michele Brizzi, Marco Carli, Gyeongmin Choe, Maximilian Diebold, Marcel Gutsche, Hae-Gon Jeon, In So Kweon, Alessandro Neri, Jaesik Park, Jinsun Park, Hendrik Schilling, Hao Sheng, Lipeng Si, Michael Strecke, Antonin Sulc, Yu-Wing Tai, Qing Wang, Ting-Chun Wang, Sven Wanner, Zhang Xiong, Jingyi Yu, Shuo Zhang, and Hao Zhu. A taxonomy and evaluation of dense light field depth estimation algorithms. In *Conference on Computer Vision and Pattern Recognition - LF4CV Workshop*, 2017.

[49] Achuta Kadambi, Vage Taamazyan, Boxin Shi, and Ramesh Raskar. Polarized 3d: High-quality depth sensing with polarization cues. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3370–3378, 2015.

[50] Juho Kannala and Sami S Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, 28(8):1335–1340, 2006.

[51] Gershon Kedem. Automatic differentiation of computer programs. Technical report, WISCONSIN UNIV MADISON MATHEMATICS RESEARCH CENTER, 1976.

[52] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. *Computer Vision—ECCV 2002*, pages 8–40, 2002.

[53] Haiting Lin, Can Chen, Sing Bing Kang, and Jingyi Yu. Depth recovery from light field using focal stack symmetry. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3451–3459, 2015.

[54] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[55] Luca Lucchese and Sanjit K. Mitra. Using saddle points for subpixel feature detection in camera calibration targets. In *APCCAS (2)*, pages 191–195. IEEE, 2002. ISBN 0-7803-7690-0. URL http://dblp.uni-trier.de/db/conf/apccas/apccas2002-2.html#LuccheseM02.

[56] Thomas Luhmann, Heidi Hastedt, and Werner Tecklenburg. Modelling of chromatic aberration for high precision photogrammetry. In *Commission V Symp. on Image Engineering and Vision Metrology, Proc. ISPRS*, volume 36, pages 173–178, 2006.

[57] Lili Ma, YangQuan Chen, and Kevin L Moore. Rational radial distortion models of camera lenses with analytical solution for distortion correction. *International Journal of Information Acquisition*, 1(02):135–147, 2004.

[58] Arthur A Magill. Variation in distortion with magnification. *JOSA*, 45(3): 148–149, 1955.

[59] John Mallon and Paul F Whelan. Which pattern? biasing aspects of planar calibration patterns and detection methods. *Pattern recognition letters*, 28(8):921–930, 2007. URL http://www.sciencedirect.com/science/article/pii/S0167865506003114;http://doras.dcu.ie/18667/1/whelan_2007_64.pdf.

[60] HA Martins, JR Birk, and R Bo Kelley. Camera models based on data from two calibration planes. *Computer Graphics and Image Processing*, 17 (2):173–180, 1981.

[61] Branislav Micusik and Tomas Pajdla. Autocalibration & 3d reconstruction with non-central catadioptric cameras. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2004.

[62] Pedro Miraldo and Helder Araújo. Calibration of smooth camera models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(9): 2091–2103, 2013. URL http://dblp.uni-trier.de/db/journals/pami/pami35.html#MiraldoA13;http://doi.ieeecomputersociety.org/10.1109/TPAMI.2012.258;http://www.bibsonomy.org/bibtex/27f664547d458b75d34cda23f37277fea/dblp.

[63] Pedro Miraldo, Helder Araújo, and Joao Queiro. Point-based calibration using a parametric representation of the general imaging model. In

Dimitris N. Metaxas, Long Quan, Alberto Sanfeliu, and Luc J. Van Gool, editors, *ICCV*, pages 2304–2311. IEEE Computer Society, 2011. ISBN 978-1-4577-1101-5. URL http://dblp.uni-trier.de/db/conf/iccv/iccv2011.html#MiraldoAQ11;http://doi.ieeecomputersociety.org/10.1109/ICCV.2011.6126511;http://www.bibsonomy.org/bibtex/2239591295f5d3dc29d694d0fd90362f0/dblp.

[64] Alessandro Neri, Marco Carli, and Federica Battisti. A multi-resolution approach to depth field estimation in dense image arrays. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 3358–3362. IEEE, 2015.

[65] Mai Nishimura, Shohei Nobuhara, Takashi Matsuyama, Shinya Shimizu, and Kensaku Fujii. A linear generalized camera calibration from three intersecting reference planes. In *ICCV*, pages 2354–2362. IEEE Computer Society, 2015. ISBN 978-1-4673-8391-2. URL http://dblp.uni-trier.de/db/conf/iccv/iccv2015.html#NishimuraNMSF15;http://doi.ieeecomputersociety.org/10.1109/ICCV.2015.271;http://www.bibsonomy.org/bibtex/2e058ba5d5dc4dd503cef381ad0cbd3f3/dblp.

[66] E. Olson. AprilTag: a robust and flexible visual fiducial system. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3400–3407, May 2011. doi: 10.1109/ICRA.2011.5979561.

[67] Simon Placht, Peter Fürsattel, Etienne Assoumou Mengue, Hannes G. Hofmann, Christian Schaller, Michael Balda, and Elli Angelopoulou. Rochade: Robust checkerboard advanced detection for camera calibration. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *ECCV (4)*, volume 8692 of *Lecture Notes in Computer Science*, pages 766–779. Springer, 2014. ISBN 978-3-319-10592-5. URL http://dblp.uni-trier.de/db/conf/eccv/eccv2014-4.html#PlachtFMHSBA14;http://dx.doi.org/10.1007/978-3-319-10593-2_50;http://www.bibsonomy.org/bibtex/2bc9d1d651193d3a7c360bc87d2dfba1f/dblp.

[68] Louis B Rall and George F Corliss. An introduction to automatic differentiation. *Computational Differentiation: Techniques, Applications, and Tools*, 89, 1996.

[69] Srikumar Ramalingam, Peter F. Sturm, and Suresh K. Lodha. Towards complete generic camera calibration. In *CVPR (1)*, pages 1093–1098. IEEE Computer Society, 2005. ISBN 0-7695-2372-2. URL http://dblp.uni-trier.

de/db/conf/cvpr/cvpr2005-1.html#RamalingamSL05;http://doi.ieeecomputersociety.org/10.1109/CVPR.2005.347;http://www.bibsonomy.org/bibtex/2c80d3637f0ed9f2427e86c7f6e1e1675/dblp.

[70] Pradeep Ranganathan and Edwin Olson. Locally-weighted homographies for calibration of imaging systems. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 404–409. IEEE, 2014.

[71] Fabio Remondino and Clive Fraser. Digital camera calibration methods: considerations and comparisons. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5):266–272, 2006.

[72] Carlos Ricolfe-Viala and Antonio-Jose Sanchez-Salmeron. Lens distortion models evaluation. *Applied optics*, 49(30):5914–5928, 2010.

[73] Carsten Rother and Andrew Fitzgibbon. Pmbp: Patchmatch belief propagation for correspondence field estimation. In *BMVC - Best Industrial Impact Prize award*, January 2012. URL https://www.microsoft.com/en-us/research/publication/pmbp-patchmatch-belief-propagation-for-correspondence-field-estimation/.

[74] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[75] J. Sattar, E. Bourque, P. Giguere, and G. Dudek. Fourier tags: Smoothly degradable fiducial markers for use in human-robot interaction. In *Computer and Robot Vision, 2007. CRV '07. Fourth Canadian Conference on*, pages 165–174, May 2007. doi: 10.1109/CRV.2007.34.

[76] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5695–5701. IEEE, 2006.

[77] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014.

[78] Hendrik Schilling. Hdmarker - high density fractal marker for camera calibration. https://github.com/hendrikschilling/hdmarker, 2016.

[79] Hendrik Schilling, Maximilian Diebold, Marcel Gutsche, and Bernd Jähne. On the design of a fractal calibration pattern for improved camera calibration. *tm-Technisches Messen*, 2017.

[80] Hendrik Schilling, Maximilian Diebold, Bernd Jähne, and Carsten Rother. Trust your model: Light field depth estimation with inline occlusion handling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[81] Sönke Schmid, Xiaoyi Jiang, and Klaus P. Schäfers. High-precision lens distortion correction using smoothed thin plate splines. In Richard C. Wilson, Edwin R. Hancock, Adrian G. Bors, and William A. P. Smith, editors, *CAIP (2)*, volume 8048 of *Lecture Notes in Computer Science*, pages 432–439. Springer, 2013. ISBN 978-3-642-40245-6. URL http://dblp.uni-trier.de/db/conf/caip/caip2013-2.html#SchmidJS13;http://dx.doi.org/10.1007/978-3-642-40246-3_54;http://www.bibsonomy.org/bibtex/2cb9ba1ca6c83d29e3eea64194675dc68/dblp.

[82] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

[83] Shishir Shah and JK Aggarwal. Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation. *Pattern Recognition*, 29(11):1775–1788, 1996.

[84] Edward Shen and Richard Hornsey. Multi-camera network calibration with a non-planar target. *IEEE Sensors Journal*, 11(10):2356–2364, 2011.

[85] Hao Sheng, Pan Zhao, Shuo Zhang, Jun Zhang, and Da Yang. Occlusion-aware depth estimation for light field using multi-orientation epis. *Pattern Recognition*, 2017.

[86] Changha Shin, Hae-Gon Jeon, Youngjin Yoon, In So Kweon, and Seon Joo Kim. Epinet: A fully-convolutional neural network using epipolar geometry for depth from light field images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4748–4757, 2018.

[87] Hendrik Siedelmann. Recording, compression and representation of dense light fields. Diplomarbeit, Universität Stuttgart, Fakultät Informatik, Elektrotechnik und Informationstechnik, Germany, Januar 2015. URL http://www2.informatik.uni-stuttgart.de/cgi-bin/NCSTRL/NCSTRL_view.pl?id=DIP-3677&engl=0.

[88] Hendrik Siedelmann, Maximilian Diebold, Marcel Gutsche, Hamza Aziz-Ahmad, and Bernd Jähne. A fractal calibration pattern for improved camera calibration. To be published, 2016.

[89] William AP Smith, Ravi Ramamoorthi, and Silvia Tozza. Linear depth estimation from an uncalibrated, monocular polarisation image. In *European Conference on Computer Vision*, pages 109–125. Springer, 2016.

[90] Michael Strecke, Anna Alperovich, and Bastian Goldluecke. Accurate depth and normal maps from occlusion-aware focal stack symmetry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017.

[91] Klaus H. Strobl and Gerd Hirzinger. More accurate pinhole camera calibration with imperfect planar target. In *ICCV Workshops*, pages 1068–1075. IEEE, 2011. ISBN 978-1-4673-0062-9. URL http://dblp.uni-trier.de/db/conf/iccvw/iccvw2011.html#StroblH11;http://dx.doi.org/10.1109/ICCVW.2011.6130369;http://www.bibsonomy.org/bibtex/2a63462c8878fc9dae7fe6e648ccf5d31/dblp.

[92] Peter Sturm, Srikumar Ramalingam, Jean-Philippe Tardif, Simone Gasparini, and João Pedro Barreto. Camera models and fundamental concepts used in geometric computer vision by. 2011.

[93] Peter F. Sturm and Srikumar Ramalingam. A generic concept for camera calibration. In Tomás Pajdla and Jiri Matas, editors, *ECCV (2)*, volume 3022 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2004. ISBN 3-540-21983-8. URL http://dblp.uni-trier.de/db/conf/eccv/eccv2004-2.html#SturmR04;http://dx.doi.org/10.1007/978-3-540-24671-8_1;http://www.bibsonomy.org/bibtex/2b2aa98f9db0c5af398d2e6215d4f8df5/dblp.

[94] Rahul Swaminathan, Michael D Grossberg, and Shree K Nayar. Non-single viewpoint catadioptric cameras: Geometry and analysis. *International Journal of Computer Vision*, 66(3):211–229, 2006.

[95] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A comparative study of energy minimization methods for markov random fields. *Computer Vision–ECCV 2006*, pages 16–29, 2006.

[96] Vaibhav Vaish and Andrew Adams. The (new) stanford light field archive. *Computer Graphics Laboratory, Stanford University*, 2008.

[97] Sven Wanner. *Orientation Analysis in 4D Light Fields*. Dissertation, IWR, Fakultät für Physik und Astronomie, Univ. Heidelberg, 2014. URL http://www.ub.uni-heidelberg.de/archiv/16439.

[98] Sven Wanner and Bastian Goldluecke. Globally consistent depth labeling of 4d light fields. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 41–48. IEEE, 2012.

[99] Sven Wanner and Bastian Goldluecke. Reconstructing reflective and transparent surfaces from epipolar plane images. In *German Conference on Pattern Recognition*, pages 1–10. Springer, 2013.

[100] Sven Wanner and Bastian Goldluecke. Variational light field analysis for disparity estimation and super-resolution. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):606–619, 2014.

[101] W Williem and In Kyu Park. Robust light field depth estimation for noisy scene with occlusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4396–4404, 2016.

[102] Williem Williem, In Kyu Park, and Kyoung Mu Lee. Robust light field depth estimation using occlusion-noise aware data costs. *(pre-print) IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[103] Tomohiro Yamazaki, Yasushi Maruyama, Yusuke Uesaka, Motoaki Nakamura, Yoshihisa Matoba, Takashi Terada, Kenta Komori, Yoshiyuki Ohba, Shinichi Arakawa, Yasutaka Hirasawa, et al. Four-directional pixel-wise polarization cmos image sensor using air-gap wire grid on 2.5-μm back-illuminated pixels. In *Electron Devices Meeting (IEDM), 2016 IEEE International*, pages 8–7. IEEE, 2016.

[104] Shuo Zhang, Hao Sheng, Chao Li, Jun Zhang, and Zhang Xiong. Robust depth estimation for light field via spinning parallelogram operator. *Computer Vision and Image Understanding*, 145:148–159,

2016. URL http://dblp.uni-trier.de/db/journals/cviu/cviu145.html#ZhangSLZX16;http://dx.doi.org/10.1016/j.cviu.2015.12.007.

[105] Xiang Zhang, Stephan Fronz, and Nassir Navab. Visual marker detection and decoding in ar systems: A comparative study. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, page 97. IEEE Computer Society, 2002.

[106] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334, Nov 2000. ISSN 0162-8828. doi: 10.1109/34.888718.