Underwater 3D Reconstruction Based on Physical Models for Refraction and Underwater Light Propagation

Dipl.-Inf. Anne Jordt

Dissertation
zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)
der Technischen Fakultät
der Christian-Albrechts-Universität zu Kiel
eingereicht im Jahr 2013

Kiel Computer Science Series (KCSS) 2014/2 v1.0 dated 2014-01-20

ISSN 2193-6781 (print version)
ISSN 2194-6639 (electronic version)

Electronic version, updates, errata available via https://www.informatik.uni-kiel.de/kcss

The author can be contacted via http://www.mip.informatik.uni-kiel.de

Published by the Department of Computer Science, Kiel University

Multimedia Information Processing

Please cite as:

▷ Anne Jordt. *Underwater 3D Reconstruction Based on Physical Models for Refraction and Underwater Light Propagation.* Number 2014/2 in Kiel Computer Science Series. Department of Computer Science, 2014. Dissertation, Faculty of Engineering, Kiel University.

```
@book{AJordt14,
  author    = {Anne Jordt},
  title     = {Underwater {3D} Reconstruction Based on Physical Models
               for Refraction and Underwater Light Propagation},
  publisher = {Department of Computer Science, CAU Kiel},
  year      = {2014},
  number    = {2014/2},
  series    = {Kiel Computer Science Series},
  note      = {Dissertation, Faculty of Engineering,
               Kiel University}
}
```

About this Series

The Kiel Computer Science Series (KCSS) covers dissertations, habilitation theses, lecture notes, textbooks, surveys, collections, handbooks, etc. written at the Department of Computer Science at Kiel University. It was initiated in 2011 to support authors in the dissemination of their work in electronic and printed form, without restricting their rights to their work. The series provides a unified appearance and aims at high-quality typography. The KCSS is an open access series; all series titles are electronically available free of charge at the department's website. In addition, authors are encouraged to make printed copies available at a reasonable price, typically with a print-on-demand service.

Please visit `http://www.informatik.uni-kiel.de/kcss` for more information, for instructions how to publish in the KCSS, and for access to all existing publications.

1. Gutachter: Prof. Dr.-Ing. Reinhard Koch
Christian-Albrechts-Universität zu Kiel

2. Gutachter: Prof. Dr.-Ing. Helmut Mayer
Bundeswehr Universtät München

Datum der mündlichen Prüfung: 12. November 2013

Zusammenfassung

In den vergangenen Jahren wurde das Aufnehmen von Unterwasserbildern immer beliebter. Die Gründe sind einerseits, dass sich handelsübliche Unterwasserkameras immer stärker verbreiten, andererseits wächst auch das Interesse am Ozeanboden – sowohl in der Wissenschaft und Forschung, als auch bei der Industrie. Bilder und Filme werden häufig nicht nur aufgenommen und angeschaut, auch das Interesse an Anwendungen aus dem Bereich des maschinellen Sehens ist gewachsen. Allerdings wird dabei oft außer Acht gelassen, dass das Wasser großen Einfluss auf die Bildentstehung hat. Zum einen wird Licht abgeschwächt und gestreut, während es sich im Wasser ausbreitet. Das ist ein von Wellenlängen abhängiger Effekt, der die starke grünliche oder bläuliche Färbung in Unterwasserbildern verursacht. Zum anderen brauchen Unterwasserkameras notgedrungen ein Gehäuse und betrachten daher die Szene durch ein entweder flaches oder gewölbtes Glas. Im Gehäuse befindet sich Luft, außerhalb ist Wasser. Das führt dazu, dass ein Lichtstrahl, der das Gehäuse erreicht, zweimal gebrochen wird; einmal am Übergang zwischen Wasser und Glas und ein zweites Mal am Übergang zwischen Glas und Luft, was die Geometrie der Bildentstehung beeinflusst. In klassischen Ansätzen zu Structure-from-Motion (SfM) wird üblicherweise das perspektivische Kameramodell verwendet, allerdings kann man leicht zeigen, dass es durch den Einfluss von Lichtbrechung in mehreren Medien (Luft, Glas und Wasser) ungültig wird.

Daher wird in dieser Arbeit gezeigt, wie der klassische Ansatz für SfM-Algorithmen angepasst werden kann, damit er Unterwasserkameragehäusen mit flachen Glasscheiben gerecht wird. Dazu wird ein vollständiges Verfahren vorgestellt, in dem die Lichtbrechung explizit modelliert wird, welches aus einem Kalibrierverfahren, Algorithmen für absolute und relative Poseschätzung, einer effizienten, nicht-linearen Fehlerfunktion für Bündelausgleich und einem Plane-Sweep-Algorithmus mit Lichtbrechung besteht. Außerdem kann im Falle von vorliegenden Kalibrierdaten ein

Modell für die Lichtausbreitung unter Wasser parametrisiert werden, das mit Hilfe dichter Tiefenkarten zur Korrektur der Texturfarben verwendet werden kann.

Vergleichende Experimente mit einem perspektivischen und mit dem vorgestellten Ansatz mit Lichtbrechung haben gezeigt, dass der perspektivische Ansatz tatsächlich einen systematischen Fehler aufweist, der von der Distanz zwischen Kamera und Glas und von einer möglichen Neigung zwischen Glas und Bildsensor abhängt. Der hier vorgeschlagene Ansatz weist keinen solchen Fehler auf, ist also in der Lage, genauere Rekonstruktionsergebnisse für Unterwasserbilder zu berechnen.

Abstract

In recent years, underwater imaging has gained a lot of popularity partly due to the availability of off-the-shelf consumer cameras, but also due to a growing interest in the ocean floor by science and industry. Apart from capturing single images or sequences, the application of methods from the area of computer vision has gained interest as well. However, water affects image formation in two major ways. First, while traveling through the water, light is attenuated and scattered, depending on the light's wavelength causing the typical strong green or blue hue in underwater images. Second, cameras used in underwater scenarios need to be confined in an underwater housing, viewing the scene through a flat or dome-shaped glass port. The inside of the housing is filled with air. Consequently, the light entering the housing needs to pass a water-glass interface, then a glass-air interface, thus is refracted twice, affecting underwater image formation geometrically. In classic Structure-from-Motion (SfM) approaches, the perspective camera model is usually assumed, however, it can be shown that it becomes invalid due to refraction in underwater scenarios.

Therefore, this thesis proposes an adaptation of the SfM algorithm to underwater image formation with flat port underwater housings, i. e., introduces a method where refraction at the underwater housing is modeled explicitly. This includes a calibration approach, algorithms for relative and absolute pose estimation, an efficient, non-linear error function that is utilized in bundle adjustment, and a refractive plane sweep algorithm. Finally, if calibration data for an underwater light propagation model exists, the dense depth maps can be used to correct texture colors.

Experiments with a perspective and the proposed refractive approach to 3D reconstruction revealed that the perspective approach does indeed suffer from a systematic model error depending on the distance between camera and glass and a possible tilt of the glass with respect to the image sensor. The proposed method shows no such systematic error and thus provides more accurate results for underwater image sequences.

Acknowledgements

I would like to seize this opportunity to thank a few people without whom I could not have completed the work on this thesis. First and most importantly, I would like to thank my mentor, Prof. Dr.-Ing. Reinhard Koch, the leader of the work group for Multimedia Information Processing at the Institute of Computer Science at Kiel University. He provided the necessary guidance in an excellent work environment that made work an enjoyable part of life.

The work group itself with current (Renate Stäcker, Torge Storm, Daniel Jung, Andreas Jordt, Robert Wulff, Markus Franke, Sandro Esquivel, Oliver Fleischmann, Dominik Wolters, and Johannes Brünger) and former members (Jan-Friso Evers Senne, Kevin Köser, Bogumil Bartczak, Kristine Bauer, Anatol Frick, Falko Kellner, Arne Petersen, Ingo Schiller, and Lilian Zhang) provided the possibility to discuss and to share the pressure of writing papers and the thesis itself.

In addition to the work group, the topic of underwater imaging offers interesting collaboration possibilities with other scientists from different areas of research in order to apply the developed methods. In this context, I would like to thank Tom Kwasnitschka from the Geomar Helmholtz Centre for Ocean Research, Florian Huber, Christian Howe, and Uli Kunz of the scientific diver group at Kiel University, and the Geomar ROV team for providing interesting underwater images along with the necessary explanations of what they show and how that came into being.

I would like to thank Prof. Dr. Helmut Mayer, Prof. Dr. Rudolf Berghammer, and Prof. Dr. Steffen Börm for being in my thesis committee.

Finally, I would like to thank my parents for providing the necessary education that allowed me to take up my work at the university and especially my husband Andreas for putting up with me during the more stressful times and for proofreading my thesis.

Contents

List of Figures

List of Tables

List of Abbreviations

AbyS	Analysis-by-Synthesis
AUV	Autonomous Underwater Vehicle
BA	Bundle Adjustment
BRDF	Bidirectional Reflectance Distribution Function
CCD	Charged Coupled Device
CMA-ES	Covariance Matrix Adaptation - Evolution Strategy
DLT	Direct Linear Transform
DoF	Degrees of Freedom
FRC	Flat Refractive Constraint
GEC	Generalized Epipolar Constraint
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
ML	Maximum Likelihood
NCC	Normalized Cross Correlation
nSVP	non-Single-View-Point
PMVS	Patch-based Multi-view Stereo
POR	Plane of Refraction
POSIT	Pose from Orthography and Scaling with ITerations
RANSAC	Random Sampling Consensus

RE	Reprojection Error
ROV	Remotely Operated Vehicle
SAD	Sum of Absolute Differences
SfM	Structure-from-Motion
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SVD	Singular Value Decomposition
VSF	Volume Scattering Function

Symbols and Notations

$\mathbb{P}^n$	$n+1$ -dimensional projective space, hosts $\mathbb{R}^n$
a	scalar
$\mathbf{A}$	matrix
$\mathbf{x} = (x, y, 1)^\mathrm{T}$	homogeneous 2D vector
$\mathbf{X} = (X, Y, Z, 1)^\mathrm{T}$	homogeneous 3D vector
$\mathrm{x} = (x, y)^\mathrm{T}$	euclidean 2D vector
$\mathrm{X} = (X, Y, Z)^\mathrm{T}$	euclidean 3D vector
$\tilde{\mathrm{X}} = (\tilde{X}, \tilde{Y}, \tilde{Z})^\mathrm{T}$	ray in 3D (normalized, i. e., $\parallel \tilde{\mathrm{X}} \parallel = 1$)
$\mathrm{X}^\mathrm{c} = (R, Z)^\mathrm{T}$	3D vector in cylinder coordinates with R being the radial coordinate, angle φ is usually omitted
$\tilde{\mathrm{X}}^\mathrm{c} = (\tilde{R}, \tilde{Z})^\mathrm{T}$	ray in cylinder coordinates, angle φ usually omitted
$\mathbf{X}^\mathrm{cc}$, $\mathbf{X}^\mathrm{wc}$	homogeneous vectors in camera coordinate system and world coordinate system
$j \in \{1, ..., M\}$	camera j from M rig cameras
$i \in \{1, ..., N\}$	image i from N captured images, i. e., images captured with the whole rig or a monocular camera
$k \in \{1, ..., K\}$	pixel position or 2D image point k
X_s	starting point on outer glass interface in refractive case and camera center in perspective case
$\tilde{\mathrm{X}}_\mathrm{a}$, $\tilde{\mathrm{X}}_\mathrm{g}$, $\tilde{\mathrm{X}}_\mathrm{w}$	rays in air, glass, and water respectively
κ	scaling rays, i. e., $\mathrm{X}_\mathrm{s} + \kappa\tilde{\mathrm{X}}\mathrm{w}$, length of rays/underwater travel distance
d	distance camera center – interface in mm
d_g	glass thickness in mm
n_a, n_g, n_w	indices of refraction (air, glass, water)

λ	wavelength, or color channel as discretized wavelength
α	white balancing of the different color channels
β	offset of the different color channels
B_{∞_λ}	veiling light color in color channel λ
η_λ	attenuation coefficient for color channel λ

Chapter 1

Introduction

The world's oceans are of great interest to mankind, although parts of space are better known and researched than the seafloor. This is caused by the great technical difficulty due to extremely high water pressure at most seafloor regions and greatly hampers exploration and mapping efforts. However, in times of scarce resources and global warming, it becomes increasingly necessary to get to know those unexplored deep sea regions. A first step of exploration is the detailed mapping of the seafloor (bathymetry) for which usually acoustic methods are used because of the water's sound carrying characteristics. For greater detail, acoustic mapping can be complemented by optical methods, i. e., utilizing camera images in methods from the area of computer vision.

This thesis is about adapting existing methods from the area of computer vision to the underwater imaging environment. In order to achieve that, a closer look at existing underwater imaging systems is required. Not all underwater imaging systems are designed for deep sea exploration. A lot of off-the-shelf consumer cameras exist, which are water proof and can be used by divers to capture underwater images (Figure 1.1, left). The water depth that can be reached by those systems is limited, some may be able to reach 100 m. Next to those systems, expensive, custom-made systems, which are for example part of Remotely Operated Vehicles (ROVs) are of interest (Figure 1.1, middle and right). ROVs are underwater robots that are suspended into the water and depend on a tether for power and control, usually provided by a ship on the surface. They are often equipped with manipulators and cameras and are widely used by the offshore industry for example oil and gas companies, but also by scientists. ROVs are constructed for retrieving samples and measuring ocean water parameters like salinity, temperature, etc. While doing so, video footage

Figure 1.1. Left: SLR camera in underwater housing. Middle: ROV Kiel 6000 from Geomar Helmholtz center for Ocean Research. Right: deep sea flat port camera.

is captured, often to aid the pilots to navigate the ROV, however, only recently, applying methods of computer vision to the captured footage has received some interest. AUVs (Autonomous Underwater Vehicles) can navigate through the water on their own on a pre-programmed path. They have limited, on-board power supplies, usually no manipulators, but can record video or single images using a strobe. The limited supply of power often limits the amount of image data that can be captured, especially because of power requirements for adequately lighting the scene. Additionally, AUVs continuously measure ocean water parameters, thus can autonomously create profiles along their pre-defined paths. Both, ROVs and AUVs can reach water depths up to thousands of meters. As an example, the ROV Kiel 6000 of the Geomar Helmholtz Centre for Ocean Research can reach up to 6000 m. This allows to reach more than 90% of the ocean floor[1]. However, the deeper an underwater robot can dive, the more technologically challenging it becomes to withstand the high water pressures.

Apart from bathymetry, other computer vision methods can be applied to the underwater images. In order to map the ocean floor, camera movements can be registered and a mosaic of the floor can be computed. More difficult, but also often more interesting, is the computation of 3D models of objects or larger structures, which allows to measure distances and volumes. The advantages of reconstructing 3D models become apparent when considering the following situation: in order to determine how a volcano came into existence, geologists will go into the field, measure

[1] http://www.geomar.de/en/centre/central-facilities/tlz/rovkiel6000/overview/

physical properties of the deposits such as grain size distribution, thickness and orientation of bedding, but also volcanotectonic features such as faults and joints. Such features are only visible over a broad range of scale, causing the scientists to wander around the outcrop, in order to gain an overview over larger structures. This is fairly easy to do on land. Underwater however, the geologists will need to use a ROV and record many Gigabytes of image data. Then, after the cruise, the video data is viewed and examined. However, it is neither possible to navigate freely, nor possible to move back, away from the structure to get an overview, nor possible to do distance or even volume measurements. When using the captured image data to create a 3D model, however, the geologist can do field work on a computer, long after the cruise, even measure small or larger structures.

Other examples include non-rigid objects of interest like fish, plankton, or other vagile fauna. Biologists for example often need to measure and/or categorize fish or plankton [CLC+06, HS98]. Ideally, this measurement can happen in situ, i. e., the fish do not need to be captured, but are automatically detected in the images and then measured, which is reducing stress for the animals. In this case a synchronized stereo camera rig can be used to capture images of the fish. Similar measurements on a different scale are required in case gas bubbles in the water column need to be investigated. This is of great interest in different areas, e. g., Climatology where it needs to be known how much carbon dioxide (CO_2) or methane (CH_4), which is emitted from deposits buried beneath the seafloor, actually reaches the atmosphere or how much is dissolved in the water. The parameters to be measured include the average gas bubble size, which is usually between 3 mm and 13 mm, the bubble's rise velocity, the total volume of a plume of gas bubbles, the size distribution histogram of a set of bubbles, and changes in average size over time. In the literature, often acoustic methods are applied in order to measure those parameters, e. g., [vDP12]. However, some optical methods exist as well [LdLC03, TZSB10]. Both approaches use only one camera to measure bubbles at a certain, known distance from the camera.

Finally, optical methods and 3D reconstruction can be used to aid vehicle navigation. This is due to the fact that the camera systems are usually rigidly coupled with the vehicle, thus have the same trajectory in

3D space. During 3D reconstruction, the camera path is computed and can be utilized for autonomous vehicle navigation for example in AUVs. In order to achieve that, the reconstruction algorithm needs to run in real time and it needs to incorporate navigation data from the vehicle's sensors. Approaches like that are categorized as SLAM (Simultaneous Localization and Mapping) and are widely used in robotics. A recent approach for autonomous ship hull inspection can be found in [KE13].

Methods for computing mosaics and 3D models from image data, for measuring distances or volumes using stereo camera rigs, or using image data to aid robot navigation, are well established in the area of computer vision for cameras moving through air.

Unfortunately, the water affects image formation in two different ways. First, while traveling through the water, the photons are absorbed and scattered by the water depending on the light's wavelength, causing the captured images to have the typical green or blue hue. Light attenuation is particularly strong in the near infra-red part of the spectrum, and hence many established measuring methods like infra-red-based structured light, e. g., Kinect [HSXS13] or Time-of-Flight cameras [LSBS99] do not work well underwater. If many particles are suspended in the water, viewing distances can be extremely short, e. g., only centimeters in turbid water and even in clear water viewing distances are limited to approximately 30 m.

The second effect on underwater image formation is refraction at the camera's underwater housing, which can usually be categorized into dome ports and flat ports, with different glass thicknesses depending on water depth. The inside of the housing is usually filled with air, outside is water. Consequently, all light rays entering the housing are refracted twice, first at the water-glass interface, and again at the glass-air interface if they do not intersect the housing at a perfectly perpendicular angle. Thus, for all flat port housings, the geometry of rays is distorted and for all not perfectly fitting dome ports as well. The above mentioned applications like mosaicing, 3D reconstruction, and stereo measurements utilize imaging geometry and therefore are affected by refraction. This thesis will investigate how established methods mainly for 3D reconstruction, but also for stereo measurements can be adapted to underwater image formation.

Main Contributions

In order to deal with effects on color, a simplified physical model of light attenuation and scattering that is well established in the literature is parametrized during a newly developed calibration routine and then utilized to correct image colors. For this, the distance between camera and object is required, hence, it can be used to correct the color textures of 3D models after depth estimation.

More important than color correction is the explicit consideration of refraction. Usually, cameras can be modeled geometrically using the perspective camera model. However, due to refraction, the perspective camera model becomes invalid because the extremal light rays do not intersect in the common center of projection anymore, thus violating one of the model's basic assumptions. Therefore, the main contribution of this thesis is the development and analysis of a method for reconstruction that explicitly models refraction at flat port underwater camera housings. In order to achieve that, the housing interface is first parametrized and calibrated. Then, a refractive Structure-from-Motion approach is presented with new algorithms for relative and absolute pose estimation. A major challenge is that projecting 3D points into the camera is computationally very expensive, requiring to solve a 12^{th}-degree polynomial. Therefore, the commonly used reprojection error cannot be used for non-linear optimization because its run-time becomes infeasible once the number of views increases. This challenge is met by introducing a virtual camera error function that allows to project 3D points efficiently and allows for fast non-linear optimization, e. g., bundle adjustment. In order to compute dense depth maps, a refractive plane sweep algorithm is developed.

Parts of the contributions of this thesis have been previously published in:

- Anne Sedlazeck, Kevin Köser, and Reinhard Koch: 3D reconstruction based on underwater video from ROV Kiel 6000 considering underwater imaging conditions, Proc. OCEANS '09. OCEANS 2009-EUROPE [SKK09], Chapters 4 and 6 (perspective SfM on underwater images with calibration and application of a model for underwater light propagation to correct texture colors)

- Robert Wulff, Anne Sedlazeck, and Reinhard Koch: 3D Reconstruction of Archaeological Trenches from Photographs. Proc. Scientifc Computing and Cultural Heritage (SCCH09), 2009. [WSK13], Chapter 6 (3D reconstruction of archaeological trenches based on images captured on orbital trajectory with explicit loop-closing)

- Robert Wulff, Anne Sedlazeck, and Reinhard Koch: Measuring in Automatically Reconstructed 3D Models. Geoinformatik 2010, [WSK10], Chapter 6 (3D reconstruction of archaeological trenches with transformation in geo-referenced coordinate system and analysis of measuring accuracy)

- Anne Sedlazeck and Reinhard Koch: Simulating Deep Sea Underwater Images Using Physical Models for Light Attenuation, Scattering, and Refraction, Proc. of VMV 2011: Vision, Modeling & Visualization [SK11b], Chapter 3 (simulator for rendering underwater images with refraction and an extension of Jaffe-McGlamery Model for underwater light propagation)

- Anne Sedlazeck and Reinhard Koch: Calibration of Housing Parameters for Underwater Stereo-Camera Rigs, Proceedings of the British Machine Vision Conference 2011 [SK11a], Chapter 5 (checkerboard-free calibration of underwater housings for stereo camera rigs using bundle adjustment)

- Anne Sedlazeck and Reinhard Koch: Perspective and Non-perspective Camera Models in Underwater Imaging – Overview and Error Analysis, Outdoor and Large-Scale Real-World Scene Analysis 2012, LNCS vol. 7474 [SK12], Chapters 3 and 4 (state-of-the-art paper concerning calibration and SfM based on underwater images, analysis of the systematic model error introduced by using the perspective camera on underwater images)

- Anne Jordt-Sedlazeck and Reinhard Koch: Refractive Calibration of Underwater Cameras, Proc. of ECCV 2012, LNCS vol. 7576 [JSK12], Chapter 4 (Analysis-by-Synthesis based approach for calibrating flat port underwater housings)

- ▻ Anne Jordt-Sedlazeck and Reinhard Koch: Refractive Structure from Motion on Underwater Images, Proc. of ICCV 2013 [JSK13], Chapter 5 (refractive SfM and comparison to perspective SfM)
- ▻ Anne Jordt-Sedlazeck, Daniel Jung, and Reinhard Koch: Refractive Plane Sweep for Underwater Images, accepted for publication in GCPR Proceedings 2013 [JSJK13], Chapter 5 (dense depth estimation with refractive camera model)

Overview

The following work is organized as follows. In Chapter 2, necessary concepts like projective geometry and Plücker lines are briefly introduced, followed by an introduction to conventional image formation with different camera models. In contrast to classic image formation, Chapter 3 discusses the water's influence on image formation, and additionally gives an overview of the state of the art concerning methods of computer vision applied to underwater images. Chapters 4 and 5 introduce the thesis' main contributions. In Chapter 4, a method for calibrating flat port cameras is presented, while Chapter 5 focuses on introducing the refractive SfM routine and the refractive plane sweep, and compares the established perspective routine to the newly presented refractive routine. Chapter 6 gives a more detailed overview over different applications of the proposed methods including results. A conclusion is presented in Chapter 7.

Chapter 2

General Concepts and Classic Image Formation

The major goal of this thesis is to investigate feasibility and adaptations for using computer vision algorithms for reconstruction on underwater images. Crucial to this investigation are general concepts like projective geometry, but especially image formation in air and water. Therefore, this chapter aims at giving an introduction to basic geometric concepts and image formation in air with a classification of different possible camera models.

2.1 Projective Geometry

In this thesis, the concept of the projective space [HZ04] is used to describe the geometrical theory of the algorithms. Therefore, this section will introduce the projective space and the major geometrical concepts and notations used throughout the thesis.

In order to be able to work with points that are infinitely far away and to deal with transformations of points and scenes linearly, the euclidean space $\mathbb{R}^n$ is extended by one dimension to form the projective space $\mathbb{P}^n$ (Figure 2.1). Let's consider the example of two dimensions. The additional dimension of $\mathbb{P}^2$ causes points in the 2D euclidean space to become lines in the 2D projective space. The additional dimension is denoted by w, and hence points in the 2D projective space are of the form:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ w \end{bmatrix}. \tag{2.1.1}$$

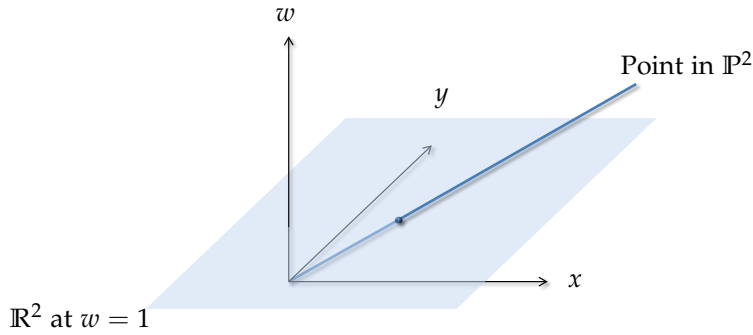

Figure 2.1. A point in $\mathbb{P}^2$ is a line intersecting the $w = 1$ plane. Since $\mathbb{R}^2$ is embedded into $\mathbb{P}^2$ at that plane, the corresponding point in $\mathbb{R}^2$ is the intersection between plane and line. Points at infinity are of the form $(x, y, 0)^{\mathrm{T}}$, thus are within the xy-plane of $\mathbb{P}^2$ and do not intersect the $w = 1$ plane.

The euclidean space is embedded in the projective space as the plane at $w = 1$. Therefore, a euclidean point can easily be retrieved from $\mathbf{x}$ by division with w. Consequently, points in the projective space are only determined up to scale. A projective point is a line through the origin of the coordinate system and, in case of euclidean points, also through the plane denoting the euclidean space at $w = 1$. A Point at infinity cannot be described in the euclidean space. In the projective space however, it is found to be a direction in the plane $w = 0$. The point that is the intersection of two parallel lines, and therefore lies on the plane of infinity, is described to be a direction of x- and y-components with $w = 0$. Another advantage of using the projective space is the linear usage of transformations for points or whole scenes. A transformation consisting of, e. g., a rotation

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 \\ r_3 & r_4 \end{bmatrix} \tag{2.1.2}$$

and a translation $\mathsf{C} = (c_x, c_y)^T$ can be expressed by the matrix $\mathbf{T}$:

$$\mathbf{x}' = \mathbf{T}\mathbf{x} = \begin{bmatrix} r_1 & r_2 & c_x \\ r_3 & r_4 & c_y \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}. \tag{2.1.3}$$

Transformations are classified depending on their degrees of freedom (DoF). In $\mathbb{P}^2$, euclidean transformations have six DoF, allowing to describe rotation and translation. Similarity transforms have an additional scale factor, thus have seven DoF. Finally, projective transformations have eight DoF, and hence the 3×3 transformation matrix is determined up to scale. That means that the bottom row is not of the form $(0, 0, 1)$ anymore and can transfer points onto or away from the plane at infinity.

The examples shown here were all in the 2D space. However, the extension to three dimensions is straight forward leading to up-to-scale four-vectors for points and 4×4 transformation matrices.

2.1.1 Plücker Lines

Compared to 3D points, lines in 3D space are more difficult to describe because they have four degrees of freedom. One possible representation of a line consists of a starting point $\mathsf{P} \in \mathbb{R}^3$ and a direction $\tilde{\mathsf{D}} \in \mathbb{R}^3$. Points X on the line are then described by:

$$\mathsf{X} = \mathsf{P} + \kappa\tilde{\mathsf{D}}, \quad \kappa \in \mathbb{R}. \tag{2.1.4}$$

In addition, Plücker lines or matrices are used in this thesis and the following description is based on [HZ04], [SRL06], and [Ple03]. If two homogeneous points $\mathbf{X}$ and $\mathbf{Y}$ are given, the corresponding Plücker matrix $\mathbf{L}$ is determined by:

$$\mathbf{L} = \mathbf{X}\mathbf{Y}^T - \mathbf{Y}\mathbf{X}^T. \tag{2.1.5}$$

$\mathbf{L}$ is a 4×4 skew symmetric ($\mathbf{L} = -\mathbf{L}^T$) matrix that is determined up to scale and has $\det(\mathbf{L}) = 0$, thus having four degrees of freedom. Note that $\mathbf{L}$ is independent of the points used for its computation. A transformation

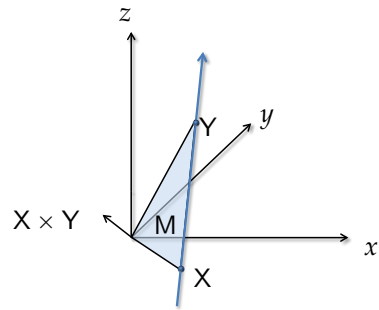

Figure 2.2. The Plücker line (blue arrow) is determined by the two points X and Y. The moment M is the plane defined by the cross product $X \times Y$.

H is applied to **L** as follows:

$$\mathbf{L}' = \mathbf{H}\mathbf{L}\mathbf{H}^{\mathrm{T}}. \tag{2.1.6}$$

Apart from using a matrix to represent a Plücker line, there is also a vector representation based on the euclidean versions of the vectors **X** and **Y**:

$$L = \begin{bmatrix} \underbrace{Y - X}_{=D} \\ \underbrace{X \times Y}_{=M} \end{bmatrix}, \tag{2.1.7}$$

where D is the direction of the line and M the normal of the plane A in Figure 2.2, called moment. As in case of the Plücker matrix, the points on the line for determining L can be chosen arbitrarily. In order to assure the four degrees of freedom, L is determined up to scale only and L is a line in 3D space if and only if $D^{\mathrm{T}}M = 0$.

If $\| D \| = 1$, then $D \times M$ is the point on the line closest to the origin. The plane M spanned by the origin and D is defined by the euclidean vectors of X and Y by $M = X \times Y$. A transform of a point X in space,

defined by a rotation matrix $\mathbf{R}$ and a translation vector C is defined by:

$$\mathbf{X}' = \begin{bmatrix} \mathbf{R} & C \\ 0^T & 1 \end{bmatrix} \mathbf{X}. \tag{2.1.8}$$

The corresponding transform of a Plücker line L is defined by:

$$L' = \begin{bmatrix} \mathbf{R} & 0 \\ -[C]_\times \mathbf{R} & \mathbf{R} \end{bmatrix} L. \tag{2.1.9}$$

Thus, even though the transformation matrix is a 6×6 matrix, transformations can be described as easily as in the case of 3D points.

The intersection of two Plücker vectors can be determined easily. Let $L_1 = (D_1, M_1)$ be the first line and $L_2 = (D_2, M_2)$ be the second line, then both lines intersect if and only if

$$D_2^T M_1 + M_2^T D_1 = 0. \tag{2.1.10}$$

2.1.2 Coordinate Systems

In this thesis, the coordinate systems are defined as follows. Let $\mathbf{X}^{wc} \in \mathbb{P}^3$ be a 3D point in the world coordinate system and $\mathbf{X}^{cc} \in \mathbb{P}^3$ be the same point in a local camera coordinate system. Let $\mathbf{R}$ be a 3×3 rotation matrix and $C \in \mathbb{R}^3$ be a euclidean translation vector. With

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & C \\ 0 & 1 \end{bmatrix} \tag{2.1.11}$$

the point $\mathbf{X}^{cc}$ can be transformed into the world coordinate system (local-to-global transform):

$$\mathbf{X}^{wc} = \mathbf{T}\mathbf{X}^{cc}. \tag{2.1.12}$$

Note that this corresponds to:

$$X^{wc} = \mathbf{R}X^{cc} + C \tag{2.1.13}$$

in euclidean coordinates. On the other hand:

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{C} \\ 0 & 1 \end{bmatrix} \tag{2.1.14}$$

transforms the point in the world coordinate system into a local coordinate system (global-to-local transform):

$$\mathbf{X}^{cc} = \mathbf{T}^{-1}\mathbf{X}^{wc}. \tag{2.1.15}$$

Those transformations between the camera and world coordinate systems are utilized in the next section, which introduces different camera models.

2.2 Geometry of Camera Models

This section deduces, how a camera projects a 3D point in space onto a 2D point in the image and the inverse of such a projection, the back-projection, i. e., the computation of a 3D ray corresponding to a 2D pixel in the image. There are different possibilities and degrees of accuracy with which to model a camera. The first one considered in this thesis is the pinhole camera model.

2.2.1 Pinhole Camera Model with Distortion

The camera obscura, a box with a very small hole in one side, through which the light enters, produces an image of the outside world on the side of the box opposite to the hole. Replacing the backside of the box with a plane able to record the amount of light hitting the plane at different pixel locations, for example, a CCD (Charged Coupled Device) chip, allows to record the image being projected – a very simple camera. Detailed introductions to the pinhole camera model can be found in a variety of books for example in [HZ04, Sze11].

As can be seen in Figure 2.3, a ray connecting a scene point with the pinhole or center of projection intersects the image plane causing the sensor to record its energy. Note that in Figure 2.3 the image plane is

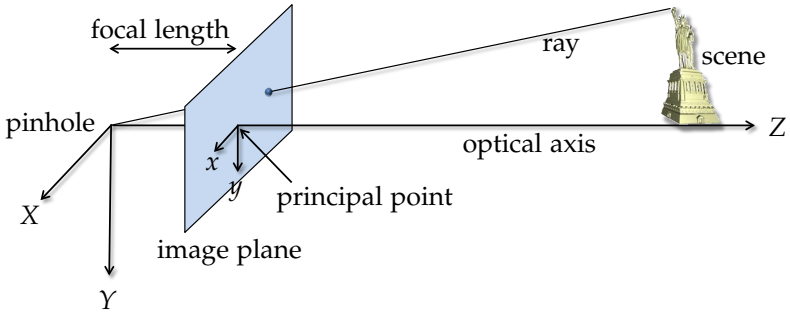

Figure 2.3. The Pinhole camera model. A ray from the scene enters the camera through the pinhole or center of projection, crossing the optical axis. After that, it is recorded in the image at the point of intersection with the image plane. Note that usually the image plane is located behind the pinhole causing the image to be up-side-down. However, when sketching the model, the image plane can just as well be drawn in front of the pinhole as depicted in the image.

set in front of the pinhole instead of behind the pinhole as is the case in a physical camera. However, both are mathematically equivalent. All rays contributing to the image pass through the center of projection or the pinhole. In reality, a lens system replaces the hole, but the pinhole model is sufficient for explaining the imaging process.

The transformation of a point in the camera coordinate system into image coordinates is described by the camera matrix $\mathbf{K}$ containing the intrinsic camera parameters:

$$\mathbf{K} = \begin{bmatrix} f & s & c_x \\ 0 & arf & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.2.1}$$

where f denotes the focal length and ar denotes the aspect ratio (which is one in case of square pixels), s describes a possible skew of the single pixels on the CCD-chip, and c_x and c_y denote the principal point, the intersection of the optical axis with the image plane.

If $\mathbf{X} = (X, Y, Z, 1)^T$ is a 3D point that is to be projected onto the point $\mathbf{x} = (x, y, z)^T$ in the image plane, the projection can be derived using

Figure 2.4. Different distortion effects viewed on a characteristic pattern. From left to right: original pattern, radial distortion with barrel effect, radial distortion with cushion effect, tangential distortion only, radial and tangential distortion. Note that tangential distortion is exaggerated in order to better visualize the effect.

Figure 2.3 and the camera matrix. In addition, the camera can have a rotation and translation with respect to the world coordinate system. This rotation and translation are described with a rotation matrix $\mathbf{R}$ and a translation vector C. The projection matrix $\mathbf{P}$, which combines camera matrix, rotation, and translation is assembled as follows:

$$\mathbf{P} = \mathbf{K}[\mathbf{R}^{\mathrm{T}}| - \mathbf{R}^{\mathrm{T}}\mathsf{C}] = \mathbf{K} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \left[\begin{array}{c|c} \mathbf{R}^{\mathrm{T}} & -\mathbf{R}^{\mathrm{T}}\mathsf{C} \\ 0^{\mathrm{T}} & 1 \end{array}\right], \tag{2.2.2}$$

resulting in the up-to-scale projection:

$$\rho\mathbf{x} = \mathbf{P}\mathbf{X}, \quad \rho \in \mathbb{R}. \tag{2.2.3}$$

A simple extension of the pinhole camera model to account for imperfect lens systems also includes tangential and radial distortion [McG04]. Radial distortion moves pixels along the radius and is basically caused by the usage of a lens system itself – the magnification is larger or smaller depending on the distance of the ray from the principal point. Tangential distortion is a translation perpendicular to the radius, which is caused by de-centered lenses. Radial distortion is usually classified into cushion and barrel distortion due to the deformation in the image (Figure 2.4). Tangential distortion can often be neglected due to very precisely manufactured lens systems. Instead of explicitly computing the refraction of rays through the lenses, both distortion types are modeled by a polynomial. Let (x, y) be a point in camera coordinates. With $r = \sqrt{x^2 + y^2}$, the distorted

points can be estimated by [Bro71]:

$$\begin{aligned} x_\mathrm{d} =& x + (x - c_\mathrm{x})(r_1 r^2 + r_2 r^4 + r_3 r^6 + ...) + \\ & [t_1(r^2 + 2(x - c_\mathrm{x})^2) + 2t_2(x - c_\mathrm{x})(y - c_\mathrm{y})] + [1 + t_3 r^2 + ...] \\ y_\mathrm{d} =& y + (y - c_\mathrm{y})(r_1 r^2 + r_2 r^4 + r_3 r^6 + ...) + \\ & [2t_1(x - c_\mathrm{x})(y - c_\mathrm{y}) + t_2(r^2 + 2(y - c_\mathrm{y})^2)] + [1 + t_3 r^2 + ...], \end{aligned} \tag{2.2.4}$$

where r_i are the coefficients for the radial distortion and t_i are the coefficients for the tangential or de-centering distortion. Note that the transformation into image coordinates is applied after computing distortion. In the literature, the models describing distortion mostly differ by the exact definition of the polynomials and the number of coefficients used, e. g., Brown [Bro71] uses three for radial and tangential distortion. [Tsa87] uses a checkerboard pattern for calibration without tangential distortion and only one coefficient for radial distortion. Heikkilä and Silvén [HS97b] use two coefficients each and describe a calibration routine using a 3D calibration target. One coefficient for radial distortion and none for tangential distortion are used by Zhang [Zha99]. A Matlab implementation of the above mentioned calibration routines that is widely used has been implemented by Bouget[1]. A tool[2] that not only calibrates one camera, but rigs that may even contain active depth cameras is described in [SBK08]. Here, planar calibration patterns are used for calibration (see Chapter 4).

2.2.2 Entrance-Pupil Camera Model

Sometimes, more detailed models including explicitly modeled lenses are required to explain effects like depth of field, focus, or photometric properties. Exemplary models can be found in a variety of works from the areas of photogrammetry and computer vision [AA02, Hec05, McG04, JHG99, HMS99]. Usually, cameras contain more than one lens, a whole system working together, in order to produce the final image. The aperture can be a hole or a diaphragm, which effectively limits the number of rays that can actually reach the image sensor and is usually located somewhere

[1] http:www.vision.caltech.edubouguetjcalib_doc

[2] http://www.mip.informatik.uni-kiel.de/tiki-index.php?page=Calibration

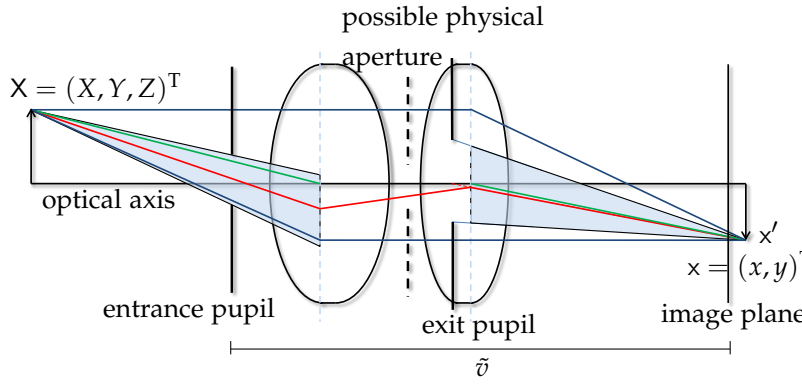

Figure 2.5. Pupil lens model. From scene point X light rays enter the system through the entrance pupil, are refracted by the lens system, and exit through the exit pupil. A scene point is imaged sharply, i. e., as a single point in x', not necessarily on the image plane in x. Adapted from [AA02].

in between the different lenses. Here, we consider the work of Aggarwal et al. [AA02], who extent the Gaussian thick lens camera model, which abstracts from modeling lenses explicitly and instead considers the camera to have one thick lens. [AA02] models the aperture to be an entrance and an exit pupil on the optical axis on the object and image side respectively (Figure 2.5). Entrance and exit pupil do not necessarily coincide with the real aperture and are purely virtual points. In [AA02], the authors derive complete imaging equations for the pupil centric imaging model and come the following conclusion, which is relevant for this thesis:

> "...the pupil-centric model for an imaging system with a frontal sensor and fixed imaging parameters is geometrically equivalent to a pin-hole model with the following parameters: the pin-hole is located at the center of the entrance pupil E_1 and the distance of the sensor plane from the pin-hole is $\tilde{v}$..."[3]

Where $\tilde{v}$ denotes a distance depending on the focal length, the position of the sensor on the optical axis, and the entrance pupil location. In essence,

[3] [AA02], p. 201

this means that the center of projection of a camera can possibly be found in front of the physical camera and its lens system. In Figure 2.5 it can be seen that the rays coming from X enter the system through the entrance pupil are refracted by the lens system, exit through the exit pupil and meet eventually in the conjugate point x'. However, only if the image plane and the conjugate point coincide, the object point can be imaged sharply, i. e., as a single point. As this cannot be the case for object points at different distances from the camera, it explains the phenomenon of depth of field, and hence the need of focusing the area on the z-axis that is required to be imaged sharply. All points that are not conjugate for the current setting will appear as blobs in the image. For a complete derivation of the equations for a thin lens model refer to [Hec05].

2.2.3 General and Axial Camera Models

A common characteristic of the camera models described above is the assumption that the camera has one center of projection in which all rays of light intersect. However, cameras exist, for which this assumption does not hold, e. g., catadioptric cameras, fish-eye cameras, camera systems consisting of more than one camera that deliver one common image [Ple03, SGN03], and also cameras in underwater housings. In this case, more general camera models need to be used. Instead of having a few parameters describing how a ray is computed for each pixel in the image, each ray for each pixel is determined independently and described by its starting point and direction. However, determining a calibration for such a camera is difficult to achieve robustly. In [SGN03] cameras are classified due to their distortions:

Perspective cameras have a center of projection and do not have any distortion, i. e., the ideal pinhole camera.

Single-View-Point (SVP) cameras encompass for example wide-angle cameras or perspective cameras with lens distortion. All rays pass through a common center of projection. Image distortions can be compensated if the camera model is known and calibrated.

Non-Single-View-Point (nSVP) cameras are for example clusters of rigidly coupled cameras, some catadioptric cameras, but also underwater

cameras. In order to compensate for distortions, the camera model and the 3D scene structure need to be known.

In order to deal with nSVP cameras, Grossberg et al. [GN05] introduce a generic camera model, where each pixel captures one ray of light, called raxel. Each raxel is modeled by a starting point and a direction that do not necessarily coincide with the physical camera. Note that the actual ray and its possible refractions and reflections are treated as a black box. One major assumption in [GN05] is that the bundle of rays has a singularity (not true for example for orthographic cameras). The locus of this singularity is the caustic, a geometric construct to which all rays are tangents, that uniquely describes the camera. Caustics are computed by differentiating the mapping from image coordinates $(x, y) \in \mathbb{R}^2$ to rays:

$$\mathsf{X}(x, y, \kappa) = \begin{bmatrix} X(x, y, \kappa) \\ Y(x, y, \kappa) \\ Z(x, y, \kappa) \end{bmatrix} = \mathsf{X}_s(x, y) + \kappa \tilde{\mathsf{X}}(x, y), \tag{2.2.5}$$

with X_s being the starting point and $\tilde{\mathsf{X}}$ being the direction of the ray. $\kappa \in \mathbb{R}$ is the length of the ray. In order to find the singularity, the determinant of the Jacobi matrix of ray function $\mathsf{X}(x, y, \kappa)$ is computed and set to zero:

$$\det(\mathbf{J}(\mathsf{X}(x, y, \kappa))) = 0. \tag{2.2.6}$$

This allows to solve for the parameter κ and thus to determine the caustic point for each pixel position (x, y). Note that for single-view-point cameras, the caustic is only one point, the center of projection. In case of non-single-view-point cameras however, the size of the caustic provides a possibility of quantifying the deviation from the single-view-point camera. Even more generic is the camera model described by Sturm et al. [SR04, SRL06], where single rays are parametrized by starting point and direction as well. However, it is not assumed that a caustic exists, merely that neighboring pixels have neighboring rays, thus making this model even more general.

A different classification, based on ray geometry is described in [Ple03, LHK08]. Here, general cameras are simulated by using a rig of several cameras, which leads to the following classification (refer also to Figure 2.6):

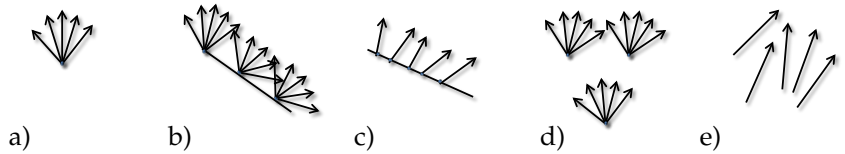

Figure 2.6. Camera types by ray geometry. a) single-view-point perspective camera, b) locally central, axial camera, c) axial camera, d) locally central, general camera, e) general camera.

a) Perspective camera with a single view point in which all rays intersect (SVP camera)

b) Locally central, axial camera, a general camera comprised of a rig of several perspective cameras, where all centers of projection lie on a common axis (nSVP camera).

c) Axial camera, a more general camera where all rays intersect a common axis (nSVP camera).

d) Locally central, general camera, a rig made of more than two perspective cameras, where the centers of projection do not lie on a common axis (nSVP camera).

e) General camera, the most general model, where no assumptions are made for the light rays (nSVP camera).

2.3 Summary

In this chapter, projective geometry with Plücker Lines and coordinate system transformations has been introduced briefly as a perquisite to understand image formation. In a more detailed discussion, different models for image formation, the pinhole camera model, the perspective camera, but also more general camera models that do no fulfill the single view point assumption were introduced and classified. All of the explained concepts are valid for cameras used in air. However, the remainder of

this thesis will investigate how water affects image formation and what adaptations to 3D reconstruction methods are required.

Chapter 3

Underwater Image Formation

Chapter 2 describes how a camera in air captures an image of a scene. When taking the camera below water, precautions must be taken. In order to avoid electric shortening or implosion, the camera needs to be confined in an underwater housing that can deal with the pressure of the water. Since pressure increases with depth, underwater housings that can capture images at water depths of several thousand meters are usually made of titanium and have glass ports that can be several centimeters thick (refer to [MBJ09] for more information on pressure housings).

When the light enters the camera housing, it is refracted twice: first at the water-glass interface, then at the glass-air interface. This means that all light rays entering the camera not parallel to the normal of the underwater housing port change their direction, and thus are affecting viewing geometry. However, even before the light rays enter the underwater housing, the light is affected by the water. Photons collide with water molecules and other matter causing absorption and scattering effects, both of which are wavelength dependent, and hence change the color recorded by the image sensor. This leads to the typical green or blue hue and low contrast in underwater images. First, the effects on color will be examined and modeled in this chapter. After that, the state of the art of modeling refractive effects and the model used in this thesis will be discussed.

3.1 Effects on Color

In addition to the described dominant green or blue colors in underwater images, vision is limited, sometimes to a few centimeters, sometimes up to 30 m. Those differences between light propagation through air and light propagation through water are mainly caused by particles, which

are far more densely packed in water compared to air. Single photons of the light beam collide and interact with the particles in the water (and to a far lesser extent with the particles in the air). These interactions are mainly categorized into absorption and scattering, i. e., photons either disappear or change their direction of travel after such a collision. The physical explanation and modeling of these effects is very complex and out of the scope of this work. Therefore, the next section introduces the basic physical principles required for the Jaffe-McGlamery model [Jaf90, McG75], a basic model, which can be utilized for computer vision purposes. The required basic radiometric quantities can be found in the Appendix A.1. [SK11b], which has been published in 2011, introduces a simulator for rendering underwater images that extends the Jaffe-McGlamery model, and parts of the following explanation are based on this work. After explaining how the water influences scene color, it is interesting if and to what extent it is possible to reverse the effect, thus restore image color. Unfortunately, the fairly complex Jaffe-McGlamery model cannot be easily inverted such that it is applicable to image color correction. However, in the literature, several works exist that simplify the Jaffe-McGlamery model until it can be used. Consequently, these methods are still based on a physical model for underwater light propagation. Others apply a set of heuristically determined filters, e. g., histogram stretching on the different color channels. In the following section, first, physical principles will be introduced briefly, then, simulating underwater color using the Jaffe-McGlamery model with some extensions will be explained, followed by a brief overview of methods for image color correction.

3.1.1 Physical Principles

This section is based on the works of Mobley [Mob94], Jerlov [Jer76], Hecht [Hec05], and Dera [Der92] and focuses on the physical principles. It concentrates on the visible part of the spectrum with comments on the propagation of near infra-red light. The physical principles described here only concern inherent optical properties, i. e., properties that depend on the medium only, more concretely light absorption and scattering.

Figure 3.1. Different water bodies and their effects on underwater image color. From left to right: swimming pool, Baltic sea (image by Florian Huber), and lab experiment with tea.

Absorption

Absorption describes the loss of photons out of a beam of light traveling through water, mainly by water molecules, but also by collisions between photons and other, differently sized particles, such as organic, yellow substances (dissolved remains of animals and plants suspended in the water) and other organic or an-organic suspended particles. In addition to the type of particle, absorption is also depending on the light's wavelength and the concentration of the particles in the water body. The dependence on wavelength is documented in the absorption spectra for different types of particles. In the range of visible light those spectra do not differ strongly for clear, salt, and distilled water. However, the absorption spectrum of organic substances differs considerably from the absorption spectrum of clear water, explaining why organic substances cause very limited visibility or water colors very different from the typical blue of clean water (see also Figure 3.1). The absorption spectra of the different molecules overlap especially in the infra-red part of the spectrum and cause a very strong absorption of infra-red light – over 50% are absorbed by only centimeters of water.

Multiple absorption spectra, describing different absorption effects for different wavelengths and particles, can be summarized using the volume absorption coefficient $a(\lambda)$ measured in $[\mathrm{m}^{-1}]$, a function depending on the wavelength λ. It can be measured by special equipment and the effects of different kinds of particles in the water aside from the actual water

molecules are summarized:

$$a = a_{\mathrm{w}} + a_{\mathrm{y}} + a_{\mathrm{p}} + a_{\mathrm{s}} + a_{\mathrm{d}}, \tag{3.1.1}$$

with a_{w} being the pure water absorption, a_{s} the absorption due to sea salt, a_{y} the absorption due to yellow substances, a_{p} the absorption due to suspended particles, and a_{d} the absorption due to other artificial contaminants. All those initiators of absorption have different absorption spectra, i. e., show different dependence on wavelength. For example, the yellow substances are highly concentrated in some rivers causing the water to appear yellow or brown. In the ocean however, they usually exist in very low concentrations, having far less impact on the overall absorption.

For a beam of light traveling through the water for a distance κ, $\in \mathbb{R}$ in [m], Lambert's law states that the loss of photons out of the beam is exponential:

$$E(\kappa, \lambda) = E(0, \lambda) e^{-a(\lambda)\kappa} \ \ [\mathrm{Wm}^{-2}], \tag{3.1.2}$$

where $E(0, \lambda)$ and $E(\kappa, \lambda)$ are the irradiance before and after traveling the distance κ through the water.

Scattering

Scattering of light is a complicated phenomenon that can be analyzed on a molecular scale, where a photon colliding with a molecule is absorbed and another photon is immediately released. If the wavelength after emitting is the same as before, elastic scattering occurred, an assumption made in [Mob94]. Taking into account how different waves of light overlap, this allows to explain phenomenons like transmission, refraction, and reflection, which are usually considered on a macroscopic scale [Hec05]. A multitude of different models and explanations for scattering effects can be found in the literature [Hec05, Der92, Mob94].

In this thesis, scattering is considered to be a random change of a photon's direction after colliding with a particle. As in case of absorption, particles can be water molecules, but also other matter found in the water body. Consequently, particles can have different sizes and concentrations and scattering is also depending on the wavelength. In contrast to absorption, the modeling of scattering effects requires to take into account the

angle $\psi \in [0, \pi]$ towards which the light is scattered. Note that scattering is symmetric around the axis formed by the light's direction of travel [Der92]. The dependence on ψ is described by the volume scattering function (VSF) $\beta(\psi, \lambda)[\mathrm{sr}^{-1}\mathrm{m}^{-1}]$ (sr - steradians, unit of solid angle A.1). In order to determine the overall scattering coefficient $b(\lambda)$ similar to the absorption coefficient $a(\lambda)$, the VSF needs to be integrated over all directions:

$$b(\lambda) = 2\pi \int_0^\pi \beta(\psi, \lambda) \sin\psi \, \mathrm{d}\psi \quad [\mathrm{m}^{-1}], \tag{3.1.3}$$

thus $b(\lambda)$ describes the number of photons that are scattered out of a beam of light while traveling a certain distance:

$$E(\kappa, \lambda) = E(0, \lambda) e^{-b(\lambda)\kappa} \quad [\mathrm{Wm}^{-2}]. \tag{3.1.4}$$

Consequently, the loss of photons from a beam per traveling distance can be summarized in the attenuation coefficient:

$$\eta(\lambda) = a(\lambda) + b(\lambda) \quad [\mathrm{m}^{-1}], \tag{3.1.5}$$

i. e., the loss of irradiance after a distance $\kappa[\mathrm{m}]$ is:

$$E(\kappa, \lambda) = E(0, \lambda) e^{-\eta(\lambda)\kappa} \quad [\mathrm{Wm}^{-2}]. \tag{3.1.6}$$

Another major difference to absorption is that apart from photons being scattered out of a beam of light, photons can also be scattered into a beam of light, thus causing in increase in irradiance. In order to model the gain in irradiance, the scattering angle ψ needs to be taken into account, thus the Volume Scattering Functions needs to be parametrized. From the models described in the literature, a combination of the Einstein Smoluchowski and the Kopelevich model, which are both introduced in [Mob94], will be used in this thesis.

Scattering at sea water, i. e., very small particles is explained by the Einstein Smoluchowski theory, where scattering is assumed to happen at local fluctuations of molecule densities, i. e., scattering particles are considered to be little, spontaneous clusters of water molecules. The

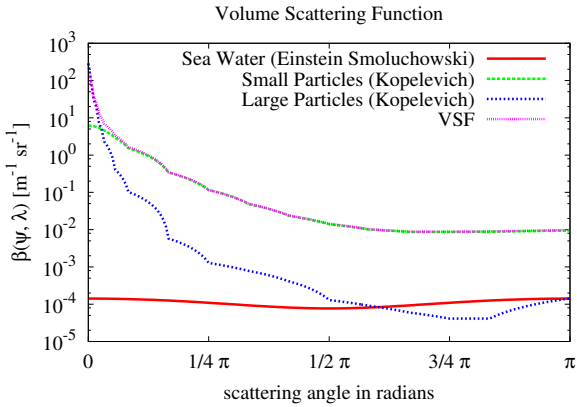

Figure 3.2. Volume Scattering Function for the red color channel using exemplary parameters. Shown is the resulting function (magenta) and its additive components derived from Einstein Smoluchowski (red), Kopelevich small particles (green) and Kopelevich large particles (blue).

resulting model equation for scattering at sea water is:

$$\beta_w(\psi,\lambda) = \beta_w(90^\circ,\lambda_0)\left(\frac{\lambda_0}{\lambda}\right)^{4.32}(1+0.835\cos^2\psi)\ \ [\mathrm{sr}^{-1}\mathrm{m}^{-1}], \quad (3.1.7)$$

where values for $\lambda_0 = 440$ and
$\beta_w(90^\circ,\lambda_0) \in \{0.000284\mathrm{sr}^{-1}\mathrm{m}^{-1}, 0.000146\mathrm{sr}^{-1}\mathrm{m}^{-1}, 0.00008\mathrm{sr}^{-1}\mathrm{m}^{-1}\}$, for the three color channels respectively, can be found in [Mob94].

The Kopelevich model considers small particles $< 1\ \mu$m, for example minerals, and large particles $> 1\ \mu$m of biological origin separately:

$$\beta_s(\psi,\lambda) = \nu_s\beta_s^*(\psi)\left(\frac{\lambda_0}{\lambda}\right)^{1.7}\ \ [\mathrm{sr}^{-1}\mathrm{m}^{-1}] \quad (3.1.8)$$

$$\beta_l(\psi,\lambda) = \nu_l\beta_l^*(\psi)\left(\frac{\lambda_0}{\lambda}\right)^{0.3}\ \ [\mathrm{sr}^{-1}\mathrm{m}^{-1}], \quad (3.1.9)$$

where a set of discrete values for β_s^* $\beta_l^*(\psi)$ can be found in [Mob94]. The

necessary interpolation of the missing values is responsible for the bumps in the Kopelevich functions seen in Figure 3.2. ν_s and ν_l are the small and large particle concentrations respectively that vary with the different water bodies. The resulting VSF used in the remainder of this thesis is the sum of all three components:

$$\beta(\psi,\lambda) = \beta_w(\psi,\lambda) + \beta_s(\psi,\lambda) + \beta_l(\psi,\lambda) \ \ [\mathrm{sr}^{-1}\mathrm{m}^{-1}]. \tag{3.1.10}$$

Figure 3.2 shows the resulting VSF for the red color channel. It can be seen that most of the scattering happens at the small angles close to zero. When modeling the gain by scattering effects, β is separated into forward-scattering $\psi \in [0, \pi/2]$ and backward-scattering $\psi \in [\pi/2, \pi]$. Small-angle-forward scattering can then be approximated using linear filters [Vos91], which have low-pass character [SK04]. Backward-scattering is modeled explicitly using the light sources and the VSF.

Note that the values for the models, especially of the parametrization of the VSF are based on measurements conducted in real water bodies.

Measurements in Real Water Bodies

Instruments exist to measure the attenuation coefficient η in real water bodies. Measurements for different wave lengths in different kinds of oceans or pure water show that the attenuation coefficient is very different for different wavelengths but also for different water bodies. Figure 3.3 shows the wavelength dependence of absorption and scattering coefficients for optically and chemically pure water based on [MP77]. When it comes to measuring the absorption and scattering coefficients for pure water, different authors come to different conclusions because it is very difficult to purify water. In addition, there are several different methods and instruments to measure the coefficients (refer to [Jer76]), and hence the resulting values differ depending on the method of measurement. However, the values in Figure 3.3 give the reader some idea about the order of magnitude and serve as an approximation of a lower bound for other types of water. Once higher concentration of other matter exists in the water, the coefficients rise considerably.

In this thesis, RGB images are utilized in image processing, hence, the

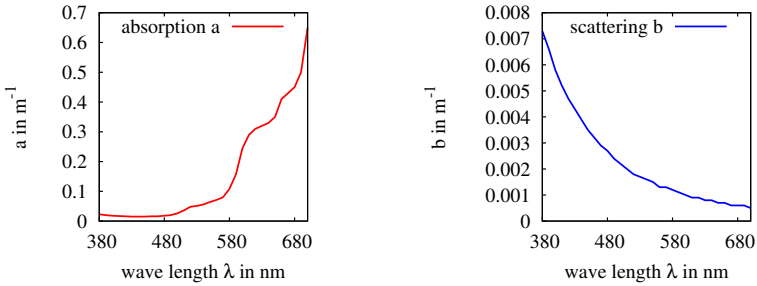

Figure 3.3. The attenuation coefficient $\eta = a + b$, is the sum of the absorption a on the left and the scattering coefficient b on the right. Plotted are values measured for pure water found in [MP77].

light is considered to be captured in three narrow bands of the visible light spectrum (red, green, and blue). Consequently, in the following, all wavelength-dependent effects like attenuation and backscatter of light will be considered for the three color channels and not for the whole spectrum. In order to show that infra-red light and with it standard Time-of-Flight cameras or infra-red based structured-light cameras cannot be used underwater, coefficients and transmittance for $\lambda = 800\,\text{nm}$ are considered here in addition to the three standard color channels. When entering the infra-red part of the spectrum, the attenuation coefficients rise quickly. For $\lambda = 800\,\text{nm}$, the coefficient is $\eta = 2.051\,\text{m}^{-1}$ according to [Jer76]. Note that for larger wavelengths than $\lambda = 800\,\text{nm}$, the attenuation coefficient η is even larger [Mob94].

In Figure 3.4, the transmittance, i. e., the percentage of light that is left after a certain traveling distance in m is plotted, using the attenuation coefficients for pure water as shown in Figure 3.3. Red, green, and blue are depicted in the corresponding colors and black shows the near infra-red behavior, at $\lambda = 800\,\text{nm}$. Exact values for 90%, 50%, 10%, and 1% are presented in Table 3.1 and show that active cameras based on infra-red light, like Time-of-Flight or ir-based, structured light, simply cannot be used in underwater imaging.

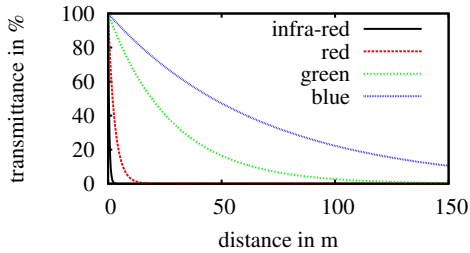

Figure 3.4. Transmittance for infra-red and the three color channels for pure water (infra-red $\lambda = 800\,\text{nm}$, red $\lambda = 650\,\text{nm}$, green $\lambda = 510\,\text{nm}$, blue $\lambda = 440\,\text{nm}$).

Table 3.1. Transmittance values for the three color channels and infra-red light for pure water.

λ in [nm]	attenuation η in m^{-1}	90% left after (in [m])	50% left after (in [m])	10% left after (in [m])	1% left after (in [m])
440 (blue)	0.015	5.545	36.48	121.2	242.4
510 (green)	0.036	2.773	18.24	60.59	121.2
650 (red)	0.350	0.301	1.98	6.579	13.16
800 (near infra-red)	2.051	0.051	0.3381	1.123	2.246

3.1.2 Adapted Jaffe-McGlamery Model

The previous section described the basic physical principles behind underwater light propagation. When working in the areas of computer vision or computer graphics, the water's influence on the imaging process needs to be modeled. The previous section concerning transmittance in pure water already gave an idea about how far the light of the wavelengths corresponding to the different color channels can travel through water. Once the red part of the spectrum has been almost completely absorbed, for example, reconstruction of the correct red color will not be possible anymore. In the literature, most methods based on physical models for underwater

light propagation are build on the Jaffe-McGlamery model [Jaf90, McG75], e. g., [SK04, TS06, NCdB09]. In [SK11b], an extension of the model, which will be explained now has already been published, allowing a simulator to render underwater images using color and multiple light sources, to compute shadows, and to compute refraction at underwater housings.

A lot of methods for rendering water or participating media exist in the literature, which are based on different underlying mathematical principles. For example, Jensen et al. [JC98] use photon maps that allow caustic computation. In the context of underwater color/lighting, caustics are flickering patterns caused by light refraction at waves on the surface. A photon-map based bio-optical model with absorption, elastic and inelastic scattering can be found in [GSMA08]. Mobley demonstrates a different approach by solving the radiative transfer equation analytically. Often, the ocean surface including waves is to be rendered and in this case, correct water color also needs to be computed based on the underlying physical principles. However, in this case the model can be simplified. Refer to [DCGG11] for an overview and to [PA01] for a model that is similar to the Jaffe-McGlamery model. Two other cases of rendering water bodies close to the surface, i. e., with sun light illumination that include rendering shafts of light and caustics can be found in [IDN02] and [PP09].

For this thesis, a simulator using a physical model is required in order to be able to render synthetic underwater images. It is not based on the radiative transfer equation, but uses the Jaffe-McGlamery model, simplifications of which have already been often applied in the area of computer vision. Two other, similar methods that divided the water volume into voxels and/or use point spread functions for small angle forward scattering can be found in [PARN04] or [CS02]. An important advantage of the Jaffe-McGlamery model is that it can be easily combined with the refractive camera model that will be introduced in the next section. Jaffe's work [Jaf90], which is concerned with the development of underwater imaging systems, introduces a simulator for different camera-light configurations. In this thesis, the goal is to experiment with computer vision algorithms on underwater images and therefore ground truth data needs to be rendered that is compliant with the underlying physical principles. In order to achieve that, the Jaffe-McGlamery model is extended to incorporate several light sources and to render shadows. Instead of

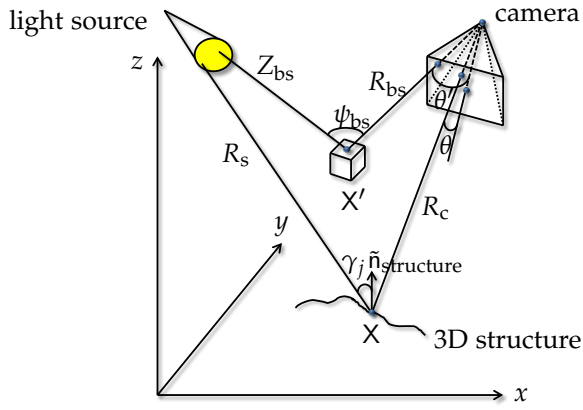

Figure 3.5. This images shows the rays for signal and backscatter computation, adapted from [Jaf90]. The backscatter portion is modeled by considering the amount of light being incident upon slabs of water (top part). Attenuation is modeled by explicitly computing the rays traveling from light source to object to camera.

rendering the xy-plane of the coordinate system, textured triangle meshes are rendered. In the model, light reaching the camera is considered to be the sum of three different components that are influenced by scattering and absorption and are described in the following paragraphs:

$$E_T(total) = E_d + E_{fs} + E_{bs,d}, \tag{3.1.11}$$

with E_d being the direct component, E_{fs} being forward scatter, and $E_{bs,d}$ being backscatter. In the following, the distance that light has been traveled below water is considered to be the distance between the object and the outer plane of the underwater camera housing.

Direct Light (Signal) The first component to be described is the signal and it is comprised of the light traveling from all light sources via reflection at the 3D structure to the camera. On its way, it is attenuated by photons being absorbed or scattered out of the beam, described by the attenuation

coefficient η. The extension of the Jaffe-McGlamery model to several light sources is achieved by modeling each light source $I_j, j \in \{1, ..., M\}$ as a point light source with a position in 3D space, its power in W and wavelength λ. For each pixel $\mathbf{x}$, the corresponding ray in water is determined (refer to the following section) and intersected with the structure in the 3D point X. Then, it is determined whether X is shadowed by other parts of the structure or by which light sources $I_j, j \in \{1, ..., M\}$ it is illuminated. According to Figure 3.5 and [Jaf90], the resulting irradiance being incident upon the structure is then:

$$E'_I(\mathsf{X}, \lambda) = \sum_{j=0}^{M} E_{I_j} \tag{3.1.12}$$

$$E_{I_j} = \begin{cases} 0 & \text{if X is shadowed} \\ I_j(\lambda) \cos(\gamma_j) \frac{e^{-\eta(\lambda) Rs_j}}{Rs_j^2} & \text{else} \end{cases} \quad [\mathrm{Wm}^{-2}],$$

where X are the coordinates on the structure in 3D space and Rs_j is the distance between the light source and X. γ_j is the angle between the ray from the light source I_j and X and the structure's normal. The irradiance upon the structure is complemented by small angle forward scattering, which adds flux to the beam. According to [Jaf90] and Section 3.1.1, it is modeled by convolution:

$$E_I(\mathsf{X}, \lambda) = E'_I(\mathsf{X}, \lambda) * g(\mathbf{x}|\overline{Rs}, G, \eta(\lambda), B) + E'_I(\mathsf{X}, \lambda) \tag{3.1.13}$$

$$g(\mathbf{x}|\overline{Rs}, G, \eta(\lambda), B) = \left[e^{-G\overline{Rs}} - e^{-\eta(\lambda)\overline{Rs}}\right] \mathcal{F}^{-1}\{e^{-B\overline{Rs}f}\},$$

where g is a filter mask with two empirical values G and B and $\mathcal{F}^{-1}$ stands for the inverse Fourier transform. $\overline{Rs} = \frac{1}{M}\sum_{j=1}^{M} Rs_j$ is the mean of all distances Rs_j, a simplification in order to efficiently incorporate several light sources. Using the linearity of convolution, the forward scattering for several light sources can thus be approximated with low computational overhead, especially considering that Schechner and Karpel determined in [SK04] that a low pass filter can be used. Hence, forward scattering computation is further simplified by approximation with a with a Gaussian filter with a variable filter mask size depending on the distance

the light traveled through water, extended by an empirical factor K similar to [TOA06], which weights the added forward scatter.

In [Jaf90], a reflectance map $M(\mathsf{X})$ is used to model reflection at the xy-plane of the coordinate system, which was rendered. Here, a textured 3D triangle mesh is rendered using a global reflectance factor M.

After reflection at the object point X the light traveling to the camera is again attenuated on the way. The camera itself is no ideal pinhole camera, but consists of a lens system that is modeled by vignetting, f-number, and lens transmittance, which changes the light measured by the sensor compared to the irradiance being incident upon the lens system. In [Jaf90] these effects are modeled similarly to the fundamental radiometric relation (A.2.1), explained in Section A.2:

$$E_d(\mathbf{x},\lambda) = \underbrace{\frac{E_I(\mathsf{X},\lambda)e^{-\eta(\lambda)R_c}M(\lambda)}{\pi}}_{\text{Signal}} \underbrace{\frac{\cos^4(\theta)T_\mathrm{l}(R_c-f_\mathrm{l})^2}{4f_\mathrm{n}R_c^2}}_{\text{Camera Transmittance}} \quad [\mathrm{Wm}^{-2}], \tag{3.1.14}$$

with R_c being the distance between X and the camera and θ being angle between the incoming ray and the camera's optical axis. The cosine term models vignetting (refer also to [Sze11]) and T_l is the lens transmittance. f_n is the camera's f-number (ratio between focal length and diameter of the entrance pupil) and f_l the camera's focal length in mm.

Forward Scatter (Signal) The light that travels from the structure to the camera is again complemented by small angle forward scattering, modeled by a distance dependent Gaussian filter as in Equation (3.1.13):

$$E_{fs}(\mathbf{x},\lambda) = E_d(\mathbf{x},\lambda) * g(\mathbf{x}|R_c,G,\eta(\lambda),B). \tag{3.1.15}$$

Note that small angle forward scattering does add some flux to the signal, but has been found to mainly blur the image depending on the distance the light has traveled. Thus, two of the three components in Equation (3.1.11) for the irradiance on a pixel $\mathbf{x}$ are accounted for.

Backscatter (Veiling Light) The final component in Equation (3.1.11) is backscatter and it is caused by multiple scattering events in the water body that eventually cause ambient or veiling light close to the light source or water surface (illuminated by sun light). The veiling light is then per chance scattered into beams traveling towards the camera and hence add to the irradiance. In McGlamery's work, backscatter is approximated by slicing the 3D space in front of a camera into N planes of thickness ΔZ_i parallel to the camera's image plane. Then, the light incidence by all light sources upon each plane is computed including small angle forward scattering. The total backscatter component is then the sum of the radiance from all backscatter planes. In order to compute that, the Jaffe-McGlamery model again needs to be extended to incorporate several light sources and the Volume Scattering Function (VSF) $\beta(\psi_j, \lambda)$ is applied to the irradiance being incident upon the slab, which can be done due to the linearity of convolution. This yields the irradiance on one backscatter plane:

$$E_s(\mathsf{X}') = E_{s,d}(\mathsf{X}') + E_{s,fs}(\mathsf{X}') \tag{3.1.16}$$

$$E_{s,d}(\mathsf{X}') = \sum_{j=0}^{M} E_{s,d_j}$$

$$E_{s,d_j} = \begin{cases} 0 & \text{if } \mathsf{X}' \text{ is shadowed} \\ L_j(\lambda) \frac{e^{-\eta(\lambda) Rbs_j}}{Rbs_j^2} \beta(\psi_j, \lambda) & \text{else} \end{cases}$$

$$E_{s,fs}(\mathsf{X}') = E_{s,d}(\mathsf{X}') * g(\mathsf{X}' \, \overline{Rbs}, G, \eta(\lambda), B), \tag{3.1.17}$$

where ψ_j is the angle between a line from the volume to the light source and a line from the volume to the camera (see Figure 3.5). This determines exactly the amount of light that is scattered out of the light source's beam into the light ray traveling towards the camera. The the sum of all backscatter planes is computed, complemented by the above described

model for camera transmittance:

$$E_{bs,d}(\mathbf{x}) = \sum_{i=1}^{N} \underbrace{\frac{e^{-\eta(\lambda)Z_{bs_i}} E_s(\mathrm{X}', i)\Delta Z_i}{\cos(\theta)(\mathrm{X}')}}_{\text{Backscatter}} \underbrace{\frac{\cos^4(\theta) T_l (Z_{bs_i} - F_l)^2}{4 f_n Z_{bs_i}^2}}_{\text{Camera Transmittance}} \quad [\mathrm{Wm}^{-2}], \tag{3.1.18}$$

with ΔZ_i being the thickness of the backscatter volume slice, Z_{bs_i} being the distance between the center point of the slice i and the camera, and i being the index of the backscatter plane. (3.1.18) is split into an irradiance part being incident upon the camera and an camera transmission part, identical to the one in (3.1.14).

3.1.3 Methods for Color Correction

The last two sections introduced the basic physical principles involved in underwater light propagation and a physical model that can be used to simulate the effect. However, there is another interesting issue related to underwater light propagation: the question if and to what extent the colors in images can be corrected. In the computer vision literature, many methods can be found and are classified into two groups by Schettini and Corchs in [SC10]. In the first group are methods that are based on a simplification of the Jaffe-McGlamery model, thus a model based on the underlying physical principles, which actually restores image colors. In the second group are all methods that use differing sets of image filters in order to enhance image colors, e. g., stretch the contrast on the different color channels individually. In general, methods in the first group perform better, however, it is often difficult to calibrate the necessary parameters of the physical model because they need to be measured in the local water body. Methods in the second group often consist of a set of filters that were chosen heuristically, and hence it is sometimes unclear why a particular filter performs well. In contrast to the parameters for the restoration methods, the filter parameters cannot be measured, thus, need to be determined by trial and error for each new image. Methods for both categories are summarized in Table 3.2. The main equation derived by

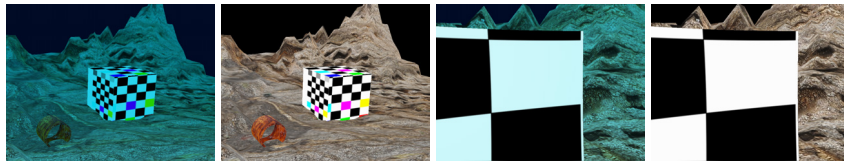

Figure 3.6. Simulation of viewing a colored checkerboard underwater from two different distances and its color corrections. Note that at larger distances (approx. 5000 mm in the first image) the blue hue on the white cube is very strong, while at close distances (approx. 600 mm in the third image), the blue hue is only very slight.

simplifying the Jaffe-McGlamery model can be found in [SK04, SK05]:

$$E_{\mathrm{cam}_\lambda} = \underbrace{E_{\mathrm{obj}_\lambda} \mathrm{e}^{-\kappa\eta(\lambda)}}_{\text{Signal}} + \underbrace{B_{\infty_\lambda}(1 - \mathrm{e}^{-\kappa\eta(\lambda)})}_{\text{Backscatter}}, \tag{3.1.19}$$

where E_{obj_λ} is the irradiance, which is attenuated exponentially with distance κ. B_{∞_λ} is the veiling light present in the water due to multiple scattering events. E_{cam_λ} is the irradiance being incident upon the image sensor. Note that in the course of simplification the effects on irradiance while traveling from the light source to the object and the effects of the lens system where omitted. (3.1.19) can be applied if κ is known for each pixel. This is a geometric entity that must be estimated for example during dense depth estimation (Chapter 5). Figure 3.6 shows simulation results for the underwater color correction using Equation (3.1.19). Depicted are two pairs of underwater and corrected images at different distances. The checkerboard cube in the first pair is at a distance of approximately 5000 mm, while the cube in the second image pair is at a distance of approximately 600 mm.

Table 3.2. Article overview for image color correction categorized into methods based on the underlying physical principles (restoration) and others using heuristically determined sets of filters (enhancement). Note that the methods on enhancement are only exemplary because so many works exist that listing all of them would be out of the scope of this work.

Authors	Category	Method
Image Color Correction		
Schechner et al. [SK04, SK05]	restoration	capturing two images through a polarization filter allows to use an equation similar to Equation (3.1.19) for image color restoration; method works with sunlight; polarization can remove backscatter, thus enhancing contrast
Treibitz, Schechner, et al. [TS06, TS08]	restoration	similar to above, but adapted to deep sea scenarios, where the scene is lit by a lamp that has a polarization filter.
Trucco, Olmos-Antillon [TOA06]	restoration	assumes small angle forward scattering to be main source of degradation; from Jaffe-McGlamery model a filter is derived based on the attenuation coefficient η.
Hou et al. [HWGF07, HGWA08]	restoration	de-blurring is achieved by deriving a point spread function based on the assumption that small-angle-forward scattering is the main source of image degradation.
Yamashita et al. [YFK07]	restoration	parametrize the attenuation part of Equation (3.1.19) by using two or more images with different distances κ_i and known camera translation; small-scale lab experiments
Queiroz et al. and Nasimento et al. [QNCBC04, NCdB09]	restoration	apply model (3.1.19) during a stereo method to enhance matching
Iqbal et al. [ISOT07]	enhancement	simple method with good results; contrast stretching on each color channel, then conversion to HSI, where saturation and intensity are stretched
Bazeille et al. [BQJM06]	enhancement	use a lot of different filters, homomorphic filtering, wavelet denoising, anisotropic filtering, suppressing dominant color by equalization, etc.

3.2 Geometric Effects

After traveling through the water, the light rays enter the camera's underwater housing, which usually consists of a piece of glass, either formed as a flat port or a dome port (Figure 3.8). Therefore, the light is first traveling through water, then glass, and finally air before it reaches the actual camera. Due to the different media being traversed, refraction of the rays occurs, thus underwater images are affected geometrically in addition to the above described effects on color. In this section, the underlying physical principles will be explained along with a state-of-the-art overview of related work found in the literature, followed by a description of refractive ray computation utilized in this thesis.

3.2.1 Refraction at Underwater Housings

The physical principle of refraction [Hec05] is defined to be a change of direction of a light ray compared to its former path. It is dependent on the optical density of the media involved and causes a change in phase velocity v:

$$n = \frac{c}{v}, \tag{3.2.1}$$

with n being the index of refraction and c being the speed of light. The effect can be described intuitively by Fermat's principle: the light ray travels the path that takes the least time to traverse. Figure 3.7 can be used to understand Fermat's principle: a ray travels from S, is refracted at O, and then travels to P. The time required to travel this distance can be calculated by [TSS08]:

$$t = \frac{\sqrt{(Z-d)^2 + (R_2 - R_{1,2})^2}}{v_1} + \frac{\sqrt{d^2 + R_{1,2}^2}}{v_2}, \tag{3.2.2}$$

where v_1 and v_2 denote the phase velocity in the corresponding medium. This is minimized by computing the derivative [TSS08]:

$$\frac{\partial t}{\partial R_{1,2}} = \frac{-(R_2 - R_{1,2})}{v_1\sqrt{(Z-d)^2 + (R_2 - R_{1,2})^2}} + \frac{R_{1,2}}{v_2\sqrt{d^2 + R_{1,2}^2}} = 0, \tag{3.2.3}$$

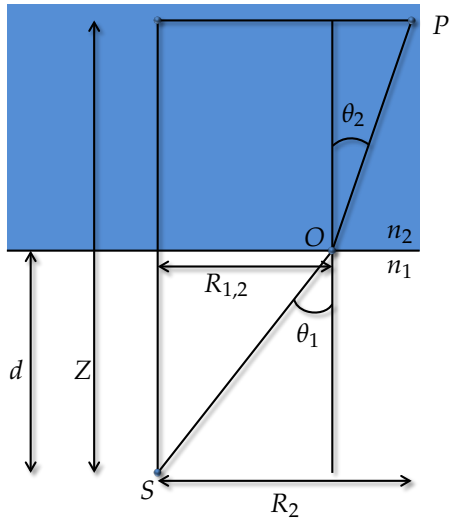

Figure 3.7. Fermat's principle based on the ray from S to P being refracted at O. Adapted from [TSS08].

and expressed by:

$$\frac{\sin\theta_1}{v_1} = \frac{\sin\theta_2}{v_2}. \tag{3.2.4}$$

With c being the speed of light in vacuum and $n_1 = c/v_1$ and $n_2 = c/v_2$, Snell's law follows:

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{n_2}{n_1}. \tag{3.2.5}$$

n_1 and n_2 are the indices of refraction and properties of the media denoted by 1 and 2. Note that the index of refraction for a vacuum is set to 1 and the indices of refraction for all other media are calibrated relative to the index for vacuum. The index of refraction for air is usually set to $n_a = 1$, which will be sufficiently accurate for this thesis as well. In water, the index of refraction is dependent on pressure, temperature, salinity, and the light's wavelength. However, according to [Mob94], the change in the relevant range of ocean water is about 3% (see Table 3.3), so in

Table 3.3. Indices of refraction for air, different kinds of water, and glass as in [Hec05] p. 163 and [Mob94] p. 85.

Medium	Index of Refraction
air ($\lambda = 589\,\text{nm}$)	1.0003
pure water ($\lambda = 700\,\text{nm}$, $30°C$, $p = 1.01\text{e}10^5\,\text{Pa}$)	1.329
pure water ($\lambda = 700\,\text{nm}$, $30°C$, $p = 1.08\text{e}10^8\,\text{Pa}$)	1.343
sea water ($\lambda = 700\,\text{nm}$, $30°C$, $p = 1.01\text{e}10^5\,\text{Pa}$)	1.335
sea water ($\lambda = 400\,\text{nm}$, $30°C$, $p = 1.08\text{e}10^8\,\text{Pa}$)	1.363
quartz glass ($\lambda = 589\,\text{nm}$)	1.4584
acrylic glass (Plexiglas, $\lambda = 589\,\text{nm}$)	1.51
crown glass ($\lambda = 589\,\text{nm}$)	1.52
light flint glass ($\lambda = 589\,\text{nm}$)	1.58
dense flint glass ($\lambda = 589\,\text{nm}$)	1.66
Lanthan flint glass ($\lambda = 589\,\text{nm}$)	1.80

the remainder of this thesis, it will be set to $n_w = 1.333$. The index of refraction for glass on the other hand shows greater variance [Hec05] (see Table 3.3), depending on the exact material, but it is usually known which kind of glass is used, and hence can be considered explicitly.

Since refraction depends on the angle between the entering ray and the normal of the interface, almost all rays are refracted in case of flat port underwater housings (Figure 3.8 on the left). Only the light rays with an incidence angle of zero to the interface normal can pass through the interface without change of direction. When following the rays in water without refraction at the interfaces (dashed lines), it can be observed that the single-view-point camera model is invalid and flat port underwater cameras can be classified as nSVP cameras (compare to Section 2.2.3). In case of ideal dome ports (Figure 3.8, right), the light rays are not refracted because all rays are exactly parallel to the interface normal at the intersection point. Ideal dome ports need to be build such that the camera's center of projection coincides exactly with the dome port sphere's center. Especially, since the center of projection is difficult to determine in practice [AA02] and can even lie in front of the physical camera and its lens system (Section 2.2.2), it is difficult to build an underwater camera with an ideal dome port. Consequently, for most real dome port cameras, the single-view-point model is invalid as well.

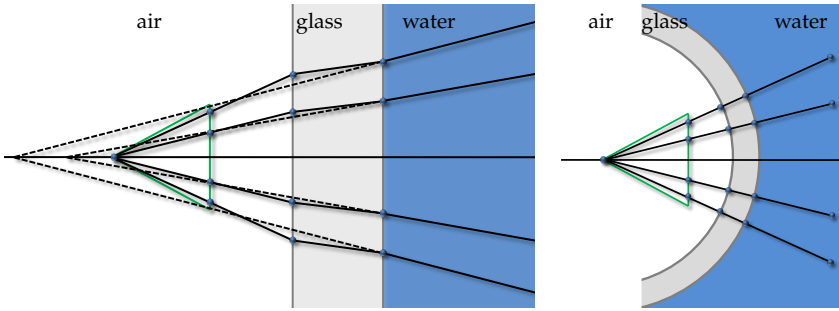

Figure 3.8. Left: refraction at flat glass interface. Right: straight rays entering the underwater housing through a dome port with a perfect fit, i. e., the center of projection coincides with the sphere's center.

Figure 3.9. Left: the caustic for a flat port camera with imperfect sensor-interface alignment. Right: caustic in the dome port case with imperfect alignment of camera center and sphere center.

As described in Section 2.2.3, nSVP cameras are more general then single-view-point cameras and can be characterized by their caustics. Exemplary caustic shapes for a flat port and a dome port underwater camera are shown in Figure 3.9. [TSS08] analyzed caustics for flat port underwater cameras without interface tilt and zero glass thickness. In the flat port case in Figure 3.9 on the left, the caustic is asymmetric due to a slight inclination of the interface normal with respect to the camera's

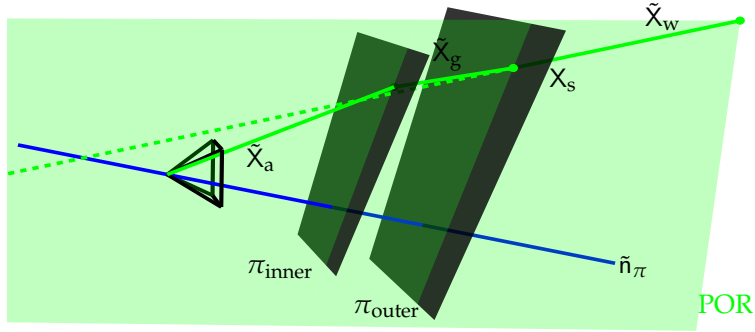

Figure 3.10. POR is the plane of refraction, the common plane of all ray segments $\tilde{X}_a$, $\tilde{X}_g$, and $\tilde{X}_w$. Additionally, all rays in water intersect the line formed by the interface normal $\tilde{n}_\pi$ and the center of projection.

optical axis.

The second part of Snell's law of refraction states that both parts of the ray before and after refraction and the interface normal all lie in one common plane. Hence all three ray segments, in water, glass, and air, lie in one common plane called the Plane of Refraction (POR) [ARTC12]. According to [ARTC12], in case of flat port cameras, this plane intersects the axis formed by the interface's normal and the camera center, thus all rays (dashed line in Figure 3.10) intersect this axis (blue line). Consequently, the flat port refractive camera is a special nSVP camera, an axial camera, i. e., all rays intersect a common axis and the refractive camera can be classified as an axial camera (refer to Section 2.2.3).

In the literature, often the perspective camera model is used on underwater images despite the fact that it is invalid and has a systematic model error. A more general camera model (completely ray-based or axial) can be used to eliminate this systematic error. However, underwater housings can also be modeled explicitly by very few parameters allowing to accurately compute the light paths through the housing. All three possibilities will now be examined more closely. Note that a similar literature overview and camera model discussion has already been published in [SK12].

3.2.2 Perspective Camera Model

When using the perspective camera model on underwater images, a simple experiment reveals how the refractive effect can be approximated to some extent: the camera needs to be calibrated above and below water. A comparison of the results reveals that the camera parameters absorb part of the model error. Two works exist that examine this effect more closely. Fryer and Fraser [FF86] calibrate with the pinhole model extended by three parameters for radial distortion and two for tangential distortion. They conclude that refraction is compensated for by multiplying the focal length measured in air f_a by the index of refraction for water:

$$1.333 f_a = f_w. \tag{3.2.6}$$

In addition, radial distortion added to the points in the image needs to be changed by: $\delta r = \left(\frac{\cos\theta_w}{\cos\theta_a} - 1\right) r$, with r being the radial distortion in air, θ_w being the angle between optical axis and water ray, and θ_a being the angle between optical axis and ray in air. Note that the angles required for computing the distortion are usually unknown. Lavest et al. [LRL00] come to the same conclusion concerning focal length. Concerning radial distortion, they conclude:

$$1.333(r_{d_a} - r_{rad_w}) = r_{d_w} - r_{rad_w}. \tag{3.2.7}$$

In experiments using two different cameras, they found (3.2.7) to be a good approximation. Figure 3.11 helps to understand why the perspective model is only an approximation of the refractive camera by regarding the situation in cylinder coordinates. (R_w^1, Z_w^1) is a 3D point to be projected onto the point r on the image plane, when considering refraction explicitly. Since (R_w^2, Z_w^2) is on the same ray segment in water, the point is projected onto the same position on the image plane, when using the refractive model. However, when using the perspective model, (R_w^1, Z_w^1) is imaged onto r_{uncomp}^1. The change in focal length is shown by the virtual image plane, which is moved away from the center of projection. Here, the projections of (R_w^1, Z_w^1) and (R_w^2, Z_w^2) are r_{per}^1 and r_{per}^2. Radial distortion can then be used to compensate the remaining error for r_{per}^1, yielding

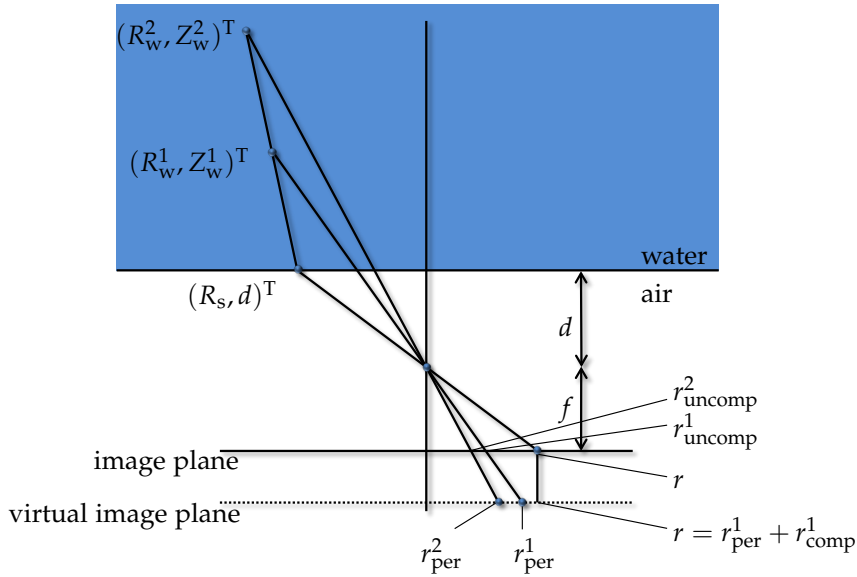

Figure 3.11. Approximation of the underwater camera by the perspective model. A virtual image plane is used in combination with larger radial distortion to project the point onto the same radial coordinate. Despite that, the two 3D points lying on the same ray in water are projected onto the same position using the underwater model, but onto different positions using the perspective model.

the correct radial distance $r = r^1_{per} + r^1_{comp}$. However, in order to get the correct radial distance for r^2_{per}, a different radial distortion compensation term is required. Unfortunately, this term is depending on the distance of the original 3D point from the camera, a characteristic that is not part of the common pinhole camera model. Hence, the substitution of the refractive camera by the perspective camera is only valid at the calibration distance.

Despite the systematic model error introduced by using the perspective camera model in underwater scenarios, a lot of works can be found in the literature, where methods, which were designed for perspective cameras, are used on underwater images. Table 3.4 summarizes works

Table 3.4. Literature on calibrating the perspective camera model on underwater images.

Authors	Application	Method
Perspective Model		
Freyer et al. [FF86]	calibration	calibrate in air and find perspective water calibration by adapting focal length ($1.333 f_a = f_w$) and distortion $\delta r = \left(\frac{cos\theta_w}{cos\theta_a} - 1\right) r$, with r being the radial distortion in air, θ_w being the angle between optical axis and water ray, and θ_a being the angle between optical axis and air ray
Lavest et al. [LRL00]	calibration	calibrate in air and find perspective water calibration by adapting focal length ($1.333 f_a = f_w$) and distortion ($1.333 u_a + du_a) = u_w + du_w$, with u_a and u_w being the distorted coordinates and du_a and du_w being the distortion corrections
Bryant et al. [BWAZ00]	calibration	find checkerboard corners robustly in turbid environments; calibrate based on underwater images; using one coefficient for radial lens distortion
Pessel et al. [POA03b, POA03a, Pes03a]	calibration	checkerboard-free self calibration approach using predefined trajectory; no distortion modeled because lens system eliminated distortion effects by 99%; calibrate on-site to adapt calibration to changing index of refraction of water

concerned with calibration. Examples for applications include stereo-based distance measurements and mosaicing (Table 3.5), or Structure-from-Motion (Tables 3.6 and 3.7) Especially in case of SfM, these methods rely on navigation data (e. g., from ROVs (Remotely Operated Vehicle)) and/or extensive bundle adjustment [TMHF00] to at least partly counter act the error. Even though, the error accumulated over time is considerable often leading to inaccurate or even useless reconstruction results.

Table 3.5. Literature describing stereo measurement and mosaicing based on underwater images using the perspective camera model.

Authors	Application	Method
Perspective Model		
Harvey et al. [HS98]	stereo measurement	usage of stereo rig to measure underwater structures; examination of calibration robustness in different water bodies; 3D calibration frame
Costa et al. [CLC+06]	stereo measurement	automatically measure fish size using a stereo rig; automatic contour detection and interest point triangulation; initial calibration without distortion, removal of inconsistencies by training neural network; 5% measuring accuracy
Gracias et al. [GSV00, GvdZBSV03]	mosaicing	mosaic computation used for navigation afterwards; self calibration with rotating camera on pan-tilt unit, sequential mosaic building followed by global optimization; geo-referenced
Garcia, Carreras et al. [GBCA01, CRGN03]	mosaicing	mosaic computation used for navigation afterwards; one parameter for radial distortion; second work using robot in pool with coded pattern on the ground for estimating the accuracy of other on-board navigation devices
Xu, Negahdaripour et al. [NXKA98, XN01]	mosaicing	first: simultaneous mosaicing, navigation, and station keeping; second: statistical combination of image-based registration data and other navigation data sources applied to mosaic computation
Eustice et al. [ESH00]	mosaicing	compares different methods for mosaicing under consideration of movement with growing complexity (from translation to full projective transformations) in underwater environments
Trucco et al. [TDO+00]	mosaicing	mosaicing approach via tracked features and homography estimation; registered images are warped into common image

Table 3.6. Literature overview on 3D reconstruction based on underwater images using the perspective camera model.

Authors	Application	Method
Perspective Model		
Hogue et al. [HGZJ06, HGJ07]	reconstruction	a bumblebee stereo camera and IMU are combined in one underwater housing and used to reconstruct and register 3D structure; reconstruction shows a lot of drift if IMU is not used and authors presume erroneous camera model to cause part of it
Jasiobedzki et al. [JSBJ08, JDL12]	reconstruction	real-time reconstruction using stereo images, registered using ICP; resulting model is bent, authors plan to incorporate refraction to eliminate the error; the second work contains an interesting extension for photosynthetic life detection using macroscopic fluorescence imaging
Sedlazeck et al. [SKK09]	reconstruction	classical, sequential SfM with adaptations to underwater environment; calibration below water, two coefficients for radial distortion, dense depth maps are used for model computation; additional color correction; absolute scale from navigation data
Pizarro et al. [PES03b, PES04]	reconstruction	calibration below water including distortion; reconstructions based on two or three images are registered against each other by a graph based algorithm; Delaunay triangulation; usage of navigation data
Johnson et al. [JRPWM10]	reconstruction	sparse sets of 3D points are meshed using a Delaunay triangulation and registered via SLAM utilizing navigation data; additional loop closing and color correction; can process thousands of images
Brandou et al. [BAP+07]	reconstruction	stereo rig is moved on predefined trajectory by a ROV arm; model is computed using dense depth maps; camera is calibrated on-site by deploying checkerboard on the sea floor
Negahdaripour et al. [NSP07, NSP09]	reconstruction	combination of optical and acoustic systems in one rig; calibration and reconstruction theory in presence of euclidean and spherical coordinates
Bingham et al. [BFS+10]	reconstruction	overview paper using AUV for shipwreck documentation with optical and acoustic methods; navigation data is utilized

Table 3.7. Literature overview on 3D reconstruction based on underwater images using the perspective camera model.

Authors	Application	Method
Perspective Model		
Beall et al. [BLID10]	reconstruction	compute 3D model of coral reef with a lot of (moving) plants; use SAM (Smoothing and Mapping) to determine smooth camera path
Kang et al. [KWY12a]	reconstruction	small-scale reconstructions based on underwater images with comparison to ground truth; conclusion is that perspective camera model can be used to reconstruct perspectively
Inglis et al. [ISVR12]	reconstruction	combination of stereo camera rig and laser sheet to obtain large area bathymetric maps; ROV movement is computed using SLAM; can process over 100,000 images
Queiroz-Neto, Nascimento et al. [QNCBC04, NCdB09]	underwater stereo	color correction routine is combined with stereo to match more robustly; no consideration of refraction

3.2.3 Ray-Based Axial and Generic Camera Models

Instead of using the perspective camera model, one can also apply more general camera models. They can account for refraction explicitly and do not require the cameras to have a single view point (refer to Section 2.2.3).

Grossberg et al. [GN05] introduced the raxel camera model (Section 2.2.3), but the proposed calibration method requires an active display and is therefore impractical in underwater calibration. A different work by Narasimhan et al. [NNSK05] researches light sheet reconstruction as an application of the described raxel model for small scale underwater images in laboratory settings. A camera is put in front of a water tank, and calibrated by placing two planes into the tank vertically with respect to the optical axis and therefore gaining two points in space for each ray.

Sturm et al. [SR04, SRL06] describe models for generic camera calibration and SfM, which in theory are applicable to the underwater case. In [RLS06], a generic SfM framework is proposed, where a generic camera is approximated by clustering rays such that each cluster can be approximated by a perspective camera, thus making bundle adjustment optimization feasible. [CS09] is specialized to the case of a refractive plane in an underwater scenario. The derivation only works for one refractive interface (thin glass) and has not yet been implemented. A complete system for general camera systems, which has also been implemented was introduced by Mouragnon et al. [MLD+07, MLD+09]. They use a catadioptric camera or a rig of perspective cameras, which are assumed to be calibrated such that for each pixel the ray in space is known. By introducing an error function that is based on angles between rays, they propose a SfM system that can run in real time.

Another possibility to deal with refraction by approximating ray-based cameras is described in [Wol07]. Here, the camera is viewed as a nSVP camera having a caustic instead of the single view point. Instead of modeling the refractive effect physically or using a generic ray-based camera, the camera is approximated by several perspective cameras for the different areas of the image. The number of virtual perspective cameras determines the accuracy of this system.

In summary, it can be said, that using a more generic camera model than the pinhole model with distortion allows to deal with refractive

Table 3.8. Ray-based methods that can be applied to underwater images.

Authors	Application	Method
Ray-based Models		
Grossberg, Narasimhan et al. [GN05, NN05, NNSK05]	calibration, reconstruction	cameras are defined via their caustics; calibration routine using an active display; second work specializes on underwater case with light sheet based reconstruction in small tank environments
Sturm et al. [SR04, SRL06]	calibration, reconstruction	development of theory for generic cameras described only by their rays (assumption is that neighboring rays are close to each other); calibration by taking several checkerboard images
Ramalingam et al. [RLS06]	reconstruction	generic SfM based on generic camera calibration above, propose to cluster generic camera rays to approximate perspective cameras
Mouragnon et al. [MLD+07, MLD+09]	reconstruction	based on an angular error between two rays, real-time 3D reconstruction is introduced for ray-based cameras
Wolff [Wol07]	seafloor reconstruction	reconstruction of sea floor in simulator (small tank); ray-based, generic camera is approximated by several perspective cameras suitable for different image regions

effects. However, using independent 3D origins and directions for each ray leads to a high degree of freedom, making the robust calibration of generic camera models difficult, especially in open water. The following section shows that far less parameters need to be determined if refraction is modeled explicitly.

3.2.4 Physical Models for Refraction

In the literature, several contributions exist that deal with refraction explicitly and propose corresponding calibration methods. Often, assumptions for simplification are made, e. g., no normal inclination and a very thin, flat glass port are assumed by Treibitz et al. [TSS08]. They argue that the glass port in use has a glass thickness of only about 5 mm, causing a maximum ray shift due to the distance traveled through glass of only about 0.28 mm. Telem et al. describe in [TF06, TF10] a system for calibrating a camera also with thin glass and parallelism of glass normal and optical axis, but in

their model they determine a point on the optical axis for each 2D point that serves as a correct center of projection and relate the measured 2D image coordinates to image coordinates eligible for perspective projection. This approach is extended in a second article [TF10] to account for normal inclinations. However, in this case the error due to glass thickness is absorbed by the interface distance, therefore, this approach is not an exact model of the refractive camera. [Kwo99, KC06] describe an often cited method, where the Direct Linear Transform (DLT) [HZ04] for pose estimation is combined with refraction. Parallelism of interface and image sensor is assumed and achieved by manually rotating the camera in front of the glass port. The distance between camera and glass port is also measured manually, and the authors have not included an automatic calibration routine into their algorithm. Li et al. [LLZ+97, LTZ96] (see also [McG04]) described a photogrammetric approach for calibrating underwater stereo cameras. They did not make any assumptions, except for the indices of refraction, which need to be known. In [LTZ96], an additional reduced central projection allows to project points from 3D through a refractive interface onto the image plane with an iterative method that solves for the required unknowns on the interfaces. Kunz et al. [KS08] consider dome and flat ports in their work. They describe a model for the computation of back-projection and a corresponding calibration routine. However, the calibration routine is not implemented and therefore not tested. In addition, they show by some simulations that the error introduced by using the perspective camera model is considerable.

A recent work by Agrawal et al. [ARTC12] has already been mentioned above. They showed that the refractive camera is actually an axial camera and proposed a calibration routine based on that insight. In addition, they showed that the projection of 3D points into the camera can be computed by solving a 12^{th} degree polynomial instead of the non-linear optimization required by [KS08], thus, making the projection of 3D points much more efficient.

The computation of SfM using the refractive camera model has only recently been considered in more detail. Chang and Chen [CC11] assumed to have a camera looking through the water surface onto a submerged scene with known vertical direction of the camera, i. e., the camera's yaw and pitch with respect to the water surface need to be known. The

method then only needs to compute the heading of the cameras instead of complete pose estimation. Kang et al. [KWY12b] proposed a system for computing refractive 3D reconstructions below water for two images based on outlier-free correspondences, which need to be selected manually. Using the reprojection error, they optimize their two-view scene with bundle adjustment and in order to compute a dense model, they run a modified version of PMVS (Patch-based Multi-view Stereo) [FP10] that can compute refraction explicitly. Gedge et al. [GGY11] propose a method for refractive dense stereo, where the epipolar curves caused by refraction are approximated. However, in order to compute dense stereo, 3D points need to be projected into images. Due to the assumed thin glass, a fourth-degree polynomial needs to be solved for each projection.

Another approach to using physical models is found in the works of Maas, [Maa92, Maa95] and a follow-up work by Putze [Put08b, Put08a]. The goal of both methods is optical fluid flow analysis in fairly small laboratory tanks, where the fluid has been marked with a set of particles. In the model, the actual 3D points in space are substituted by corresponding virtual 3D points that fit the perspective back projection. The computation of these points is based on an iteration with known interface parameters and indices of refraction. In order to calibrate the system, a calibration pattern below water at known distances is used and optimized by a bundle adjustment routine. The method has been found to perform well if the indices of refraction, especially for the glass are known. A statistical correlation analysis shows high correlation between focal length and distance between camera center and glass interface for all three calibrated cameras. The works of Maas also contain an introduction to epipolar geometry [HZ04] in case of refractive imaging, where the epipolar lines are bent into curves. If the ray in water from one camera is known, several points on this ray are projected into the second image defining a linear approximation of the epipolar curve. This is for example used in [FW00] examining surface reconstruction.

In addition, there exist some more unusual applications also considering refraction explicitly. In contrast to the approaches described above, where the indices of refraction are assumed to be known, they can be calibrated in confined laboratory scenarios. See [MK05, YHKK03, YFK08, KYK09, FCS05] for more detailed information.

A more complete overview and classification of existing methods can be found in Tables 3.4 to 3.10. In summary it can be said that no refractive SfM system with arbitrary camera movement and thick, possibly inclined glass exists that can handle more than two cameras and does not require to project points, i. e., to solve a 12^{th} degree polynomial. Chapter 5 will propose such a system.

3.2.5 Refractive Camera Model with Thick, Inclined Glass

The refractive model used in this thesis assumes potentially thick glass ports, which are required for large water depths. In addition, the interface may be tilted with respect to the imaging sensor. By parametrizing the glass interface, refraction of rays can be modeled explicitly, thus eliminating the need to calibrate the ray for each pixel separately or to approximate refraction using the pinhole camera model. The following parameters are used in the flat port and dome port case:

1. Flat Port Underwater Housing (compare to Figure 3.10):
   - ▷ indices of refraction for air, glass, and water (n_a, n_g, and n_w)
   - ▷ interface distance and interface thickness (d, d_g)
   - ▷ interface normal with respect to the optical axis ($\tilde{\mathsf{n}}$)
2. Dome Port Underwater Housing:
   - ▷ indices of refraction for air, glass, and water (n_a, n_g, and n_w)
   - ▷ inner radius of sphere d, glass thickness d_g
   - ▷ sphere center with respect to camera center (X_dome)

Using these parameters, refraction for each ray through the housing can be computed explicitly. In the following, first the back-projection of a pixel onto a ray will be described, then the projection of a 3D point to a 2D pixel is derived. Both descriptions start with the flat port case (based on [ARTC12]) and are extended by the necessary deviations for dome ports (based on [KS08]).

Note that the interface normal is determined by a normalized three-vector $\tilde{\mathsf{n}} = (n_1, n_2, n_3)^\mathrm{T}$, but can also be described by angles $\theta_1 = \tan^{-1}(n_2/n_1)$ and $\theta_2 = \cos^{-1}(n_3)$ (Figure 3.12).

Table 3.9. Literature overview of methods for calibrating refractive camera models.

Authors	Application	Method
Refractive Underwater Models		
Treibitz et al. [TSS08]	calibration	physical model for underwater imaging is developed assuming thin glass and interface-sensor parallelism; analytical derivation of projection using cylinder coordinates; includes calibration routine; link to caustics
Telem et al. [TF06, TF10]	calibration	each point is mapped to a point eligible for perspective projection by moving the point in the image and computing the correct intersection with the optical axis; second work extends this mapping to the case of tilted interfaces; only correct for thin glass
Kwon et al. [Kwo99, KC06]	calibration, measurement	refraction is modeled in combination with the DLT for pose estimation; assumed thin glass; no rotation between glass and camera sensor
Kunz et al. [KS08]	calibration	hemispherical and flat ports are modeled and synthetic data is used to demonstrate inaccuracies using the perspective model; calibration routine is described, but not implemented; general case for non-parallel interfaces and thick glass
Li et al. [LTZ96, LLZ+97]	calibration	development of complete physical model and its calibration; stereo rig is used to triangulate the derived rays in water; indices of refraction are assumed to be known
Sedlazeck et al. [SK11a]	calibration	calibration of underwater housings of stereo rig; no checkerboard required, runs with bundle adjustment based on features; virtual camera error
Agrawal et al. [ARTC12]	calibration	derive that flat port refractive cameras are axial cameras, i. e., all rays intersect in common axis; calibration routine for interface distance and tilt based on that insight; derivation of projection with 12^{th} degree polynomial by computing polynomial coefficients on POR
Jordt-Sedlazeck et al. [JSK12]	calibration	uses [ARTC12] for initialization, then a non-linear optimization with an Analysis-by-Synthesis approach is proposed that is independent of errors in corner detection
Maas [Maa92, Maa95]	fluid flow measurement	a complete physical model for a rig is derived assuming interface parallelism; calibration; 3D points are iteratively moved to positions eligible for perspective projection; rig is used to reconstruct fluid flow marked by suspended particles; indices of refraction not calibrated

Table 3.10. Literature overview summarizing methods for different 3D applications with a refractive camera model.

Authors	Application	Method
Refractive Underwater Models		
Putze [Put08b, Put08a]	calibration, fluid flow measurement	follow-up work to the one above increasing robustness
Förstner et al. [FW00]	reconstruction	[Maa92] and its specialized epipolar geometry are used for surface reconstruction
Morris et al. [MK05]	wave surface reconstruction	a calibrated stereo rig views the bottom of a water tank on which a checkerboard pattern is placed; refraction is used to determine the wave's normals on the liquid's surface
Yamashita, Kawai et al. [YHKK03, YFK08, KYK09]	measurements	in small water tanks within a lab, a stereo system, a laser beam, or active patterns are used to gain reconstructions of objects completely or half emerged in the water
Ferreira et al. [FCS05]	underwater stereo	the underwater model is linearized to compensate for the majority of the errors induced by using the perspective model for stereo
Chang et al. [CC11]	reconstruction	3D reconstruction of a scene viewed by a camera through the water surface; camera's yaw and pitch need to be known
Gedge et al. [GGY11]	dense stereo	epipolar curves caused by refraction are approximated and used for stereo matching; requires to project points into stereo image pairs
Kang et al. [KWY12b]	reconstruction	a system for 3D reconstruction computation from two views; relative pose is computed on outlier-free correspondences, then 3D points are triangulated; perspective BA; sparse cloud is filled using PMVS [FP10] adapted to the refractive camera model
Chari et al. [CS09]	reconstruction	theory, but no implementation of refractive SfM (thin glass)
Jordt-Sedlazeck et al. [JSK13]	reconstruction	refractive SfM for long image sequences; efficient bundle adjustment using virtual camera error function
Jordt-Sedlazeck et al. [JSJK13]	dense stereo	refractive plane sweep; can apply color correction; comparison between perspective and refractive plane sweep

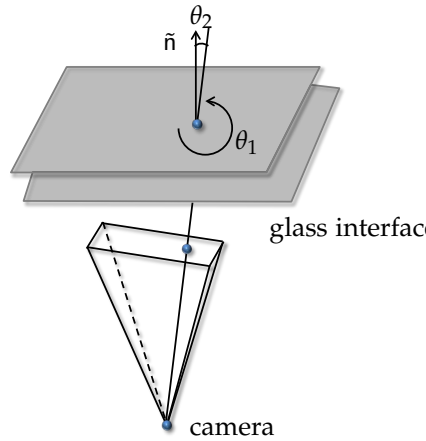

Figure 3.12. Angles describing interface normal. θ_1 describes the rotation around the interface normal ñ and θ_2 is the angle between interface normal and the camera's optical axis.

Back-Projection from 2D to 3D

Let $\mathbf{x}$ denote a homogeneous 2D point that describes a pixel position. Then, the first segment of the ray is determined by (refer to Section 2.2.1):

$$\tilde{\mathsf{X}}_\mathrm{a} = \frac{\mathbf{K}^{-1}\mathbf{x}}{||\mathbf{K}^{-1}\mathbf{x}||_2}, \tag{3.2.8}$$

with additional consideration of lens distortion, where $\tilde{\mathsf{X}}_\mathrm{a}$ is normalized to $\|\tilde{\mathsf{X}}_\mathrm{a}\| = 1$. The next ray segment $\tilde{\mathsf{X}}_\mathrm{g}$ is determined using Snell's law, the interface normal ñ, the interface distance d, and the indices of refraction for air n_a and glass n_g. Considering Figure 3.13 the following equation is derived in [Gla94]. The ray $\tilde{\mathsf{X}}_\mathrm{g}$ can be split into two components, with $\mathsf{X}_{\mathrm{g}\parallel}$ being parallel to the interface normal, and $\mathsf{X}_{\mathrm{g}\perp}$ being perpendicular to the interface normal. From that, several properties between the angle θ_g and the ray can be determined:

$$\begin{aligned} \tilde{\mathsf{X}}_\mathrm{g} &= \mathsf{X}_{\mathrm{g}\parallel} + \mathsf{X}_{\mathrm{g}\perp} \\ \mathsf{X}_{\mathrm{g}\parallel} &= \tilde{\mathsf{X}}_\mathrm{g} - \mathsf{X}_{\mathrm{g}\perp} \end{aligned} \tag{3.2.9}$$

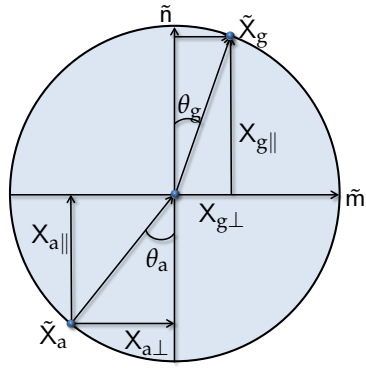

Figure 3.13. Refraction of rays using Snell's law. $\tilde{m}$ denotes the interface between the two media, here air (bottom) and glass (top). $\tilde{X}_a$ is the ray in air, $\tilde{X}_g$ is the ray in glass. Refraction is determined by the angles θ_a and θ_g and Snell's law. Adapted from [Gla94].

$$
\begin{aligned}
0 &= X_{g\|}^T X_{g\perp} \\
\|\tilde{X}_g\|^2 &= \|X_{g\|}\|^2 + \|X_{g\perp}\|^2 \\
X_{a\perp} &= \tilde{X}_a - \cos\theta_a \tilde{n}
\end{aligned}
$$

Note that Snell's law can be applied to the angles θ_a and θ_g:

$$
n_a \sin\theta_a = n_g \sin\theta_g. \tag{3.2.10}
$$

This can be used to express $\cos\theta_g$ with $\cos\theta_a$:

$$
\cos\theta_g = \sqrt{1-\sin^2\theta_g} = \sqrt{1-\left(\frac{n_a}{n_g}\right)^2 \sin^2\theta_a} = \sqrt{1-\left(\frac{n_a}{n_g}\right)^2 (1-\cos^2\theta_a)}. \tag{3.2.11}
$$

The vector $\tilde{\mathrm{m}}$ perpendicular to the interface normal $\tilde{\mathrm{n}}$, as depicted in Figure 3.13, can be computed by:

$$\tilde{\mathrm{m}} = \frac{1}{\sin\theta_\mathrm{a}}(\tilde{\mathrm{X}}_\mathrm{a} - \cos\theta_\mathrm{a}\tilde{\mathrm{n}}) \tag{3.2.12}$$

Using $\tilde{\mathrm{m}}$, it follows that:

$$\begin{aligned}
\tilde{\mathrm{X}}_\mathrm{g} &= \tilde{\mathrm{m}}\sin\theta_\mathrm{g} + \tilde{\mathrm{n}}\cos\theta_\mathrm{g} \\
&= \frac{\sin\theta_\mathrm{g}}{\sin\theta_\mathrm{a}}(\tilde{\mathrm{X}}_\mathrm{a} - \cos\theta_\mathrm{a}\tilde{\mathrm{n}}) + \tilde{\mathrm{n}}\cos\theta_\mathrm{g} \\
&= \frac{n_\mathrm{a}}{n_\mathrm{g}}(\tilde{\mathrm{X}}_\mathrm{a} - \cos\theta_\mathrm{a}\tilde{\mathrm{n}}) + \tilde{\mathrm{n}}\cos\theta_\mathrm{g} \\
&= \frac{n_\mathrm{a}}{n_\mathrm{g}}\tilde{\mathrm{X}}_\mathrm{a} + \left(-\frac{n_\mathrm{a}}{n_\mathrm{g}}\cos\theta_\mathrm{a} + \cos\theta_\mathrm{g}\right)\tilde{\mathrm{n}} \\
&= \frac{n_\mathrm{a}}{n_\mathrm{g}}\tilde{\mathrm{X}}_\mathrm{a} + \left(-\frac{n_\mathrm{a}}{n_\mathrm{g}}\tilde{\mathrm{X}}_\mathrm{a}^\mathrm{T}\tilde{\mathrm{n}} + \sqrt{1 - \left(\frac{n_\mathrm{a}}{n_\mathrm{g}}\right)^2(1-\cos^2\theta_\mathrm{a})}\right)\tilde{\mathrm{n}} \\
&= \underbrace{\frac{n_\mathrm{a}}{n_\mathrm{g}}}_{=:a}\tilde{\mathrm{X}}_\mathrm{a} + \underbrace{\left(-\frac{n_\mathrm{a}}{n_\mathrm{g}}\tilde{\mathrm{X}}_\mathrm{a}^\mathrm{T}\tilde{\mathrm{n}} + \sqrt{1 - \left(\frac{n_\mathrm{a}}{n_\mathrm{g}}\right)^2(1-(\tilde{\mathrm{X}}_\mathrm{a}^\mathrm{T}\tilde{\mathrm{n}})^2)}\right)}_{=:b}\tilde{\mathrm{n}} \\
&= a\tilde{\mathrm{X}}_\mathrm{a} + b\tilde{\mathrm{n}}.
\end{aligned} \tag{3.2.13}$$

Equation 3.2.13 shows how a ray is refracted when entering a new medium. By exchanging glass for air and water for glass, the ray in water $\tilde{\mathrm{X}}_\mathrm{w}$ can be computed respectively. After normalization, the rays $\tilde{\mathrm{X}}_\mathrm{a}$, $\tilde{\mathrm{X}}_\mathrm{g}$, and $\tilde{\mathrm{X}}_\mathrm{w}$ are known. With interface distance d and interface thickness d_g, the starting position of the ray in water on the outer interface X_s can be computed by:

$$\mathrm{X}_\mathrm{s} = \frac{d}{\tilde{\mathrm{X}}_\mathrm{a}^\mathrm{T}\tilde{\mathrm{n}}}\tilde{\mathrm{X}}_\mathrm{a} + \frac{d_\mathrm{g}}{\tilde{\mathrm{X}}_\mathrm{g}^\mathrm{T}\tilde{\mathrm{n}}}\tilde{\mathrm{X}}_\mathrm{g}. \tag{3.2.14}$$

Thus, the ray in water is determined by starting point X_s and direction $\tilde{\mathrm{X}}_\mathrm{a}$ for flat port underwater housings.

In case of dome port housings with perfect alignment of the dome center and the camera's center of projection, the project and back project functions are equal to the common pinhole camera model. However, in case of imperfectly fitted dome ports, a ray needs to be refracted using Equation (3.2.14). The only difference is that the normal $\tilde{n}$ needs to be computed for each ray using the ray in air $\tilde{X}_a$ and the dome port center X_{dome}. In order to compute the intersection point, the inner and outer dome spheres are parametrized by using the quadric:

$$\mathbf{Q} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \tag{3.2.15}$$

A transformation containing the sphere's inner d or outer $d + d_g$ radius and the translation of the dome's center $X_{dome} = (X_{dome}, Y_{dome}, Z_{dome})^T$ is applied to the quadric to get the matrix describing the dome [HZ04]:

$$\begin{aligned} \mathbf{H}_d &= \begin{pmatrix} d & 0 & 0 & X_{dome} \\ 0 & d & 0 & Y_{dome} \\ 0 & 0 & d & Z_{dome} \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ \mathbf{D}_d &= (\mathbf{H}_d^{-1})^T \mathbf{Q} \mathbf{H}_d^{-1}. \end{aligned} \tag{3.2.16}$$

A homogeneous point $\mathbf{X}$ lies on the quadric $\mathbf{D}_d$ if $\mathbf{X}^T \mathbf{D}_d \mathbf{X} = 0$. Using the parametrization for the ray in air or in glass, the intersections of the rays with the inner or outer dome surface can be determined. The normals at those intersection points can be found by using the line from the center of the dome to the intersection points. Once the normals, the intersection points, and the ray directions in air and glass are known, the remaining derivation of the refraction is exactly the same as in the flat port case.

Additional Constraints

Agrawal et al. derived in [ARTC12] two constraints for flat port underwater cameras. Both are based on an existing correspondence between a 3D point

X and a 2D image point $\mathbf{x}$. For the image point, the corresponding ray segments in water $\tilde{X}_w$ and air $\tilde{X}_a$ and the starting point on the outer interface X_s are determined using Equations (3.2.13) and (3.2.14). The first constraint is based on the idea that once a 3D point X is transformed into the camera coordinate system, and then translated to the corresponding starting point on the outer interface, it should be parallel to the corresponding ray in water $\tilde{X}_w$, resulting in the Flat Refractive constraint (FRC):

$$(\mathbf{R}^T X - \mathbf{R}^T C - X_s) \times \tilde{X}_w = 0. \tag{3.2.17}$$

The second constraint introduced by Agrawal et al. in [ARTC12] is based on the fact, that the all ray segments (in air, glass, and water) lie in a common plane, the Plane of Refraction (POR). For a corresponding 3D point X, which is transformed into the local camera coordinate system, follows that it has to lie in the POR as well:

$$(\mathbf{R}^T X - \mathbf{R}^T C)^T (\tilde{n} \times \tilde{X}_a) = 0. \tag{3.2.18}$$

Projection from 3D to 2D

The previous section described the back-projection of a 2D image point onto the corresponding 3D ray. Back-projections can be computed efficiently in case of refractive underwater cameras. More involved is the computation of the projection of a 3D point into the image. Here, it is unknown, where the ray intersects the inner and outer interface planes and therefore at which angles it is refracted. [KS08] use the above described back-projection function in a numerical optimization scheme, which is initialized using the common perspective projection. In our implementation, the Levenberg-Marquardt algorithm is used to compute the correct 2D point. Using such an optimization scheme is time-consuming, however, it allows to compute the correct projection even in case of the entrance pupil, and thus the camera center, lying in front the actual glass (Section 2.2.2).

Another approach can be derived building upon [TSS08] and generalizing their proposed method to incorporate thick glass and a tilted interface. A formula for the projection is derived by applying Fermat's principle. The total traveling time of the ray is the sum of three components: the time spent in the underwater housing (in air), the time spent in the glass

of the interface, and the time spent in the water. The derived equation contains four unknowns, the x- and y-coordinates on the inner and outer interface planes (X_{in} and Y_{in} and X_{s} and Y_{s} respectively) and depends on the 3D point $\mathsf{X} = (X, Y, Z)^{\mathrm{T}}$:

$$\begin{aligned} t(X_{\text{in}}, Y_{\text{in}}, X_{\text{s}}, Y_{\text{s}}) = \\ & n_{\text{a}} \sqrt{X_{\text{in}}^2 + Y_{\text{in}}^2 + Z_{\text{in}}^2} + \\ & n_{\text{g}} \sqrt{(X_{\text{s}} - X_{\text{in}})^2 + (Y_{\text{s}} - Y_{\text{in}})^2 + (Z_{\text{s}} - Z_{\text{in}})^2} + \\ & n_{\text{w}} \sqrt{(X - X_{\text{s}})^2 + (Y - Y_{\text{s}})^2 + (Z - Z_{\text{s}})^2}. \end{aligned} \tag{3.2.19}$$

Since the light always travels the path that takes the least time to traverse (Fermat's principle), this equation's partial derivatives are used to minimize the traveling time with respect to the unknowns:

$$\frac{\partial t}{\partial X_{\text{in}}} = 0 \qquad \frac{\partial t}{\partial Y_{\text{in}}} = 0 \qquad \frac{\partial t}{\partial X_{\text{s}}} = 0 \qquad \frac{\partial t}{\partial Y_{\text{s}}} = 0. \tag{3.2.20}$$

The plane equations are utilized to eliminate the Z-components with $\tilde{\mathsf{n}} = (n_1, n_2, n_3)^{\mathrm{T}}$:

$$\begin{aligned} Z_{\text{in}} &= \frac{d - n_1 X_{\text{in}} - n_2 Y_{\text{in}}}{n_3} \\ Z_{\text{s}} &= \frac{d + d_{\text{g}} - n_1 X_{\text{s}} - n_2 Y_{\text{s}}}{n_3}. \end{aligned} \tag{3.2.21}$$

The resulting system of equations with four unknowns and four equations is solved numerically, starting from an initial solution, using e. g., Powell's hybrid method[1] [PVT+02]. After that, the points on the inner and outer interface planes are determined, however, only the point on the inner interface plane is relevant for projecting it onto the image plane with the usual perspective projection. In our tests, we found that it is difficult to find the correct solution using this method, especially in case of a negative camera-interface distance. In case of thin or no glass, parallelism between interface and image sensor, and positive interface distance d, (3.2.20) is only

[1] e. g., in GSL library from www.gnu.org/software/gsl/

depending on the radial coordinate on the refractive plane. The derivative in this direction becomes a polynomial of fourth degree [GS00, TSS08]. For this special case, [GS00] proved that the correct/practical root is found in the interval $[0, R_w]$. Experiments showed that in the more general case, thick glass with possibly negative d and non-parallel interface, this is no longer true.

Most recently, [ARTC12] determined that a flat port refractive camera can be classified as an axial camera, i. e., all rays intersect a common axis. By using this insight, Agrawal et al. developed a projection method by deriving a 12^{th} degree polynomial based on the idea of solving the projection on the plane of refraction (POR). After eliminating all complex roots from the set of solutions, Snell's law can be used to determine the correct root. This approach is far more time-efficient than both approaches described above.

3.3 Simulation Results

Until now, this chapter has described effects of water on image formation. While the light travels through the water, light intensity is effected depending on the color channels, then, the light is refracted when entering the underwater housing either through a flat port interface or a dome port interface. The described adaptations to the Jaffe-McGlamery model and the described equations for refractions allow to implement a simulator that can be used to render underwater images. Note that the following results have already been published [SK11b].

The simulator can render camera trajectories of a textured 3D model, and hence allows to create synthetic ground truth data required for testing refractive calibration and reconstruction methods. The simulator has been realized as a ray-caster, i. e., for each pixel the refractive 3D ray is computed that starts on the outer interface surface. It is determined if any triangles of the model are intersected by this ray, how this triangle is illuminated (direct, attenuated lighting from possibly several light sources or shadows), or how much backscatter needs to be added. Triangle illumination and back-scattering are determined using Equations (3.1.14) (3.1.15) and (3.1.18).

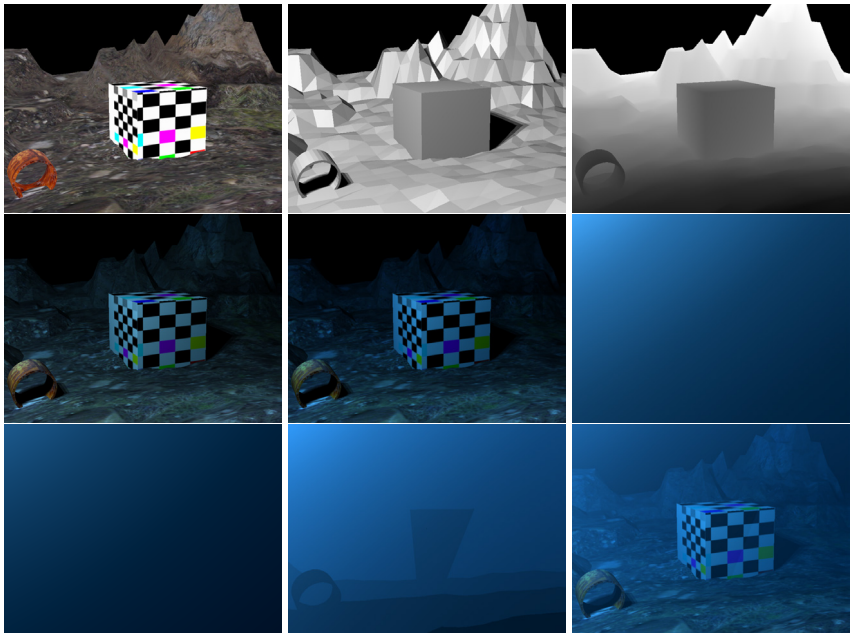

Figure 3.14. Lamps are placed on the left hand side of the camera. Single tiles, top row from left to right: scene, 3D surface with shadows, depth map. Middle row: light incident on 3D structure, signal, first backscatter plane. Bottom row: second backscatter plane, overall backscatter, and final result.

The required model for refraction made it necessary to implement the simulator instead of using a ready-made software package. The implementation is build upon BIAS[2] and the open source library OpenSceneGraph[3]. It was assumed that mainly the lighting situation created by using ROVs is of importance. A ROV usually carries a set of lights at its front illuminating the scene in front of it. Also somewhere at the front are the cameras, consequently, it can be assumed that the camera(s) and lights are roughly in one common plane. The simulator allows placing an arbitrary number of point light sources (refer to Section A.1) relative to the camera.

[2] http://www.mip.informatik.uni-kiel.de/tiki-index.php?page=BIAS
[3] http://www.openscenegraph.org/projects/osg

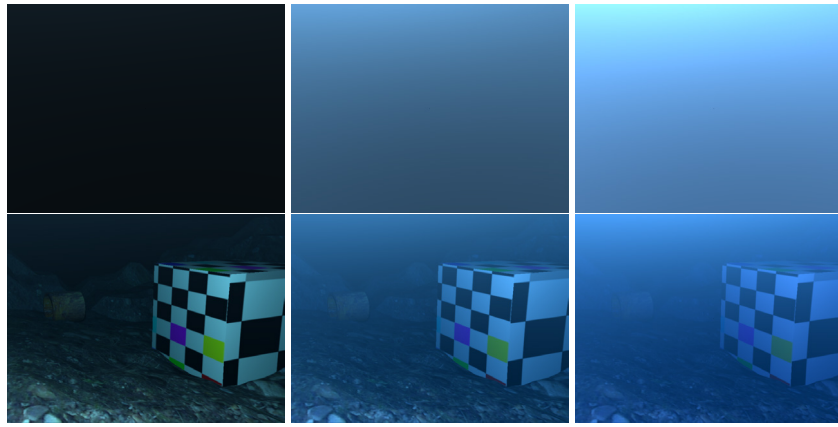

Figure 3.15. From left to right, increasingly turbid water. In all three cases the lamps were positioned above the camera, but close by. Top row: backscatter components. The minimum and maximum image values show how strongly backscatter increases with increasing turbidity (min/max backscatter image values over the whole image from left to right: 7.74/16.6, 35.62/78.98, 70.13/156.23). Bottom row: complete result.

Using this set-up, arbitrary, textured surface meshes can be rendered. This was tested using a synthetic model of an underwater scene, containing a background landscape, a cube textured with a checkerboard pattern, and a rusty tube. Three lights were placed to the upper left of the camera. Figure 3.14 shows the final rendering result and the different components. The surface rendering clearly shows the shadows produced by three lamps at the upper left of the camera. Geometric ground truth is rendered in form of depth maps recording the camera-3D point distance for each pixel. The fourth image shows the light that is incident on the structure including the texture and reflection. After adding forward scatter and attenuating on the way to the camera, the signal (fifth image) results. The first backscatter plane is not yet occluded by the structure and clearly shows the falloff of the backscatter with growing distance from the light source. The final result is then the sum of the signal and the backscatter components.

The physical model for light propagation implemented in the simulator allows to try out different camera-light configurations, but also to simulate

Figure 3.16. The lamp was moved from being directly at the camera to 2 m at the upper right. As Jaffe already concluded in his work, the backscatter portion is causing the contrast to decrease if the lamp is close to the camera. The top row shows the backscatter component and the bottom row the resulting images.

different kinds of water, i. e., to simulate poor or good visibility, etc. For example Figure 3.15 shows the increasingly limited contrast due to strong backscatter in increasingly turbid water. In Figure 3.16, a lamp was first incident with the camera and then moved away increasing light-camera distance to approximately 2 m. This was done in fairly turbid water, hence, a large amount of backscatter reaches the camera. No structure, only the backscatter is visible in the first image. With growing camera-light distance, more of the structure becomes visible. This is in accordance with one of the major conclusions in [Jaf90] concerning the placement of camera and light in underwater imaging scenarios: there is always a compromise between contrast and power reaching the imaging system due to backscatter and attenuation.

Apart from simulating the color effects, the simulator can also demonstrate how strongly refraction distorts imaging geometry compared to using the same camera in air. Formerly straight lines become curves, and the whole scene is enlarged by a factor of 1.333, the index of refraction for water. This can be observed in Figure 3.17, where all three images are partly rendered with and without water.

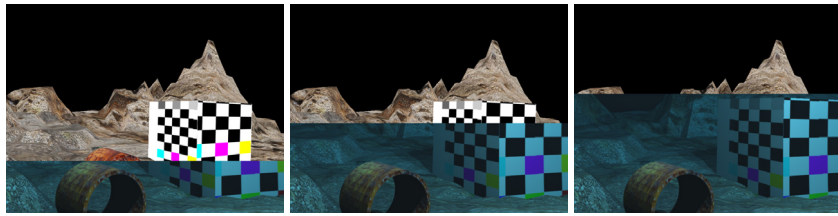

Figure 3.17. All three images are partly rendered with and without water simulation. Note how much geometry differs at the borders between water and air.

3.4 Summary

This chapter introduced the effects of water on image formation. In summary, while still traveling through the water, light is attenuated and scattered, mostly depending on the light's wavelength and the distance traveled. Effects on color can be modeled with the Jaffe-McGlamery model or the proposed extension. It allows to render the effect in synthetic images. A strong simplification of the model that has been widely used in the literature in the recent years can be parametrized using checkerboard images captured in the local water body. This allows to correct image colors if the camera-object distance is known. Once, the light enters the underwater housing, it is refracted at the glass port. In the literature, this effect is often approximated using the perspective camera model. Only recently, explicitly modeling refraction has gained some interest. However, until now, no complete system for refractive SfM with general camera movement exists. Additionally, the chapter introduced the methods for projection and back-projection using the refractive camera model with thin glass and a possibly tilted interface. The simulator can render synthetic ground truth images with refractive and color effects that can be utilized in experiments.

Chapter 4

Calibration of Camera and Underwater Housing

The perspective and refractive camera models described in the previous chapters are parametrized by a set of parameters that can be grouped into intrinsic camera parameters, housing parameters, and extrinsic camera parameters including the rigid transformations between cameras of a rig. In order to work with the implicitly contained geometry of the images, all camera parameters need to be calibrated. This chapter will describe the necessary calibration approaches starting with the calibration of the camera's intrinsic parameters. Then, the calibration of the camera housing parameters will be described for flat port underwater housings. At this point, the parameters for underwater light propagation can be conveniently calibrated in addition to the geometric properties. The last section evaluates the calibration routines, with special emphasis on a comparison of accuracy between calibrating on underwater images perspectively and calibrating the housing parameters. Note that the described housing calibration approach and its results on synthetic images have already been published in [JSK12]. Results of the calibration of the underwater light propagation model conclude the chapter.

4.1 Perspective Calibration

As briefly described in Section 2.2.1, the perspective camera model is parametrized by focal length f, aspect ratio ar, principal point c_x and c_y, skew s, and radial r_1, r_2 and tangential distortion t_1, t_2. In case of a rig with several rigidly coupled cameras, one camera is appointed to

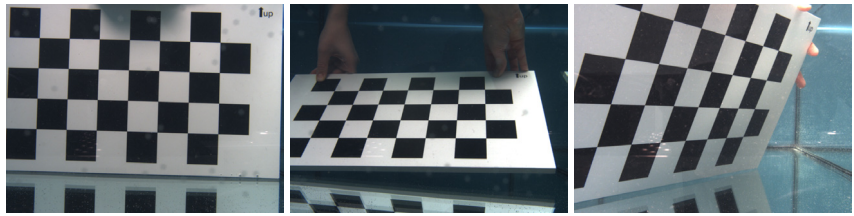

Figure 4.1. Exemplary underwater checkerboard images from different points of view.

be the master camera, which is defined to be in the origin of the rig's coordinate system. Accordingly, all other cameras, called slave cameras, need to be calibrated with respect to the master camera. For multiple camera rigs of perspective, rigidly coupled cameras, several calibration methods exist. Often, they rely on special calibration targets, for example a planar checkerboard (Figure 4.1). The camera to be calibrated is used to capture a set of images of the calibration target from different view points (Figure 4.2), then, the checkerboard corners are detected in all images and the resulting 2D-3D-correspondences are used for calibration. In this thesis, [SBK08] is used for perspective calibration, which is based on the standard single-camera-calibration routine of OpenCV[1], and then proposes to optimize the initial solution using a model-based analysis-by-synthesis (AbyS) approach [Koc93], where the camera model and a set of parameters is used to render synthetic checkerboard images, which are then compared to the real, captured checkerboard images. The pixel-wise error is minimized during optimization. The AbyS approach makes the algorithm independent of errors in corner detection. Additionally, Schiller et al. propose how to calibrate PMD-cameras, i. e., infrared-based, active cameras, which measure camera-object distance. However, this feature is not used within the context of this thesis. Considering a set of N different poses of captured checkerboard images with a rig of M cameras, the following set of parameters needs to be calibrated:

- N poses of the rig, i. e., master camera poses defined in the world coordinate system ($6N$),

[1] `http://opencv.org/`

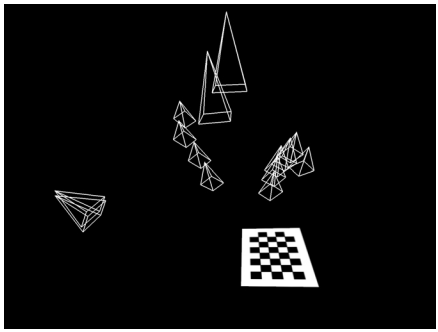

Figure 4.2. Calibration scenario with different views of the checkerboard.

- M cameras within the rig, i. e., each slave camera has a pose relative to the master camera ($6(M-1)$),
- each camera within the rig has a set of intrinsic parameters containing focal length, principal point, and lens distortion ($8M$), and
- 2 parameters for each image for gray-level offset and white balance ($2NM$).

Note that the calibration approach uses gray-level images, and hence for each image one parameter is used for white balancing and one for offset. OpenCV returns an initial solution with independent poses for each image and an initial set of intrinsics for each camera in the rig. By definition, the checkerboard lies in the xy-plane of the global coordinate system with the upper left corner being in the world coordinate system origin. The camera poses are determined relative to the checkerboard. After enforcing the constraints on poses of a rigidly coupled rig, AbyS is used for non-linear optimization. Note that after corner detection, it is known which set of pixels $\mathbf{x}_k$, $k \in \{1, ..., K\}$ in each image observed the checkerboard. The error function is derived as follows: for each pixel $\mathbf{x}_{ijk}$ seeing the checkerboard in image $i \in \{1, ..., N\}$, with rig camera $j \in \{1, ..., M\}$, at pixel position $k \in \{1, ..., K\}$, a ray in the local camera coordinate system is determined using the camera's intrinsic parameters that is defined by a starting point

$\mathsf{X}^{cc}_{s_{ijk}}$ and a direction $\tilde{\mathsf{X}}^{cc}_{a_{ijk}}$:

$$(\mathsf{X}^{cc}_{s_{ijk}}, \tilde{\mathsf{X}}^{cc}_{a_{ijk}}) = \mathrm{Ray}_{persp}(\mathbf{x}_{ijk}, \mathbf{K}, r_1, r_2, t_1, t_2, \mathbf{R}_j, \mathsf{C}_j), \tag{4.1.1}$$

where the function Ray_{persp} is derived from Equations (2.2.3) and (2.2.4). In case of the camera not being the master camera, the inverse, relative rig transformation comprised of $\mathbf{R}_j$ and C_j is applied. Then, the ray is transformed into the world coordinate system by:

$$(\mathsf{X}^{wc}_{s_{ijk}}, \tilde{\mathsf{X}}^{wc}_{a_{ijk}}) = (\mathbf{R}_i \mathsf{X}^{cc}_{s_{ijk}} + \mathsf{C}_i, \mathbf{R}_i \tilde{\mathsf{X}}^{cc}_{a_{ijk}}). \tag{4.1.2}$$

Based on the ray in world coordinates, $\kappa_{ijk} \in \mathbb{R}$ can be computed such that:

$$\mathsf{X}_{ijk} = \mathsf{X}^{wc}_{s_{ijk}} + \kappa_{ijk} \tilde{\mathsf{X}}^{wc}_{a_{ijk}} \tag{4.1.3}$$

intersects the xy-plane and thus the checkerboard in X_{ijk}. Using X_{ijk} and information about the checkerboard (square sizes and number of squares), the pixel color based on the parameter set can be synthesized $I_{ren}(\mathbf{x}_{ijk}) = \alpha_{ij} I_{check}(\mathbf{x}_{ijk}) + \beta_{ij}$ and compared to the measured pixel color $I(\mathbf{x}_{ijk})$, with α_{ij} and β_{ij} being a white balancing factor and offset respectively. This leads to the following residual r based on an explicit error function that can be minimized using the Gauss-Markow model (Section A.4):

$$\mathsf{r} = \mathsf{f}_{AbyS-persp}(\mathsf{p}_{AbyS-persp}) - \mathsf{l} := \begin{bmatrix} I_{ren}(\mathbf{x}_{111}) - I(\mathbf{x}_{111}) \\ \vdots \\ I_{ren}(\mathbf{x}_{ijk}) - I(\mathbf{x}_{ijk}) \\ \vdots \\ I_{ren}(\mathbf{x}_{NMK}) - I(\mathbf{x}_{NMK}) \end{bmatrix}. \tag{4.1.4}$$

In summary, for each image, the proportion showing the checkerboard is rendered using the current parameter set and then compared to the captured image. This yields a pixel-wise function $\mathsf{f}_{AbyS-persp}$ for which the Jacobian can be computed numerically and which is then minimized using the Gauss-Markow method described in Section A.4.

4.2 Underwater Housing Calibration

Once the intrinsic parameters have been calibrated, the housing parameters of all cameras in the rig can be determined in a similar fashion. The method proposed in [JSK12] uses the refractive method introduced in [ARTC12] for initialization instead of OpenCV. This gives an initial estimation of camera poses and housing parameters interface distance d and the interface normal $\tilde{n}$ with respect to the optical axis. Usually, it can be assumed that the type of glass, and hence the index of refraction n_g, and the thickness d_g of the interface are known, consequently, they are not optimized in this approach. Using the checkerboard corners, this initial solution is optimized. Then, an AbyS approach similar to the perspective one above is applied, thereby making the method invariant against errors in corner detection. In contrast to [SBK08] a different optimization routine is utilized, for the reason that even in established methods like [SBK08], it is a well-known problem that correlations between parameters can cause the algorithm to converge towards a local minimum instead of the global optimum. Therefore, instead of using the Gauss-Markow model for housing calibration, CMA-ES (Covariance Matrix Adaptation - Evolution Strategy) [HO01](Section A.4.2), an evolutionary algorithm, is used for optimization. It performs well on non-linear, non-convex error functions with noisy observations and has been successfully used in computer vision before [SGdA+10, JK11].

Similar to the perspective calibration described above, the following parameters are optimized:

- N poses of the rig, i. e., master camera poses defined in the world coordinate system ($6N$),
- M cameras within the rig, i. e., each slave camera has a pose relative to the master camera ($6(M-1)$),
- each camera within the rig has an interface with tilt and distance ($3M$), and
- 2 parameters for each image for grey-level offset and white balance ($2NM$).

Once the geometric calibration is completed, the knowledge about black and white areas on the checkerboard can be used to calibrate the simple model for underwater light propagation (Equation (3.1.19)). In contrast to the first step, where gray level images were used, color images are required for calibrating underwater light propagation, hence, the following parameters need to be optimized in this step:

- 2 parameters for water color correction for each color channel (6) and
- 2 parameters for each color channel and image parametrizing offset and white balance ($6NM$).

As in the perspective case described above, for each pixel $\mathbf{x}_{ijk}$ that saw the checkerboard in image $i \in \{1, ..., N\}$, with rig camera $j \in \{1, ..., M\}$, at pixel position $k \in \{1, ..., K\}$, a ray in the local camera coordinate system needs to be computed. However, in contrast to the perspective case, this time, the starting point $\mathsf{X}^{\mathrm{cc}}_{\mathrm{s}_{ijk}}$ lies on the outer glass interface plane and the direction is the ray in water $\tilde{\mathsf{X}}^{\mathrm{cc}}_{\mathrm{w}_{ijk}}$. The function $\mathrm{Ray}_{\mathrm{refr}}$ computes the ray in water based on Equations (3.2.8) to (3.2.14), which is then transformed into the global coordinate system and intersected with the xy-plane determining the scalar $\kappa_{ijk} \in \mathbb{R}$:

$$(\mathsf{X}^{\mathrm{cc}}_{\mathrm{s}_{ijk}}, \tilde{\mathsf{X}}^{\mathrm{cc}}_{\mathrm{w}_{ijk}}) = \mathrm{Ray}_{\mathrm{refr}}(\mathbf{x}_{ijk}, d, \tilde{\mathrm{n}}, \mathbf{R}_j, \mathsf{C}_j) \tag{4.2.1}$$

$$(\mathsf{X}^{\mathrm{wc}}_{\mathrm{s}_{ijk}}, \tilde{\mathsf{X}}^{\mathrm{wc}}_{\mathrm{w}_{ijk}}) = (\mathbf{R}_i \mathsf{X}^{\mathrm{cc}}_{\mathrm{s}_{ijk}} + \mathsf{C}_i, \mathbf{R}_i \tilde{\mathsf{X}}^{\mathrm{cc}}_{\mathrm{w}_{ijk}}) \tag{4.2.2}$$

$$\mathsf{X}_{ijk} = \mathsf{X}^{\mathrm{wc}}_{\mathrm{s}_{ijk}} + \kappa_{ijk} \tilde{\mathsf{X}}^{\mathrm{wc}}_{\mathrm{w}_{ijk}}. \tag{4.2.3}$$

After initialization, the known 2D corners of the checkerboard that were detected in each image are used in a non-linear optimization, i. e., for each 2D corner, the corresponding point on the 3D checkerboard plane X_{ijk} is computed and compared to the real 3D checkerboard point $\check{\mathsf{X}}_k$. Since in contrast to the perspective method described above, CMA-ES is to be used for optimization, the fitness function is the sum of squared distances for all checkerboard corners:

$$E_{\mathrm{corner-refr}} = \sum_{i<N} \sum_{j<M} \sum_{k<K} ||\mathsf{X}_{ijk} - \check{\mathsf{X}}_k||_2^2. \tag{4.2.4}$$

After that, the AbyS method is applied, where X_{ijk} is used to determine the checkerboard color and the corresponding rendered color $I_{\text{ren}}(\mathbf{x}_{ijk}) = \alpha_{ij} I_{\text{check}}(\mathbf{x}_{ijk}) + \beta_{ij}$ and then compared to the measured image color $I(\mathbf{x}_{ijk})$ for each pixel. The fitness function to be optimized using CMA-ES is the sum of squared errors for all observations:

$$E_{\text{AbyS-refr}} = \sum_{i<N} \sum_{j<M} \sum_{k<K} (I_{\text{ren}}(\mathsf{x}_{ijk}) - I(\mathsf{x}_{ijk}))^2. \tag{4.2.5}$$

Once, geometric calibration is completed, a set of images with known camera poses and known black and white areas exists. That is a unique situation, which can be utilized to calibrate the underwater light propagation model described in Equation (3.1.19). In contrast to the geometric calibration, this time, color images need to be used. For each color channel $\lambda \in \{R, G, B\}$, initially $\alpha_\lambda = 1$ and $\beta_\lambda = 0$. Then,

$$E_{\text{cam}_\lambda} = \alpha_{\lambda_{ij}} (E_{\text{obj}_\lambda} + e^{-\eta_\lambda z} + B_{\infty_\lambda} (1 - e^{-\eta_\lambda z})) + \beta_{\lambda_{ij}}, \quad \lambda \in \{R, G, B\} \tag{4.2.6}$$

can be used to determine initial values for η_λ and B_{∞_λ} by setting E_{obj_λ} to one or zero depending on the checkerboard color. Afterwards, Equation (4.2.6) is used in a non-linear Levenberg-Marquardt routine to optimize η_λ and $B_{\infty\lambda}$ for the whole scene and $\alpha_{ij\lambda}$ and $\beta_{ij\lambda}$ as additional parameters for white balance and offset for each image and color channel.

4.3 Experiments

The preceding sections introduced methods for perspective and refractive calibration of cameras. The method for perspective calibration allows to calibrate a perspective camera in air in order to determine its intrinsic parameters, possibly in addition to the relative transformations between rig cameras. The results can then be used to calibrate the underwater housing configuration using checkerboard images captured below water. However, as described in Section 3.2.2, a lot of methods in the literature use the perspective model for underwater images, even though a systematic model error occurs. In order to do that, checkerboard images captured below water are used to calibrate the perspective camera model, which

causes the parameters to absorb the model error to some extent. The next section will extend the analysis shown in [SK12] and provides some new insights on how interface distance or tilt changes affect the calibrated intrinsic parameters.

4.3.1 Perspective Calibration on Underwater Images

For this experiment, a set of 68 different housing configurations was created with $d \in \{-10\,\text{mm}, 0\,\text{mm}, 10\,\text{mm}, 20\,\text{mm}, ..., 150\,\text{mm}\}$, $\theta_1 = 30^\circ$, and $\theta_2 \in \{0^\circ, 1^\circ, 2^\circ, 3^\circ\}$. The interface thickness was always $d_g = 30\,\text{mm}$. For all sets, a stereo camera rig was used with no rotation between the cameras and a translation of 200 mm along the x-axis. The image size of both images was $800 \times 600\,\text{px}$ with a focal length of 800 px. The principal point was in the middle of the image and lens distortion was set to zero. For each housing configuration, a set of 30 stereo underwater checkerboard images was rendered from different points of view at a camera-object distance between 1000 mm and 4000 mm. Then, each configuration was calibrated perspectively using [SBK08] (described in Section 4.1). [FF86] and [LRL00] noted that the focal length in water is $f_w = 1.333 f_a$, however, Figure 4.3 on the top left shows an additional dependence on interface distance and tilt. An additional discrepancy to the results in [FF86] and [LRL00] is the direct influence of the interface tilt on the principal point (Figure 4.3 second and third plot). Clearly, the error introduced by tilting the interface is systematically absorbed by the principal point. The fourth and fifth plots show the resulting coefficients for radial distortion r_1 and r_2. Note that the actual perspective camera within the housing had zero distortion, hence, all of the measured distortion is absorbing the refractive effect and depends strongly on interface distance and tilt. The final plot in the bottom row on the right shows the average error distance that arises when a set of 3D points with different distances to the camera is projected with the ground truth refractive projection and the calibrated perspective projection. Note that with growing interface distance, but especially with increasing interface tilt, the error increases. During the calibration process, the average reprojection error was $\varnothing \approx 0.05\,\text{px}$, on exact corners, i. e., without any corner detection errors. The large discrepancy to Figure 4.3 (bottom, right) indicates that not only the intrinsics absorb the systematic

model error, but also the extrinsics parameters, which are adapted for each checkerboard view independently during calibration, but not in Figure 4.3 (bottom, right).

In Section 3.2.2, it was shown that the approximation of refraction using the perspective camera model is only correct for a certain distance from the camera (Figure 3.11). This distance is determined by the calibration algorithm and the set of underwater checkerboard images captured. An interesting question in this context is where this best fit occurs for the different housing configurations. Figure 4.4 examines the reprojection errors depending on the camera-3D point distance for five exemplary housing configurations of the above described image set. For each distance, the principal point, the four corner points, and an additional four points between the principal point and the corner points were projected. Figure 4.4 shows the resulting mean reprojection errors and standard deviations for both rig cameras compared to the true refractive projection. The most interesting case is the first one, where the error is almost zero. Treibitz et al. [TSS08] noted that in case of very thin glass, zero interface distance, and zero interface tilt, the refractive effect can be absorbed completely by the perspective camera model. In this experiment, the interface thickness was $d_g = 30\,\text{mm}$. The ground truth underwater housing configuration for the upper right result in Figure 4.4 has $d = 10\,\text{mm}$ and zero interface tilt. It has the lowest average reprojection error of all cases, thus is the configuration with the most accurate perspective approximation of the refractive camera model. A comparison of all five test cases reveals that the best fit of the perspective projection can be at different distances from the camera. This indicates that it is not possible to predict where the closest fit will be because it is determined by the captured checkerboard images. In the next section, the caustic sizes depending on interface distance and tilt for the 68 different housing configurations will be presented.

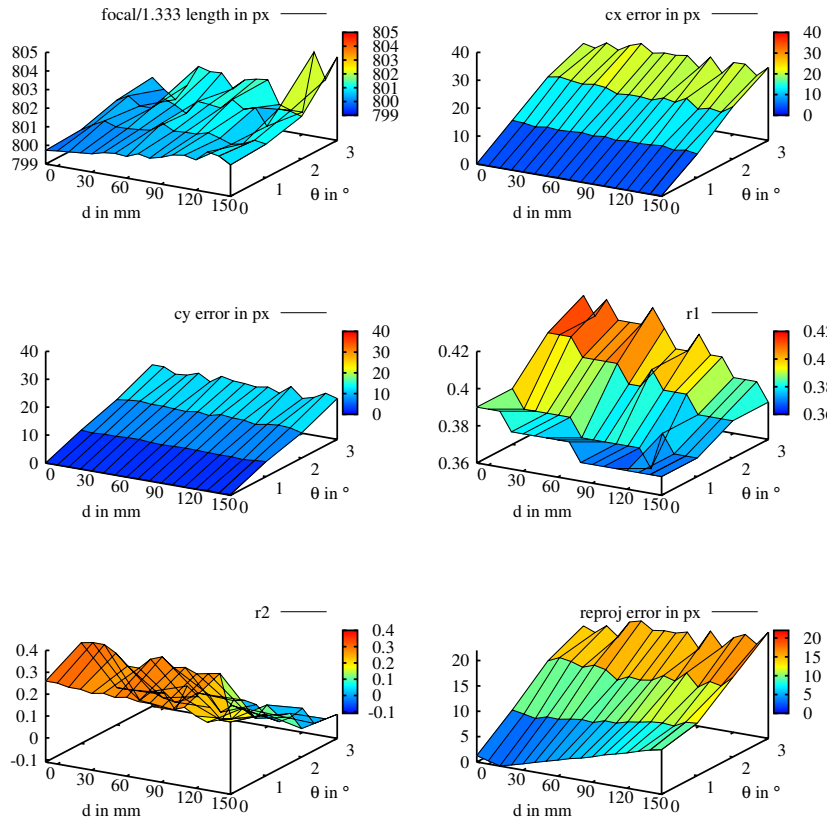

Figure 4.3. Calibration results of calibrating the perspective camera model on underwater images. The interface distance varied from -10 mm to 150 mm and the interface tilt between 0° and 3°. Left column, from top to bottom: focal length $f/1.333$ (true $f = 800$ px), principal point errors for c_x and c_y (true principal point was in the middle). Right column, from top to bottom: resulting radial distortion coefficients r_1 and r_2 (true $r_1 = 0$, $r_2 = 0$, and reprojection error of resulting calibration on a random set of 3D points.

	d in mm	θ in $^\circ$
1	10	0
2	150	0
3	130	2
4	20	3
5	60	3

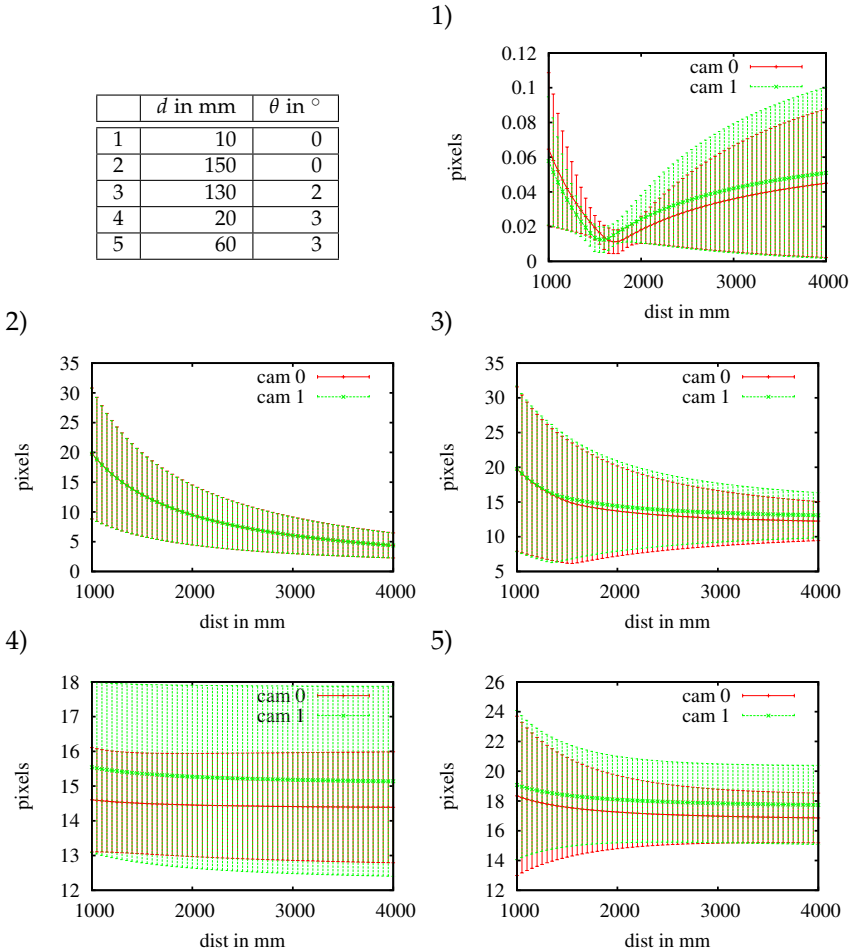

Figure 4.4. Exemplary, distinct error curves for perspective calibration of the described synthetic data set depending on the distance between 3D point and camera. The table on the upper left shows the housing configuration for the five following perspective error plots for both cameras of the stereo rig.

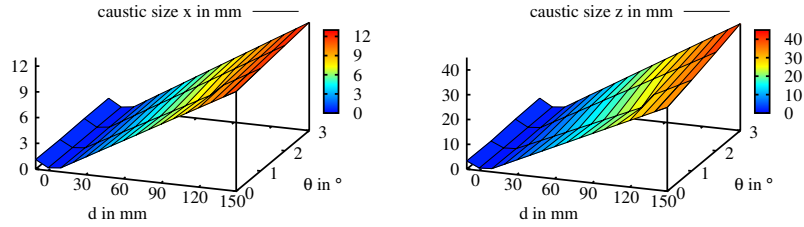

Figure 4.5. Caustic sizes in x- and z-directions, depending on interface distance d and interface tilt θ.

4.3.2 Caustics

In Section 2.2.3 caustics were introduced as the singularity of the bundle of rays. In case of the camera being perspective, i. e., having a single view point, the caustic size is zero because all rays intersect the center of projection. Hence, for nSVP cameras, the caustic size serves not only as a characterization of the camera, but also a as a measure of deviation from the single-view-point camera model. Figure 4.5 depicts the caustic sizes in x- and z-direction for all housing configurations in the synthetic data set described above. Clearly, the caustic is smallest in case of zero interface distance and tilt, where according to [TSS08], the perspective camera model is valid, at least in case of very thin glass. With growing interface distance, the caustic size increases, while a stronger interface tilt mainly causes the caustic to become asymmetrical and to only slightly increase in size.

4.3.3 Stereo Measurement Errors

A calibrated stereo rig can be used in case the scene or object of interest is not rigid, e. g., moving gas bubbles or fauna. In these cases, the classic 3D reconstruction approach cannot easily be applied. It is however possible to capture images with a synchronized stereo camera rig. Especially, when using only two images, the measurement error due to refraction can be

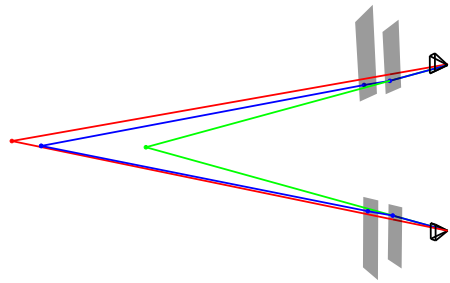

Figure 4.6. Stereo triangulation with different camera configurations. A stereo camera rig is depicted with interface distance $d = 70\,\text{mm}$, interface thickness $d_g = 30\,\text{mm}$, and interface tilt $\theta = 2°$. The baseline between the cameras is 200 mm long. Three different points have been triangulated. Blue: refractive triangulation. Green: perspective triangulation without approximation. Red: perspective triangulation with approximation, i. e., underwater checkerboard images were rendered, then used for perspective calibration, allowing focal length, principal point, and radial distortion to absorb the bulk of the refractive effect.

very large, sometimes larger than the objects to be measured. Figure 4.6 visualizes the effect. A stereo camera rig is used to triangulate 3D points, showing a comparison of different camera models for triangulation. In blue, the refractively triangulated point, which is the correct point is given. Shown in red is the point triangulated using the perspective camera model with a calibration based on underwater images. As described and analyzed above, most of the refractive effect has been absorbed by focal length, principal point, and radial distortion. However, the red point is still not correct. The green point has been triangulated by ignoring refraction completely, i. e., triangulated perspectively with the calibration of the cameras based on checkerboard images captured in air and demonstrates why objects imaged underwater appear enlarged. Since the error of the green point is very large, it will not be considered any further. More interesting is the error of the perspective triangulation with approximation of refraction. Figure 4.7 shows 3D triangulation errors and the resulting distance measurement errors between two points. The distance between camera and 3D points was between 1000 mm and 1500 mm and for each interface distance and tilt combination 1000 pairs of 3D points were ran-

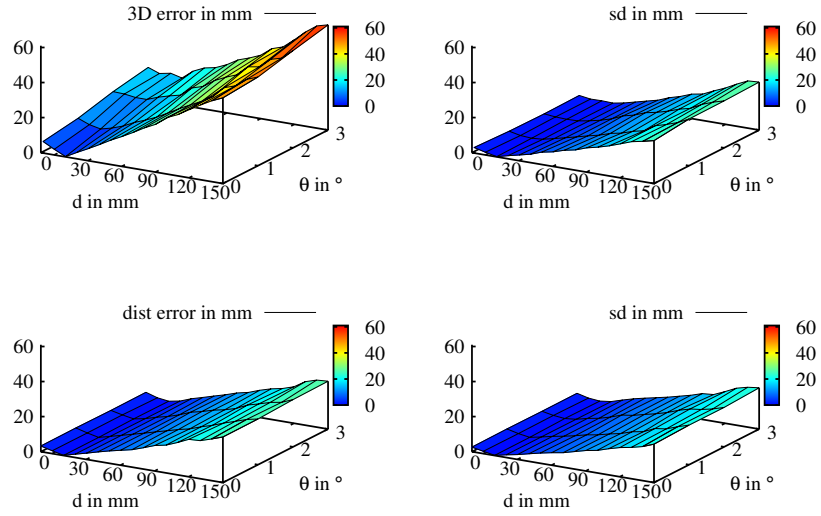

Figure 4.7. Errors in triangulation and stereo distance measurement depending on interface distance and interface tilt. Top: average 3D triangulation errors and standard deviation (sd) for 3D points being between 1000 mm and 1500 mm away from the camera. Bottom: distance measurement errors resulting from the 3D triangulation errors and the standard deviation for the distance errors.

domly generated sampling the whole overlap between the two views, because the error depends on the incidence angle of the rays in air on the interface. Figure 4.7 shows a strong dependence of the error depending on the interface distance and angle.

The dependence on the pixel position in the image, and consequently on the incidence angle between the ray in air and the interface can be observed in Figure 4.8, where the blue planes are triangulated using the underwater model (interface distance 120 mm, glass thickness 30 mm), while the blue surfaces result from triangulation using perspective calibrations. The left image shows results for a refractive camera with zero interface tilt, while in the right image, a slight rotation of the interface

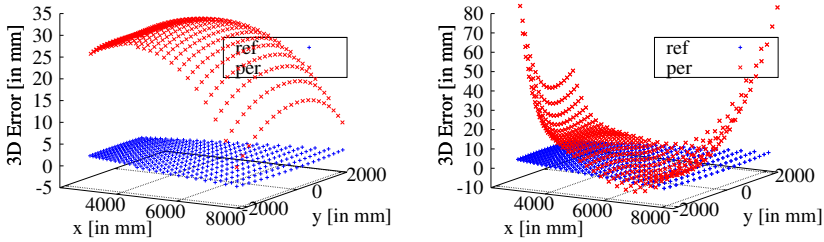

Figure 4.8. Two of the perspective calibration scenarios, both with interface distance 120 mm were used to triangulate points on the xy-plane, with the camera being approximately 2 m away, viewing the xy-plane at a 45° angle. The left scenario has a parallel interface with respect to the imaging sensor, while in the right scenario the interface was tilted by $(30°, 2°)$ with the resulting errors in the perspective calibration.

plane causes far larger errors.

Note that the actual error depends on variables such as the camera's focal length and image resolution, interface distance and tilt, camera-object distance, and stereo baseline, but also on the accuracy of the calibration of intrinsics, housing parameters, and extrinsics. The errors introduced by using the perspective approximation can be eliminated by calibrating refractively, for which the next section will show results.

4.3.4 Refractive Housing Calibration

In this section, it is assumed that the intrinsic camera parameters are known, i. e., have been calibrated using the perspective method on images captured in air. With known indices of refraction and glass thickness, this allows to calibrate flat port underwater housings.

Synthetic Data

Note that the results presented here have already been published in [JSK12]. In order to evaluate the accuracy of the proposed method, synthetic checkerboard images were rendered. In contrast to the calibration

experiments using the perspective camera model, the robustness against noise in the images is more interesting in this case, because of the AbyS approach. Therefore, a different set of rendered checkerboard images is used, where several configurations with increasing image noise were rendered. The intrinsic camera parameters, extrinsic stereo rig transform, and checkerboard where chosen as above. For five noise levels $\sigma \in \{0, 6.4, 12.8, 19.1, 25.5\}$, eight sets of images were rendered, i. e., normal distributed noise was added to the image color values which are in the interval $[0, 255]$. The housing configuration was the same for all trials, in order to measure robustness: the interface distance was $d = 10\,\text{mm}$, the interface thickness $d_g = 50\,\text{mm}$, and the interface tilt was set to $\theta_1 = 30^\circ$ and $\theta_2 = 0.5^\circ$. The main advantage of the AbyS-approach is its independence of errors in corner detection. Therefore, this time, automatically detected checkerboard corners were used, which are always slightly erroneous. The results are plotted in Figure 4.9. Note that the normal error depicted is the angle between the true normal and the computed normal.

In this refractive calibration approach, CMA-ES was used for non-linear optimization (Section A.4.2). Over time, hundreds of different generations of individuals are tested and allow learning the covariance matrix for all parameters, hence, over time, not only the parameters improve, but the algorithm also learns pair-wise correlations between parameters. In Figure 4.10, this adaptation process can be observed on the example of interface distance and camera translation in z-direction. Shortly after generation 1000, the uncertainty of parameters within the generation increases suddenly, i. e., CMA-ES suddenly tried a greater variety of values for both parameters, which then leads to a quick drop in the absolute errors, and hence a correlation was recognized and successfully dealt with. Other possible parameter correlations detected in the proposed method for housing calibration were between camera rotation around the x-axis and the interface normal in y-direction and the camera rotation around the y-axis and the interface normal in x-direction.

Real Data

The method was tested on images of a camera in a controlled lab scenario and on images captured with a camera used on the ROV Kiel 6000. In

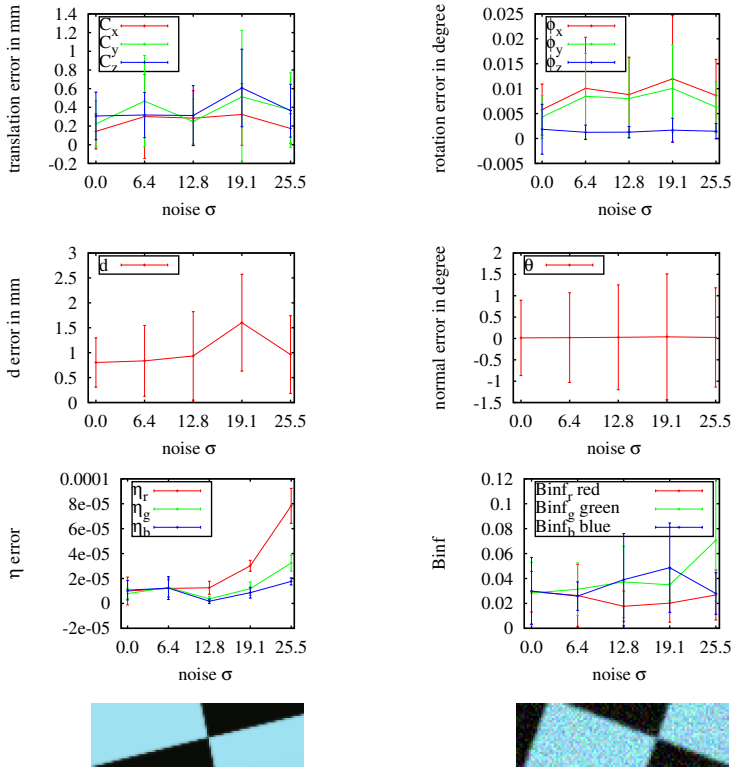

Figure 4.9. Synthetic images, accuracy of the different parameters in presence of growing noise in images. For rendered image intensity values $I \in [0, 255]$, normal distributed noise was added: $I_N = I + N(0, \sigma)$, with a cut-off for $I_N < 0$ or $I_N > 255$. B_∞ (B_{inf}) is the veiling light color. At the bottom: exemplary image cut-outs with zero noise (left) and highest noise level (right). The checkerboards were between 1000 mm and 4000 mm away from the camera. Data previously published in [JSK12].

the first scenario, a tank (500 mm × 1000 mm × 500 mm) was filled with water. A stereo camera rig was placed in front of the tank, allowing to simulate different camera housing interface configurations by moving the camera rig backwards or tilting the rig with respect to the interface.

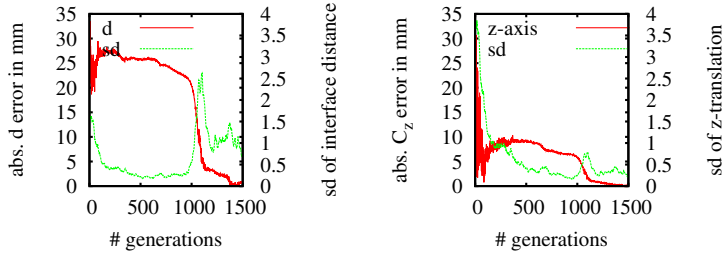

Figure 4.10. Estimation path for an exemplary camera. Left: interface distance error (solid line) and standard deviation (sd) from CMA-ES adaptation (dashed line). Right: camera translation error in z-direction (solid line) and standard deviation from CMA-ES adaptation (dashed line). Data previously published in [JSK12].

Note that seven different configurations were tested. Six of these settings were captured using the stereo rig, one only using a monocular camera (Figure 4.11).

The complete results of calibrating the intrinsics in air and of calibrating perspectively on underwater images are found in Tables A.3 and A.4. Table 4.1 shows the results of calibrating refractively, i. e., calibrating the interface distance d and the interface tilt $\tilde{n}$. a) to g) denote the seven different camera-interface settings. In case b) only one camera was used for capturing images, in all other cases both cameras were used for calibration alone, then the rig was calibrated. The true baselines between the two rig cameras were about $(-60\,\text{mm}, 0\,\text{mm}, 0\,\text{mm})^{\mathrm{T}}$ in cases c), e), f), and g) and about $(-50\,\text{mm}, 0\,\text{mm}, 0\,\text{mm})^{\mathrm{T}}$ in cases a) and d). Since the settings are real-data settings, no ground truth is known. However, the results of calibrating interface distance and tilt of both cameras alone and in the stereo rig should be similar, thus comparing the results allows to examine calibration stability. In Table 4.1, it can be seen that the differences between monocular and stereo calibrations indicate that correlations between interface distance and camera translation in z-direction sometimes cause challenges, however, the overall accuracy is good. The angle θ_2 (compare to Figure 3.12) can be estimated with high accuracy. Note that the angle θ_1 cannot be estimated in case of the angle θ_2 being close to zero. This can

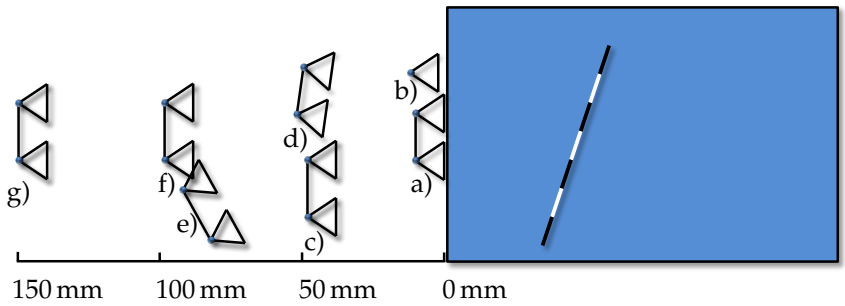

Figure 4.11. Calibration setting. On the right is a water-filled tank in which the checkerboard is moved. The cameras were always placed at the short side of the tank in order to allow maximum depth deviation. The different camera-interface settings are denoted with a) to g). Note that only interface distance and tilt are depicted correctly.

be seen in the results. However, in case of stronger interface tilts (cases d and e), θ_1 can be estimated accurately. A new conclusion from examining the results of perspective calibration on underwater data on the synthetic data in Section 4.3.1 was that the principal point absorbs interface tilts. Table 4.1 shows that in cases d) and e) the interface was tilted strongly. The perspective calibration results show an equally strong shift of the principal point. In case d) the principal point moved about 90 px in x-direction, while in case e) it moved about -201 px in x-direction. This is in direct accordance to the calibrated interface tilts.

In the second scenario, a camera was calibrated based on images captured in a pool with a strong blue hue (Figure 4.14, left). The estimated interface distance in this case was 74 mm, the angle θ between optical axis and interface normal was estimated to be 1.39°.

4.3.5 Calibrating Underwater Light Propagation

As seen in Section 4.2, the model for underwater light propagation can be calibrated after the geometric calibration. This is due to known black and white parts on the checkerboard images and the known camera poses from calibration. The calibration routine is tested on synthetic data by creating

Table 4.1. Calibration results for refractive calibration. Shown are results for seven different camera-interface configurations as depicted in Figure 4.11. With exception of b), in all scenarios checkerboard images were captured using a stereo rig, and calibration was run on both cameras alone and on the rigidly coupled rig. Therefore, results are shown for camera 1 and camera 2, but also for both cameras in the rig.

scenario	d in mm	θ_2 in $^\circ$	θ_1 in $^\circ$	C_{rig} in mm
a) camera 1	7.88	0.34	-23.41	
a) camera 2	8.07	0.29	25.06	
a) rig	19.38	0.27	81.39	
	0.72	0.28	75.22	$(-49.40, 0.39, 0.00)^{\mathrm{T}}$
b) camera 1	10.60	0.25	-30.64	
c) camera 1	51.95	0.29	-16.77	
c) camera 2	44.47	0.23	20.35	
c) rig	48.07	0.11	-43.71	
	58.82	0.42	-73.31	$(-60.02, 0.24, 0.00)^{\mathrm{T}}$
d) camera 1	50.53	**8.06**	178.77	
d) camera 2	47.30	**7.97**	178.45	
d) rig	45.30	**7.99**	178.32	
	28.34	**7.79**	180.33	$(-48.89, -0.76, -0.00)^{\mathrm{T}}$
e) camera 1	86.25	**29.29**	-0.74	
e) camera 2	89.67	**29.16**	0.01	
e) rig	79.96	**28.40**	-1.08	
	102.81	**27.79**	-0.26	$(-59.92, 0.02, 0.00)^{\mathrm{T}}$
f) camera 1	95.54	0.12	-82.37	
f) camera 2	99.60	0.29	77.77	
f) rig	100.39	0.22	-37.90	
	113.16	0.89	-44.37	$(-60.92, 0.30, -0.02)^{\mathrm{T}}$
g) camera 1	149.97	0.12	-46.54	
g) camera 2	150.0	0.39	-33.99	
g) rig	147.13	0.05	70.38	
	160.47	0.09	-6.06	$(-59.85, 0.60, -0.00)^{\mathrm{T}}$

a set of random distances for which black and white underwater colors are determined. Normal distributed noise with $\sigma = 5$ is added to the eight-bit image colors corresponding to 2 % noise. Then, the linear initialization and the non-linear optimization routine are applied to the underwater colors utilizing the known camera-object distances to determine η, B_∞, α,

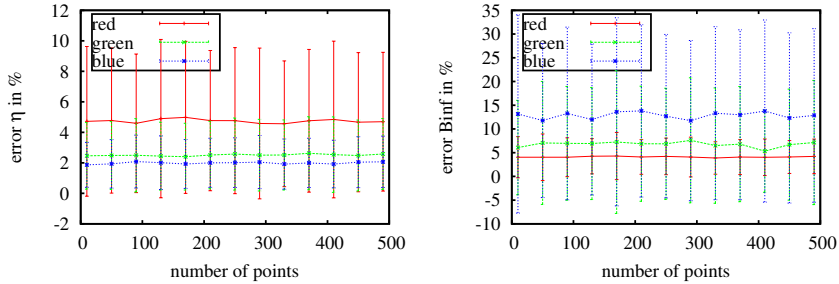

Figure 4.12. Estimation of parameters depending on the number of points. Left: error when estimating η, right: error when estimating the veiling light color B_∞ (B_{inf}).

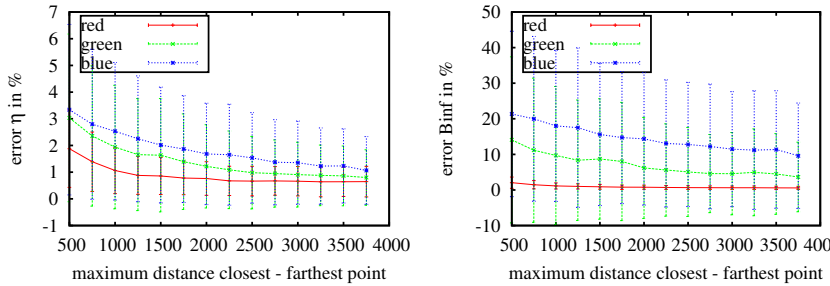

Figure 4.13. Estimation of parameters depending on distance between closest and farthest point. Left: error when estimating η, right: error when estimating the veiling light color B_∞ (B_{inf}).

and β for all three color channels. Figures 4.12 and 4.13 show the results. Two test scenarios are depicted. In Figure 4.12 the number of points has been changed, testing the robustness, when only very few points have been used. The left image shows the results of estimating η, the right image the results of estimating the veiling light color. As can be seen, the estimation has been fairly stable, even for 20 points.

In the second test case (Figure 4.13), the distance deviation between the closest and farthest point has been varied. In general, the closest point has always been 1 m away from the camera centers. The maximum distance

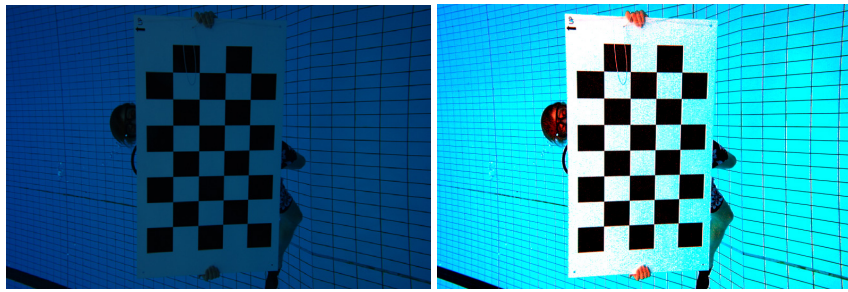

Figure 4.14. Left: original color image of checkerboard with strong green hue. Right: corrected colors of the same image. Note that only the colors on the checkerboard are corrected because they lie on the xy-plane of the world coordinate system for which the camera-object distance is known after calibration.

has been varied between 1.5 m and 5 m causing a difference between 0.5 m and 4 m as depicted in the figures. Figure 4.13 shows that the estimation of η (left image) and the estimation of the veiling light color (right image) generally become more stable with growing distance between the points considered. This is not surprising, since increasing distance differences lead to increasing differences in color, therefore resulting in a more stable estimation of the physical parameters involved.

Figure 4.14 shows an exemplary input image on the left and the resulting color corrected image on the right for the camera calibrated in the pool mentioned above. The colors on the checkerboard are corrected with the estimated distance between checkerboard and camera. Near the bottom of the checkerboard, the red channel cannot be corrected due to the red light being almost completely absorbed by the water. Figure 4.15 shows that the color restoration also works in a water body with strong green hue, like the Baltic sea. In this case, the method was also used to correct texture colors, as can be seen in Figure 6.5.

4.4 Summary

In this chapter, methods for calibrating the intrinsics of a perspective camera and for calibrating a flat port underwater housing were presented.

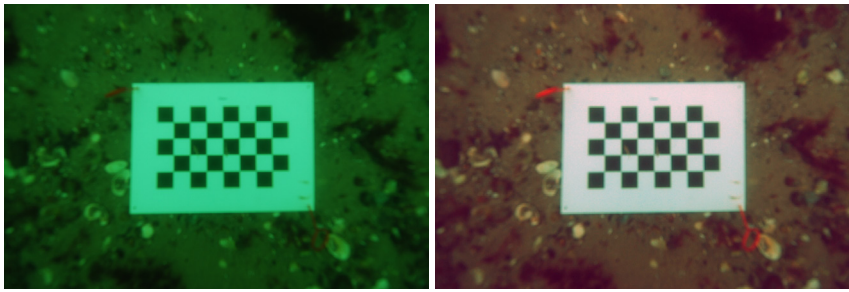

Figure 4.15. Left: checkerboard with with strong green hue captured in the Baltic Sea. Right: checkerboard and seafloor with corrected colors. Input image by Florian Huber.

Additionally, both methods can calibrate relative rig extrinsics in case more than one camera is used in a rigidly coupled camera rig. Experiments were conducted for two scenarios. First, a set of underwater images was rendered in order to investigate the systematic model error introduced when calibrating the perspective model on underwater images. This yielded the conclusion that all intrinsic parameters absorb part of the model error depending on interface distance and tilt. Additionally, a part of the error is compensated by the camera pose. Secondly, the proposed refractive approach was tested on synthetic and real images. Here, it was important to show the approach's invariance against image noise and errors in corner detection. The proposed method was shown to work accurately. Finally, results for calibrating the model for underwater light propagation were shown.

Utilizing the gained calibrations of intrinsics and housing parameters, the next chapter will propose all necessary components for refractive SfM and a refractive Plane Sweep algorithm.

Chapter 5

Structure-from-Motion and Multi-View-Stereo

When using camera models like the perspective model with lens distortion, it is possible to extract the implicitly contained geometric information from a set of 2D images of a scene. This can either mean the computation of a panorama, where the camera is only allowed to rotate around its center of projection, or the complete 3D reconstruction of a rigid object or scene [Sze11, HZ04]. In the latter case, the camera is required to move, i. e., translate as well as rotate. Algorithms for reconstruction are usually classified as Structure-from-Motion algorithms (SfM), where not only the camera path, but also a sparse 3D point cloud is recovered. The first section of the chapter will introduce the geometric concepts and algorithms used in SfM methods. In order to retrieve a dense model instead of a sparse 3D point cloud, multi-view stereo (MVS) is applied to the images once the camera poses are known. The second section of the chapter will therefore cover multi-view stereo and 3D model computation. Refer to Figure 5.1 for an overview of this chapter's structure.

5.1 Structure-from-Motion

SfM algorithms for use on images captured in air are well researched (for overviews refer to [Sze11, HZ04]). However, as seen in Section 3.2.1, the perspective camera model is invalid in underwater scenarios due to refraction and causes a systematic model error, especially in sequential approaches, where an error made at the beginning of the sequence can easily accumulate to a considerable measuring error. In the previous chapter, it

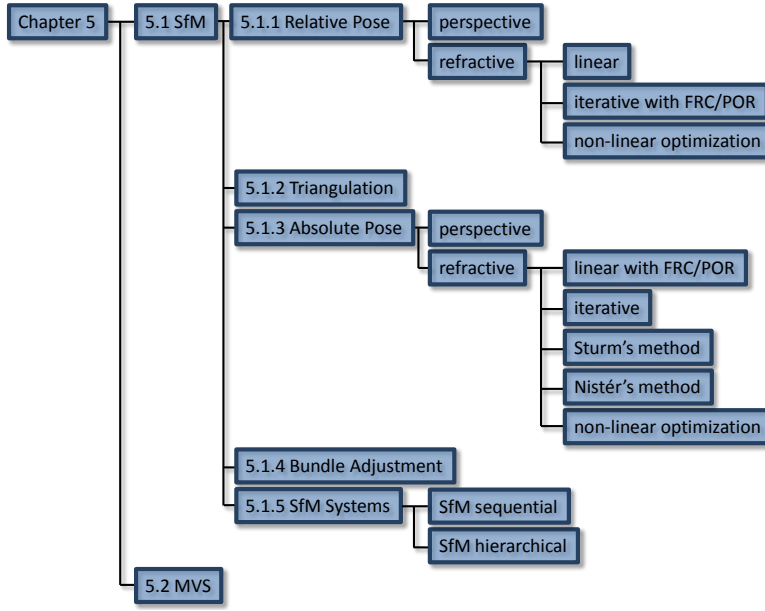

Figure 5.1. Overview Chapter 5.

was demonstrated how the underwater housing of a camera can be calibrated. The resulting information is now going to be utilized to develop a Structure-from-Motion approach, which explicitly models refraction, hence eliminating the systematic model error that can be observed when reconstructing underwater scenes with a perspective approach (parts on the method have been previously published in [JSK13]). Figure 5.2 shows some exemplary input images on the left and the SfM result, the retrieved camera path and sparse 3D point cloud on the right. The ultimate goal is a refractive SfM approach that self-calibrates the underwater housing assuming the camera's intrinsics to be calibrated beforehand, thus eliminating the need of using a checkerboard below water, which is at best impractical and often infeasible. In order to model refraction explicitly, the nSVP nature of the refractive camera requires the use of more general methods of geometry estimation compared to established methods using

Figure 5.2. Left: exemplary input images of input sequence. Right: camera path and sparse 3D point cloud of object resulting from SfM algorithm. Input images by Christian Howe.

the perspective camera model. Some approaches for other camera models, e. g., multi-view camera rigs or catadioptic cameras have already been introduced in [Ple03, LHK08, SRL06], where usually a ray is computed for each pixel and then used for geometry estimation, However, as can be seen in Table 3.10, no complete system for refractive SfM has been introduced and tested on real data and compared to perspective SfM. Therefore, this section will introduce geometry estimation for refractive and perspective cameras and will compare the performance of different methods.

5.1.1 Relative Pose

Perspective Relative Pose

A scene imaged twice by the same camera from different poses contains implicit geometric information in the images. This geometric information can be exploited by determining 2D-2D point correspondences between the images. A multitude of methods exists for the detection and matching of features [TM08]. In this thesis, SIFT features, introduced by Lowe [Low04] are utilized, or more concretely [Wu07], a fast implementation with great matching accuracy delivers key point matches, which are used in the following algorithms for geometry estimation. Figure 5.3 depicts such a situation. The first image on the left has a point $\mathbf{x}$ in the image and the camera center C. It is clear, that the ray from the camera center

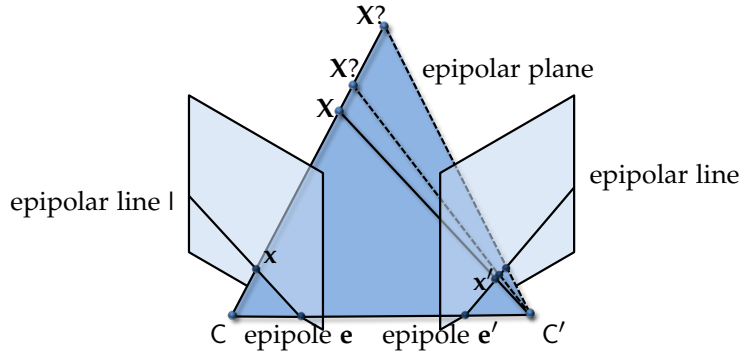

Figure 5.3. Epipolar geometry. The epipoles $\mathbf{e}$ and $\mathbf{e}'$ are computed by projecting the camera centers C' and C respectively and the ray through $\mathbf{x}$ and the baseline connecting the camera centers forms the epipolar plane. The projections of all possible points $\mathbf{X}$ form the epipolar line in the second image.

through $\mathbf{x}$ must contain the corresponding 3D point $\mathbf{X}$ that was imaged onto $\mathbf{x}$. Unfortunately, prior to determining any 3D structure, it is unclear where on the ray the exact 3D point lies. It is possible, however, to image the whole ray in the second image, which corresponds to the line, l' in Figure 5.3. Somewhere on this epipolar line, the corresponding point $\mathbf{x}'$ must lie in the second image. The plane spanned by both camera centers and the 3D point is called epipolar plane. The projections of the other respective camera center into the images are called epipoles. Since all rays intersect the camera center of one camera, all epipolar lines intersect in the epipole of the other image. Without loss of generality, the first camera is set into the coordinate system origin, thus the second camera's pose $[\mathbf{R}^{\mathrm{T}}|-\mathbf{R}^{\mathrm{T}}\mathsf{C}]$ is determined relative to the first camera's pose. In order to compute the epipolar line l' in the second image, the epipole $\mathbf{e}'$ is constructed by projecting the first camera center $(0,0,0,1)^{\mathrm{T}}$ into the second camera such that $\mathbf{e}' = \mathbf{K}'[\mathbf{R}^{\mathrm{T}} - \mathbf{R}^{\mathrm{T}}\mathsf{C}](0,0,0,1)^{\mathrm{T}}$. A second point on l' is determined by the infinite homography that maps the point via the plane at infinity [HZ04] into the second image. l' is then determined by

the cross product:

$$l' = \left[\mathbf{e}'\right]_\times \mathbf{x}'_\infty = \left[\mathbf{e}'\right]_\times \underbrace{\mathbf{K}'\mathbf{R}^T\mathbf{K}^{-1}}_{\mathbf{H}_\infty}\mathbf{x}, \tag{5.1.1}$$

which defines the Fundamental matrix $\mathbf{F}^T = [\mathbf{e}']_\times \mathbf{K}'\mathbf{R}^T\mathbf{K}^{-1}$. $\mathbf{F}$ describes the mapping in Figure 5.3. In general, $\mathbf{F}$ is a 3×3 up-to-scale matrix with rank two ($[\mathbf{e}']_\times$ has rank two), which has seven degrees of freedom (eight ratios, and $\det(\mathbf{F}) = 0$).

In this thesis, however, it will in general be assumed, that the intrinsic parameters of the cameras are known. The result is that all 2D-2D point correspondences can be normalized using the camera matrices: $\mathbf{x}_n = \mathbf{K}^{-1}\mathbf{x}$ and $\mathbf{x}'_n = \mathbf{K}'^{-1}\mathbf{x}'$. Using these correspondences in epipolar geometry and assuming the first camera to be located at the origin, this yields the Essential matrix $\mathbf{E}$ instead of the fundamental matrix. The relation between the two is:

$$\mathbf{E} = \mathbf{K}^T\mathbf{F}\mathbf{K}' = [C]_\times \mathbf{R} \tag{5.1.2}$$

assuming that $\mathbf{K}$ is the camera matrix of the first image, and $\mathbf{K}'$ is the camera matrix of the second image. Note that the essential matrix has only five degrees of freedom, three for the rotation between the two cameras and two for the baseline, which has no determined scale. This causes constraints on the singular values of the Essential matrix: one singular value is zero, while the other two are equal. Using $\mathbf{E}$, this yields the following relations between correspondences:

$$\mathbf{x}_n^T\mathbf{E}\mathbf{x}'_n = 0 \quad \text{(epipolar constraint)} \tag{5.1.3}$$

$$\mathbf{E}\mathbf{x}'_n = l \quad \text{(epipolar line first image)} \tag{5.1.4}$$

$$\mathbf{E}^T\mathbf{x}_n = l' \quad \text{(epipolar line second image).} \tag{5.1.5}$$

In case of computing reconstructions, one usually has a set of images captured from different points of view. A pair of two such pictures can now be brought into relation using the epipolar geometry in order to determine the camera poses of the images. Assuming calibrated cameras, a normalized set of K 2D-2D correspondences is matched between the two images and each correspondence $k \in \{1, ..., K\}$ yields one constraint for

estimating the Essential matrix:

$$\mathbf{x}_{\mathrm{n}_k}^{\mathrm{T}} \mathbf{E} \mathbf{x'}_{\mathrm{n}_k} = 0 \quad k \in \{1, ..., K\}. \tag{5.1.6}$$

There are several algorithms for computing the essential matrix. The linear Eight-Point algorithm uses a set of at least eight correspondences [HZ04]. In order to achieve that, Equation 5.1.6 is expanded and a set of eight equations, i. e., one for each correspondence, is used to stack a linear system of equations $\mathbf{A}x = 0$, with x being the vector with the unknown elements of $\mathbf{E}$, which is solved using the SVD (Section A.3) . Note that one correspondence is required for each of the eight up-to-scale entries, but the essential matrix does not have full rank. Therefore, the constraints on the singular values need to be enforced after solving for the entries of $\mathbf{E}$ using the SVD. In order to do that, the SVD is applied to $\mathbf{E} = \mathbf{U}\mathrm{diag}(\mathsf{S})\mathbf{V}^{\mathrm{T}}$ and $\mathsf{S} = (a, b, c)^{\mathrm{T}}$ is changed as follows [HZ04]:

$$\mathsf{S}' = \left(\frac{a+b}{2}, \frac{a+b}{2}, 0 \right)^{\mathrm{T}}. \tag{5.1.7}$$

The least squares approximation $\hat{\mathbf{E}}$ is computed by:

$$\hat{\mathbf{E}} = \mathbf{U}\mathrm{diag}(\mathsf{S}')\mathbf{V}^{\mathrm{T}}. \tag{5.1.8}$$

The described approach is simple to implement and easy to understand, however, eight correspondences are required. There are other ways of estimating the Essential matrix, which allow to use the theoretical minimum of five correspondences for computing $\mathbf{E}$ [Nis04, BBD08]. Consequently, the null space of matrix $\mathbf{A}$ used for solving the equation system has more than one dimension. Constraints on the rotation $\mathbf{R}$ are used to find a set of several possible solutions, from which the correct one needs to be chosen.

Refractive Relative Pose using Linear Estimation

When considering the relation between two views distorted by refraction, the classic epipolar geometry needs to be generalized. In case of a set of underwater cameras, the epipolar lines turn into curves [Maa92, CS09, GGY11] (Figure 5.4). Instead of relating points in the first image to epipolar

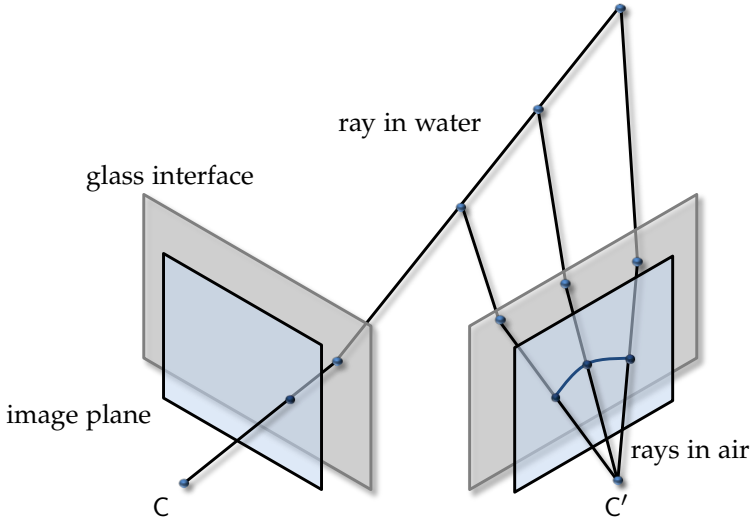

Figure 5.4. Generalized epipolar geometry with (quartic) epipolar curve in underwater case (after [Maa92]).

lines in the second image, an approach described by Pless [Ple03] utilizes the intersection between Plücker lines corresponding to the rays in water computed for both corresponding image points. The preceding chapters already showed, that for a given point $\mathsf{x} = (x, y) \in \mathbb{R}^2$ in one image, the Plücker line can be determined if the intrinsic parameters and the housing parameters are known. Let $\mathsf{L}_k = (\tilde{\mathsf{X}}_{\mathsf{w}_k}, \mathsf{M}_k)$ and $\mathsf{L}'_k = (\tilde{\mathsf{X}}'_{\mathsf{w}_k}, \mathsf{M}'_k)$ be the Plücker lines for two image points of the k^{th} correspondence, with $\tilde{\mathsf{X}}_{\mathsf{w}_k}$ denoting the line's direction in water and $\mathsf{M}_k = \tilde{\mathsf{X}}_{\mathsf{w}_k} \times \mathsf{X}_{\mathsf{s}_k}$ denoting the line's moment. Note that both lines are still in the local camera coordinate system. As in case of perspective epipolar geometry, the first camera is assumed to be in the world coordinate system origin.

In order to determine the relative camera pose of the second camera, $\mathbf{R}$ and C need to be determined, such that the line in the second local camera coordinate system is transformed into the world coordinate system [Ple03]:

$$\mathsf{L}'^{\mathsf{wc}}_k = (\mathbf{R}\tilde{\mathsf{X}}'_{\mathsf{w}_k}, \mathbf{R}\mathsf{M}'_k - [\mathsf{C}]_\times \mathbf{R}\tilde{\mathsf{X}}'_{\mathsf{w}_k}) \tag{5.1.9}$$

The intersection between both lines is determined by:

$$\tilde{\mathsf{X}}_{\mathsf{w}_k}^{\mathsf{T}} \mathsf{M}'^{\mathsf{wc}}_k + \mathsf{M}_k^{\mathsf{T}} \tilde{\mathsf{X}}'^{\mathsf{wc}}_{\mathsf{w}_k} = 0 \tag{5.1.10}$$

Equations 5.1.9 and 5.1.10 combined yield:

$$\begin{aligned} 0 =& \tilde{\mathsf{X}}_{\mathsf{w}_k}^{\mathsf{T}} (\mathbf{R} \mathsf{M}'_k - [\mathsf{C}]_\times \mathbf{R} \tilde{\mathsf{X}}'_{\mathsf{w}_k}) + \mathsf{M}_k^{\mathsf{T}} (\mathbf{R} \tilde{\mathsf{X}}'_{\mathsf{w}_k}) \\ =& \begin{pmatrix} \tilde{\mathsf{X}}_{\mathsf{w}_k} \\ \mathsf{M}_k \end{pmatrix}^{\mathsf{T}} \underbrace{\begin{pmatrix} -[\mathsf{C}]_\times \mathbf{R} & \mathbf{R} \\ \mathbf{R} & \mathbf{0}_{3\times 3} \end{pmatrix}}_{\mathbf{E}_{\mathrm{GEC}}} \begin{pmatrix} \tilde{\mathsf{X}}'_{\mathsf{w}_k} \\ \mathsf{M}'_k \end{pmatrix} \\ =& \mathsf{L}_k^{\mathsf{T}} \mathbf{E}_{\mathrm{GEC}} \mathsf{L}'_k, \end{aligned} \tag{5.1.11}$$

where $\mathbf{E}_{\mathrm{GEC}}$ is the generalized Essential matrix and the equation is called the Generalized Epipolar constraint (GEC)[Ple03]. Expanding this term yields an equation with the unknown entries of $\mathbf{E}_{\mathrm{GEC}}$ for the k^{th} point correspondence:

$$\mathsf{L}_k^{\mathsf{T}} \mathbf{E}_{\mathrm{GEC}} \mathsf{L}'_k = \mathsf{L}_k^{\mathsf{T}} \begin{pmatrix} e_{11} & e_{12} & e_{13} & r_{11} & r_{12} & r_{13} \\ e_{21} & e_{22} & e_{23} & r_{21} & r_{22} & r_{23} \\ e_{31} & e_{32} & e_{33} & r_{31} & r_{32} & r_{33} \\ r_{11} & r_{12} & r_{13} & 0 & 0 & 0 \\ r_{21} & r_{22} & r_{23} & 0 & 0 & 0 \\ r_{31} & r_{32} & r_{33} & 0 & 0 & 0 \end{pmatrix} \mathsf{L}'_k. \tag{5.1.12}$$

Table A.5 shows the resulting coefficients for each entry in $\mathbf{E}_{\mathrm{GEC}}$. At this point, at least 17 equations with those coefficients from different correspondences are used to build a matrix $\mathbf{A}_{\mathrm{Pless}}$ such that the equation system $\mathbf{A}_{\mathrm{Pless}} \mathsf{x} = 0$ can be solved using the SVD in order to determine the entries of $\mathbf{E}_{\mathrm{GEC}}$. x is the vector containing all the variables as in Table A.5. The SVD solution is determined by finding the closest solution subject to $||\mathsf{x}|| = 1$, and the method will be called Pless-method in the remainder of the thesis. In presence of noise in the correspondences, the necessary constraints will need to be enforced on the resulting rotation matrix, i. e., the eigenvalues need to be set to one, as well as the determinant.

However, Li, et al. argue in [LHK08] that mainly due to the missing constraints of the two rotation matrices involved being identical, this

method does not work well in practical applications. [LHK08] propose a more robust method, where instead of determining the solution subject to $||\mathsf{x}|| = 1$, the solution subject to $||(e_{11}, ..., e_{33})^{\mathrm{T}}|| = 1$ is computed. This is due to the matrix $\mathbf{A}$ often having one or more singular values close to zero depending on the underlying ray configuration as depicted in Figure 2.6. As described in the article, two matrices $\mathbf{A}_{\mathrm{E}}$ and $\mathbf{A}_{\mathrm{R}}$ need to be determined, which is matrix $\mathbf{A}$ split according to whether the entries are constraints on the variables $e_{11}, ..., e_{33}$ or $r_{11}, ..., r_{33}$. With $\mathbf{A}_{\mathrm{R}}^{+}$ denoting the pseudo inverse, the SVD is used to solve

$$(\mathbf{A}_{\mathrm{R}}\mathbf{A}_{\mathrm{R}}^{+} - \mathbf{I})\mathbf{A}_{\mathrm{E}} \begin{pmatrix} e_{11} \\ ... \\ e_{33} \end{pmatrix} = 0 \tag{5.1.13}$$

This yields entries similar to the classic essential matrix. However, the decomposition needs to account for the more general problem: as in case of the classic essential matrix, two differing rotation matrices $\mathbf{R}$ can be determined. For both of these rotation matrices an equation system is used to solve for the entries in C using the GEC (5.1.11):

$$\tilde{\mathsf{X}}_{\mathrm{w}_k}^{\mathrm{T}} \left[\mathsf{C}\right]_{\times} \mathbf{R}\tilde{\mathsf{X}'}_{\mathrm{w}_k} = \tilde{\mathsf{X}}_{\mathrm{w}_k}^{\mathrm{T}} \mathbf{R}\mathsf{M}'_k + \mathsf{M}_k^{\mathrm{T}}\mathbf{R}\tilde{\mathsf{X}'}_{\mathrm{w}_k}, \tag{5.1.14}$$

which yields coefficients for the unknowns in C as in Table A.6. The computation of the linear least squares solution $\mathbf{A}_{\mathsf{C}}\mathsf{C} = \mathsf{b}$ for both matrices $\mathbf{R}$ yields two solutions for C, the one with the smaller residuum $r = ||\mathbf{A}_{\mathsf{C}}\mathsf{C} - \mathsf{b}||$ is chosen. This method will be called Li-method in the following. Note that in theory the overall scene scale is already encoded in the rays being determined by the camera calibration, so there are six degrees of freedom. Due to instabilities, the authors of Li et al. [LHK08] propose to alternately compute the rotation and translation, thus creating a method that converges quickly.

When applying the above described Pless-method to underwater images captured through a flat port, i. e., an axial camera, [LHK08] found the number of non-zero singular values to be 16, and hence applying the Pless-method yields a two-dimensional solution space spanned by the vectors x_1 and x_2 corresponding to the zero-singular values. Consequently,

all possible solutions can be described by:

$$\mathsf{x} = \mu \mathsf{x}_1 + \nu \mathsf{x}_2 \quad \mu, \nu \in \mathbb{R}, \tag{5.1.15}$$

where μ can be set to one due to the solution being up to scale. In order to find the correct solution, i. e., the correct ν, the part of the solution vectors containing the entries for the Essential matrix $\mathbf{E}_{\mathrm{GEC}}$ are utilized. It is known, that the determinant of $\mathbf{E}_{\mathrm{GEC}}$ is zero. Thus,

$$\det(\mu \mathsf{x}_{\mathbf{E}_1} + \nu \mathsf{x}_{\mathbf{E}_2}) = 0 \quad \mu = 1, \nu \in \mathbb{R}, \tag{5.1.16}$$

where ν can be determined by solving the corresponding 3^{rd}-degree polynomial. This method will be used in the following experiments. Note that the minimal number of correspondences depends on the actual underlying ray configuration and is 17 for a general ray configuration and 16 for axial cameras, and thus the considered flat port underwater camera.

Refractive Iterative Computation

An approach tailored specifically to the underwater camera, can be derived using the Plane of Refraction equation (POR) and the Flat Refractive constraint (FRC) that have been proposed by Agrawal et al. in [ARTC12]. Starting from a set of K 2D-2D correspondences between two images, the first camera is again set into the origin of the world coordinate system. The rotation $\mathbf{R}$ and translation C of the second camera relative to the first are computed. Two constraints are used for determining $\mathbf{R}$ and C for each correspondence $k \in \{1, ..., K\}$:

$$\begin{aligned} \mathsf{X}_{\mathsf{s}_k} + \kappa_k \tilde{\mathsf{X}}_{\mathsf{w}_k} &= \mathbf{R}\mathsf{X'}_{\mathsf{s}_k} + \mathsf{C} + \kappa'_k \mathbf{R}\tilde{\mathsf{X}}'_{\mathsf{w}_k} \quad \kappa_k, \kappa'_k \in \mathbb{R} \\ \text{and} \qquad (\mathbf{R}\mathsf{X'}_{\mathsf{s}_k} &+ \mathsf{C} + \kappa'_k \mathbf{R}\tilde{\mathsf{X}}'_{\mathsf{w}_k} - \mathsf{X}_{\mathsf{s}_k}) \times \tilde{\mathsf{X}}_{\mathsf{w}_k} = 0, \end{aligned} \tag{5.1.17}$$

where the first one is the triangulation constraint that describes the triangulation of the unknown 3D point in which both rays intersect. κ_k and κ'_k are scaling factors for the rays in water such that both rays intersect in their common 3D point. The second constraint is the FRC (3.2.17), where the unknown 3D point is parametrized by $\mathbf{R}\mathsf{X'}_{\mathsf{s}_k} + \mathsf{C} + \kappa'_k \mathbf{R}\tilde{\mathsf{X}}'_{\mathsf{w}_k}$. In both constraints, the unknowns are $\mathbf{R}$, C, and κ_k and κ'_k for all $k \in \{1, ..., K\}$, thus,

both Equations (5.1.17) are non-linear in the unknowns. Consequently, an iterative approach is applied to solve for the unknowns $\mathbf{R}$ and C, by solving the equation system resulting of stacking Equations (5.1.17) for all correspondences (refer to Table A.7). Note that of the six resulting equations, three in linearly independent. This determines the minimum number of required correspondences to be four, however, in order to increase robustness, we use six. Then, using the updated $\mathbf{R}$ and C, κ_k and κ'_k for all $k \in \{1, ..., K\}$ are updated by solving for κ_k and κ'_k in a constraint based on the POR (3.2.18):

$$\begin{aligned}(\mathbf{R}\mathsf{X}'_{\mathsf{s}_k} + \mathsf{C} + \kappa'_k \mathbf{R}\tilde{\mathsf{X}}'_{\mathsf{w}_k})^{\mathrm{T}}(\tilde{\mathsf{n}} \times \tilde{\mathsf{X}}_{\mathsf{w}_k}) = 0 \\ (\mathbf{R}^{\mathrm{T}}(\mathsf{X}_{\mathsf{s}_k} + \kappa_k \tilde{\mathsf{X}}_{\mathsf{w}_k}) - \mathbf{R}^{\mathrm{T}}\mathsf{C})^{\mathrm{T}}(\mathsf{n} \times \tilde{\mathsf{X}}'_{\mathsf{w}_k}) = 0.\end{aligned} \tag{5.1.18}$$

Both described linear and iterative approaches allow to determine an initial estimate of the pose of the second camera. However, neither method is very good in terms of absolute accuracy in presence of noise on the 2D-2D-correspondences. Hence, a non-linear optimization step is necessary to improve the initial estimate.

Scene Scale

Theoretically, the absolute scale of the scene can be estimated by the relative pose problem for refractive underwater cameras as opposed to the perspective relative pose problem, where the baseline between the two views is usually set to one. This is due to the rays starting on the outer interface being metric. Note that in case of perspective reconstructions, a scene can be scaled consistently by applying the scaling transform $\mathbf{T}$ to all projections and the inverse $\mathbf{T}^{-1}$ to all 3D points. However, in case of a refractive reconstruction, interface distance and thickness would need to be scaled additionally, thereby changing the starting points and direction of the rays in water. Consequently, the relative pose problem in case of refractive cameras is not invariant to scale changes of the translation vector.

Optimization

In addition to noisy 2D-2D correspondences disturbing the initial pose estimates, the need to enforce constraints on for example the rotation matrix causes the result not to be the best fit. This can be improved by non-linear optimization, therefore, different non-linear optimization functions will be discussed now. The

Reprojection Error is the most commonly used error function in case of perspective SfM:

$$\mathrm{r}_k = \mathrm{f}_{\mathrm{RP}_k} - \check{\mathbf{x}}_k \tag{5.1.19}$$

$$E_{\mathrm{RP}} = \sum_{k=0}^{K} ||\mathrm{f}_{\mathrm{RP}_k} - \check{\mathbf{x}}_k||_2^2. \tag{5.1.20}$$

3D points are projected onto 2D points $\mathbf{x}_k$ by the function $\mathrm{f}_{\mathrm{RP}_k}$ depending on the current camera parameters and the error is the distance to the measured 2D points $\check{\mathbf{x}}_k$. $\mathrm{f}_{\mathrm{RP}_k}$ is the function used in optimization schemes like bundle adjustment or classic Levenberg-Marquardt routines, while E_{RP} is the function that can be optimized by CMA-ES. However, projecting points in the presence of refraction requires solving a 12^{th} degree polynomial (Section 3.2.5). While, this insight is a huge improvement compared to using non-linear optimization in order to determine the projected point, it is still infeasible to use in applications like bundle adjustment [TMHF00].

Angle Error was introduced by Mouragnon et al. [MLD+07, MLD+09] and basically computes the angle between the ray in water corresponding to a 2D image point and the ray corresponding to the 3D point transformed into the camera coordinate system (Figure 5.5). Note that this is basically the arccos of the the FRC:

$$\mathrm{g}_{\mathrm{A}_k} = \arccos\left(\tilde{\mathrm{X}}_{\mathrm{w}_k}^{\mathrm{T}} \frac{\mathrm{X}_k^{\mathrm{cc}} - \mathrm{X}_{\mathrm{s}_k}}{||\mathrm{X}_k^{\mathrm{cc}} - \mathrm{X}_{\mathrm{s}_k}||}\right) \tag{5.1.21}$$

$$E_{\mathrm{A}} = \sum_{k=0}^{K} \left|\mathrm{g}_{\mathrm{A}_k}\right|. \tag{5.1.22}$$

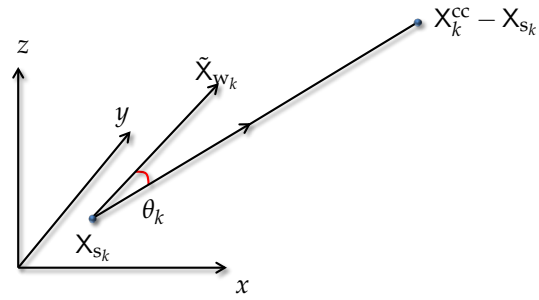

Figure 5.5. In order to compute the angular error θ_k, a 3D point X_k is transformed into the camera coordinate system. For the corresponding image point the ray in water is computed represented by starting point X_{s_k} and direction $\tilde{X}_{w_k}$. To compute θ_k, X_{s_k} is subtracted from X_k^{cc}.

Note that in [MLD+07, MLD+09] for better convergence, coordinate system (5.1.21) is rotated such that $\tilde{X}_k = (0, 0, 1)^T$

GEC Error is the error for all correspondences using Equation (5.1.11):

$$g_{GEC_k} = \tilde{X}_{w_k}^T (\mathbf{R} M'_k - [C]_\times \mathbf{R} \tilde{X}'_{w_k}) + M_k^T (\mathbf{R} \tilde{X}'_{w_k}) \tag{5.1.23}$$

$$E_{GEC} = \sum_{k=0}^{K} |g_{GEC_k}| \tag{5.1.24}$$

Note that the GEC error can only be directly applied in two-view scenarios, i. e., not in applications like non-linear optimization of absolute pose estimation or multi-view bundle adjustment. The GEC error is an algebraic error, where the zero solution is always correct, but also wrong.

POR/FRC Error uses specific characteristics of the flat port underwater camera as described by Agrawal et al. in [ARTC12]. Using the Plane of Refraction constraint to compute κ and κ' as shown in Equation (3.2.18) allows to retrieve a 3D point X for the current relative pose. The Flat Refractive constraint as shown in Equation (3.2.17) then provides an error measurement. This error function also only works in two-

view scenarios.

Virtual Camera Error can be determined in case a 3D point is known or in case a 3D point can be triangulated (refer to Section 5.1.2) in the two view case. For each point $k \in \{1, ..., K\}$, a virtual camera (Figure 5.6) is determined with the camera center $\mathsf{C}_{\mathrm{v}_k}$ lying on the camera axis, i. e., the axis defined by interface normal $\tilde{\mathrm{n}}$ and center of projection, where all rays in water $\tilde{\mathsf{X}}_{\mathrm{w}_k}$ intersect. $\mathsf{C}_{\mathrm{v}_k}$ can be determined by solving for κ_k in $\mathsf{X}_{\mathrm{s}_k} + \kappa_k \tilde{\mathsf{X}}_{\mathrm{w}_k} = \tilde{\mathrm{n}}$. The virtual rotation $\mathbf{R}_{\mathrm{v}_k}$ is defined by a rotation axis and a rotation angle, which are defined by the cross product of interface normal and optical axis and scalar product of interface normal and optical axis respectively. The virtual focal length is set to $f_{\mathrm{v}_k} = d$, thus the image plane is parallel to the outer interface plane. A 3D point X_k can then be transformed into the local coordinate system of the virtual camera $\mathsf{X}_{\mathrm{v}_k}$ by:

$$\mathsf{X}_{\mathrm{l}_k} = \mathbf{R}^{\mathrm{T}} \mathsf{X}_k - \mathbf{R}^{\mathrm{T}} \mathsf{C} \tag{5.1.25}$$

$$\mathsf{X}_{\mathrm{v}_k} = \mathbf{R}_{\mathrm{v}_k}^{\mathrm{T}} \mathsf{X}_{\mathrm{l}_k} - \mathbf{R}_{\mathrm{v}_k}^{\mathrm{T}} \mathsf{C}_{\mathrm{v}_k}. \tag{5.1.26}$$

In order to compute the error, the 2D image point $\mathbf{x}_k$ is transformed into its corresponding ray in water $(\mathsf{X}_{\mathrm{s}_k}^{\mathrm{T}}, \tilde{\mathsf{X}}_{\mathrm{w}_k}^{\mathrm{T}})$ and then transformed into the virtual camera coordinate system as well:

$$\mathsf{X}_{\mathrm{vx}_k} = \mathbf{R}_{\mathrm{v}_k}^{\mathrm{T}} \mathsf{X}_{\mathrm{s}_k} - \mathbf{R}_{\mathrm{v}_k}^{\mathrm{T}} \mathsf{C}_{\mathrm{v}_k}. \tag{5.1.27}$$

Both points $\mathsf{X}_{\mathrm{v}_k}$ and $\mathsf{X}_{\mathrm{vx}_k}$ are then projected into the virtual camera and used to compute the virtual camera error:

$$\mathrm{g}_{\mathrm{v_k}} = \begin{pmatrix} \frac{f_{\mathrm{v}_k}}{\mathsf{X}_{\mathrm{v}_k z}} \mathsf{X}_{\mathrm{v}_k x} - \frac{f_{\mathrm{v}_k}}{\mathsf{X}_{\mathrm{vx}_k z}} \mathsf{X}_{\mathrm{vx}_k x} \\ \frac{f_{\mathrm{v}_k}}{\mathsf{X}_{\mathrm{v}_k z}} \mathsf{X}_{\mathrm{v}_k y} - \frac{f_{\mathrm{v}_k}}{\mathsf{X}_{\mathrm{vx}_k z}} \mathsf{X}_{\mathrm{vx}_k y} \end{pmatrix} \tag{5.1.28}$$

$$E_{\mathrm{v}} = \sum_{k=0}^{K} ||\mathrm{g}_{\mathrm{v_k}}||_2. \tag{5.1.29}$$

A similar, but far more time-consuming interface error was introduced in [SK11a], where the virtual camera center C_{v} was the caustic point

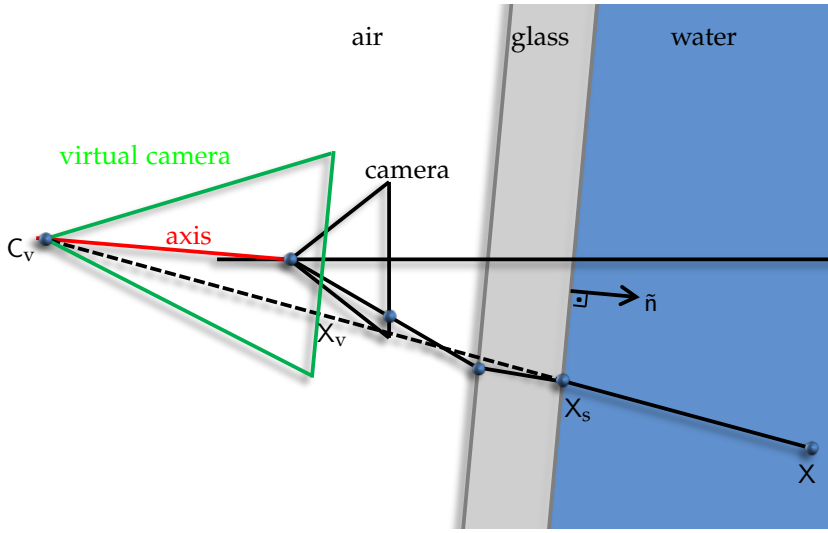

Figure 5.6. Virtual camera definition. The virtual camera center C_v can be found by intersecting the ray in water with the line defined by the camera's center of projection and the interface normal. The rotation $\mathbf{R}_v$ is defined by the interface normal. Note that a 3D point X can be projected in the resulting virtual camera perspectively.

corresponding to the 2D image point, the computation of which was expensive. The insight of the flat port camera being an axial camera eliminates the need to compute the caustic point. The intersection with the axis defined by the camera's center of projection and interface normal can be computed much more efficiently. In addition, a fixed virtual focal length is used, eliminating the strong correlation between interface distance and error that was a problem in [SK11a]. Note that the use of the virtual camera error function allows to compute analytic derivatives of the error function in the direction of the parameters, an advantage compared to the reprojection error and the former version based on caustic points as virtual camera centers.

All of the above described error functions can be used in different configurations, i. e., different optimization frameworks like bundle adjustment, classic Levenberg-Marquardt, CMA-ES, or as outlier detectors within RANSAC (Random Sampling Consensus) frameworks [FB81]. Additionally, different configurations of parameters can be optimized, e. g., housing parameters and extrinsics or extrinsics only. For example in case of using CMA-ES to optimize a relative pose estimate, the function

$$E_{\mathrm{v}} = \sum_{k=0}^{K} ||\mathrm{g}_{\mathrm{v}_k}(\mathsf{C}, \mathbf{R}, \mathsf{X}_{\mathrm{s}_k}, \tilde{\mathsf{X}}_{\mathrm{w}_k}, \mathsf{C}_{\mathrm{v}_k}, \mathsf{X'}_{\mathrm{s}_k}, \tilde{\mathsf{X}}'_{\mathrm{w}_k}, \mathsf{C'}_{\mathrm{v}_k})||_2 \tag{5.1.30}$$

is optimized, where $\mathsf{X}_{\mathrm{s}_k}$, $\tilde{\mathsf{X}}_{\mathrm{w}_k}$, $\mathsf{C}_{\mathrm{v}_k}$, $\mathsf{X'}_{\mathrm{s}_k}$, $\tilde{\mathsf{X}}'_{\mathrm{w}_k}$, and $\mathsf{C'}_{\mathrm{v}_k}$ are observations describing the ray correspondence. By considering the additional rigid rig transform, the reprojection error, angular error, and virtual camera error can easily be extended to incorporate the optimization of rigid multi-camera rigs. Note that computing the reprojection error is infeasible for use in large applications due to its high computational cost. The GEC and the POR/FRC errors are only defined for two-frame scenarios, thus will not be considered for geometry estimation. That leaves the angular error and the virtual camera error as the two most practical error functions to be considered for optimization in refractive scenarios.

Experiments

A theoretic result of refractive relative pose computation is scene scale can be determined as opposed to the perspective relative pose computation. In case of synthetic data and zero noise, this was found to be true as can be seen in Figure 5.7. A set of correspondences between two views was used to compute all five of the above described error functions for an exemplary translation and rotation of the second view. The first camera was set into the world coordinate system origin. Then, while maintaining everything else, the scale of the translation was changed. This was done for zero noise, then, increasing amounts of noise were added to the 2D-2D correspondences. The image size was 800 px × 600 px, and the noise was normal distributed with σ in px. Obviously, even a small amount of noise added to the correspondences causes the error functions to not be able

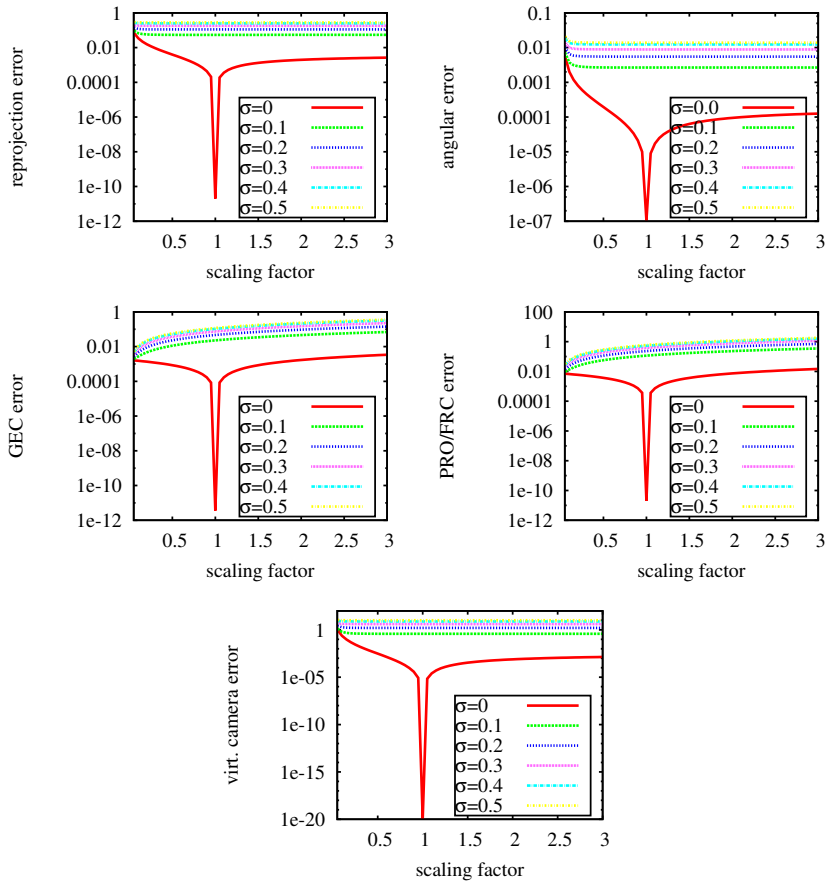

Figure 5.7. Invariance of error function against scaling of scene/translation in case of relative pose problem. Depicted are the results for the reprojection error, the angular error, the GEC error, the POR/FRC error, and the newly proposed virtual camera error.

to determine scene scale correctly. Yet, for zero noise, the theoretical possibility of correctly determining the scale holds true. In case of noise being added to the correspondences, the starting points of the rays move to some small extent. Considering the distances involved, i. e., a few millimeters on the outer housing interface, a few centimeters of camera movement, but a few meters of distance between 3D points and cameras, it becomes clear that the noise is superimposing the signal and therefore makes it impossible to correctly determine the scene's scale.

Figure 5.8 shows the same data as Figure 5.7, with a non-logarithmic scale and as a 3D plot, such that the errors are depicted depending on the noise and the scaling factor. Due to the non-logarithmic scale on the error axis, the clear minima at zero noise and the correct scale are no longer visible. However, it becomes clear that the error functions are not invariant against changes in scale. Figure 5.8 shows error functions for two views only. In case of using a scene with 50 views and a correspondingly large set of 3D points however, the error functions can still not be used to determine scale, as can be seen in Figure 5.9. Note that in order to be able to plot the GEC and the POR/FRC error for 50 views, 2D-2D correspondences between pairs of two views were used. The conclusion drawn from the investigations concerning the scene scale is that scene scale cannot be determined with the proposed non-linear error-functions in case of inexact, i. e., automatically detected corners. However, neither are the error functions completely invariant against scale changes as in the perspective case.

Due to not being able to correctly compute the scene's scale, the results of the test runs experimenting with the different methods for relative pose estimation (Figures 5.11 to 5.13) have been determined after applying the correct scale to the translation. The described relative pose estimation algorithms were tested on synthetically, randomized sets of 2D-2D correspondences. In order to compare perspective and refractive methods, the following procedure was followed:

1. Define refractive camera with image size $800 \times 600\,\text{px}$, focal length $f = 800\,\text{px}$, principal point at $(399.5\,\text{px}, 299.5\,\text{px})$, zero skew, aspect ratio one, and radial distortion $r_1 = 0.1$ and $r_2 = -0.2$. The interface distance was $d = 10\,\text{mm}$, interface thickness $d_g = 20\,\text{mm}$, and interface

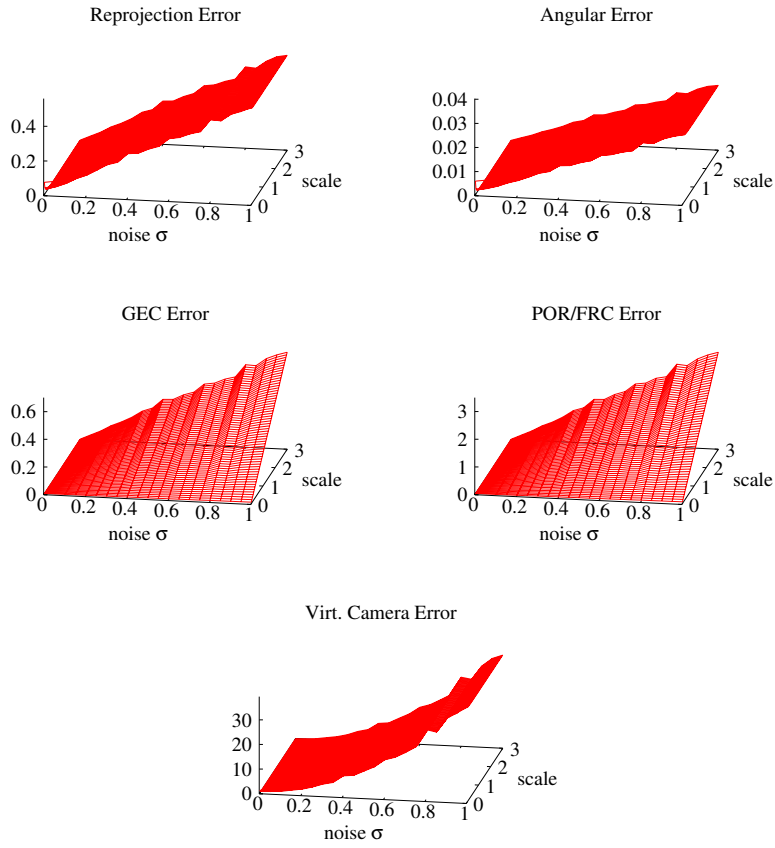

Figure 5.8. For an exemplary set of two views, the translation scale between the views was varied and is depicted on the scale-axis. The noise axis depicts the normal distributed noise that was added to the 2D-2D correspondences. Shown are the reprojection error, the angular error, the GEC error, the POR/FRC error and the virtual camera error.

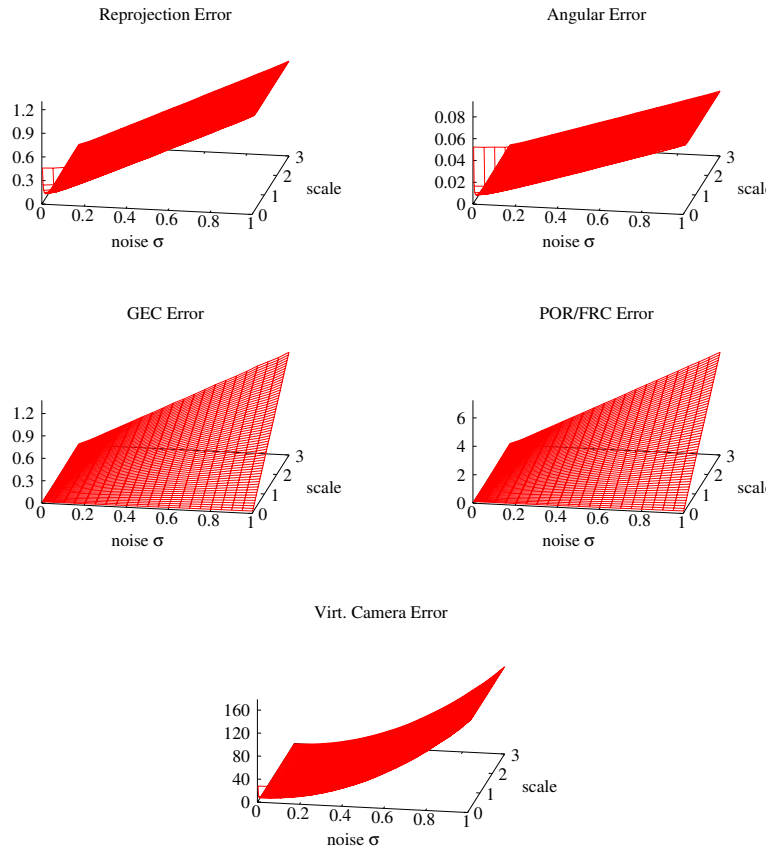

Figure 5.9. Depicted are the five different error functions as in Figure 5.8. In this case a scene with 50 views and a correspondingly large set of 3D points were used for computing the error function instead of just two.

tilt $\theta_1 = 0^\circ$, $\theta_2 = 0.5^\circ$.

2. Render calibration images to find the best fitting perspective camera, which had focal length $f = 1066.79\,\text{px}$, principal point $(400.48\,\text{px}, 299.64\,\text{px})$, $r_1 = -0.19$, and $r_2 = -0.93$.

3. Create different relative poses for the second camera, while setting the first camera into the world coordinate system origin. For each configuration, project a set of 3D points by the perspective and by the refractive camera models to obtain perspective and refractive ground truth 2D-2D correspondences.

4. Three camera model configurations can now be experimented with: perspective camera model on perspective data, perspective camera model on underwater data, and refractive camera model on underwater data.

5. By adding increasing amounts of normal distributed noise to the data, robustness can be tested.

6. After adding outliers to the correspondences, RANSAC frameworks [FB81] can be tested with a linear, initial estimation and a Maximum-Likelihood (ML) optimization.

7. For each pose estimation method, the results are summarized in one plot (example plot in Figure 5.10): the x-axis shows the increasing amount of noise. On the left y-axis, the pose error is depicted, combining rotation and translation, i. e., in case of translation (green) the unit is mm and in case of rotation (red) the unit is degrees. On the right y-axis are the reprojection error and the failure rate. That means in case of the reprojection error (magenta) the unit is pixel and the failure rate (blue) is the fraction of completely failed runs that are not part of the evaluation, i. e., the failure rate is always between zero and one. Each column in the following plots shows first the results of the linear method on outlier-free data, the results of the linear method combined with a non-linear optimization on outlier-free data, and finally a combination of the linear and non-linear method within a RANSAC framework [FB81] on data with outliers.

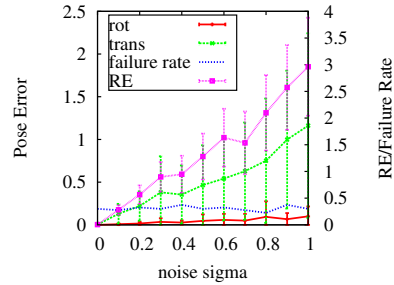

Figure 5.10. Exemplary evaluation plot for pose estimation. On the x-axis is the increasing amount of noise (in pixels) added to the 2D image points. On the left y-axis is the pose error. In case of the camera translation (green), the unit is mm. In case of rotation (blue), the unit is degrees. On the right y-axis is the reprojection error (magenta) in px. Additionally, the failure rate is plotted in (blue). The failure rate is always in the interval $[0, 1]$ and shows how many pose results were not considered in the evaluation due to computation failures.

Figure 5.11 shows results of relative pose estimation on perspective data using the perspective camera model (left column), and for using the perspective model on underwater data (right column). In Figures 5.12 and 5.13 are the results for applying different refractive methods on underwater data to determine relative pose. The comparison between perspective relative pose on perspective data and perspective relative pose on underwater data shows that both methods perform similarly well and there does not seem to be a large error introduced by ignoring refraction. However, firstly, ignoring refraction causes a non-zero error for zero noise and secondly, one has to keep in mind that no 3D points exist yet, and hence the error is compensated for by triangulating 3D points during error function computation. The extent of the error introduced by triangulation has already been demonstrated in the analysis in Section 4.3.3. When considering the results of the refractive methods, the CAM-ES optimization and the iterative approach have the highest accuracy and robustness in case of the relative pose problem using refractive cameras. However, the run-time of the CMA-ES optimization, especially in combination with a RANSAC approach, is much higher compared to ML-optimization. There-

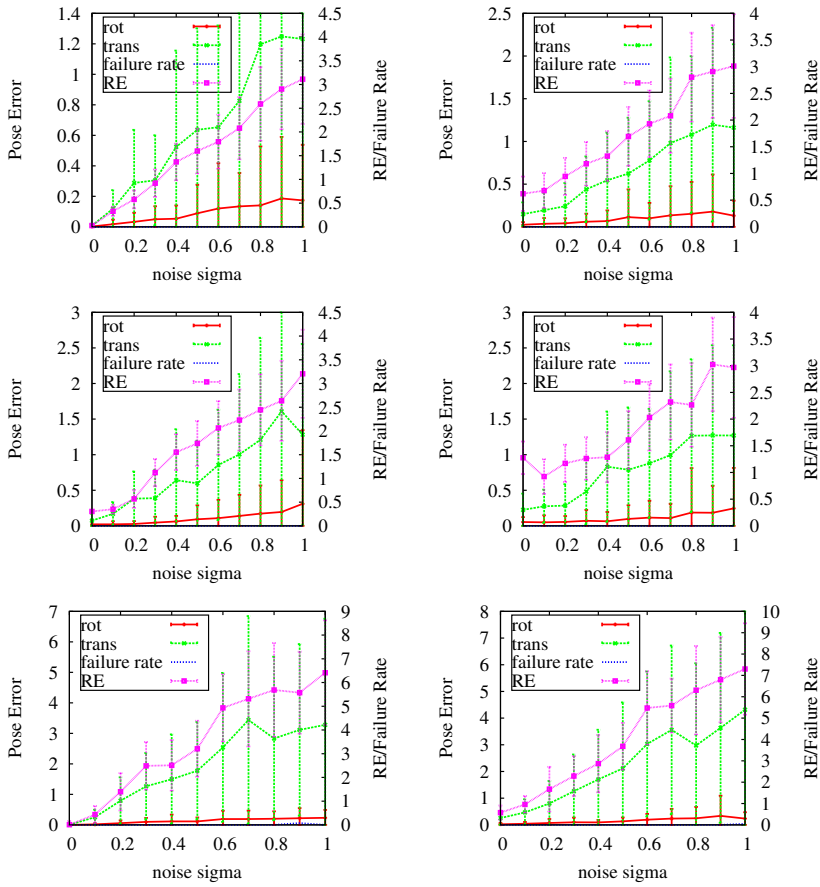

Figure 5.11. Results of perspective, relative pose computation. Left column: perspective camera model on perspective data. Right column: perspective camera model on underwater data. From top to bottom are linear method on noisy data, linear method with non-linear optimization, and RANSAC approach [FB81] using a linear method for generating sample solutions and non-linear ML optimization.

fore, in the following, the iterative approach will be used to determine initial solutions and the Levenberg- Marquardt algorithm will be used for

optimization. The proposed virtual camera error function outperforms the angular error, and hence will be used for non-linear optimization in the following.

5.1.2 3D-Point Triangulation

Assuming the camera poses, intrinsics, and housing parameters of at least two images are known in addition to a set of correspondences between those images, it is possible to retrieve the 3D point for each correspondence. Triangulation based on a set of rays (starting points and directions) can be accomplished by using the midpoint method [HS97a]. It is assumed that for N cameras ($N >= 2$), the poses are known and that a correspondence between those cameras exists. If $\mathsf{X}_{\mathrm{s}_i}$ are the starting points and $\tilde{\mathsf{X}}_{\mathrm{w}_i}$ are the directions in the world coordinate system for $i \in \{1, ..., N\}$ respectively, the common 3D point can be computed for each view $\mathsf{X} = \mathsf{X}_{\mathrm{s}_i} + \kappa_i \tilde{\mathsf{X}}_{\mathrm{w}_i}$ $\kappa_i \in \mathbb{R}$. Then, with $\mathsf{X} \in \mathbb{R}^3$ being the newly triangulated 3D point:

$$\epsilon = \underbrace{\min}_{\mathsf{X},\kappa_1,...,\kappa_n} \sum_{i \in \{1,...,N\}} |\mathsf{X}_{\mathrm{s}_i} + \kappa_i \tilde{\mathsf{X}}_{\mathrm{w}_i} - \mathsf{X}|^2 \tag{5.1.31}$$

is the linear least squares problem that needs to be solved in order to calculate the 3D point X.

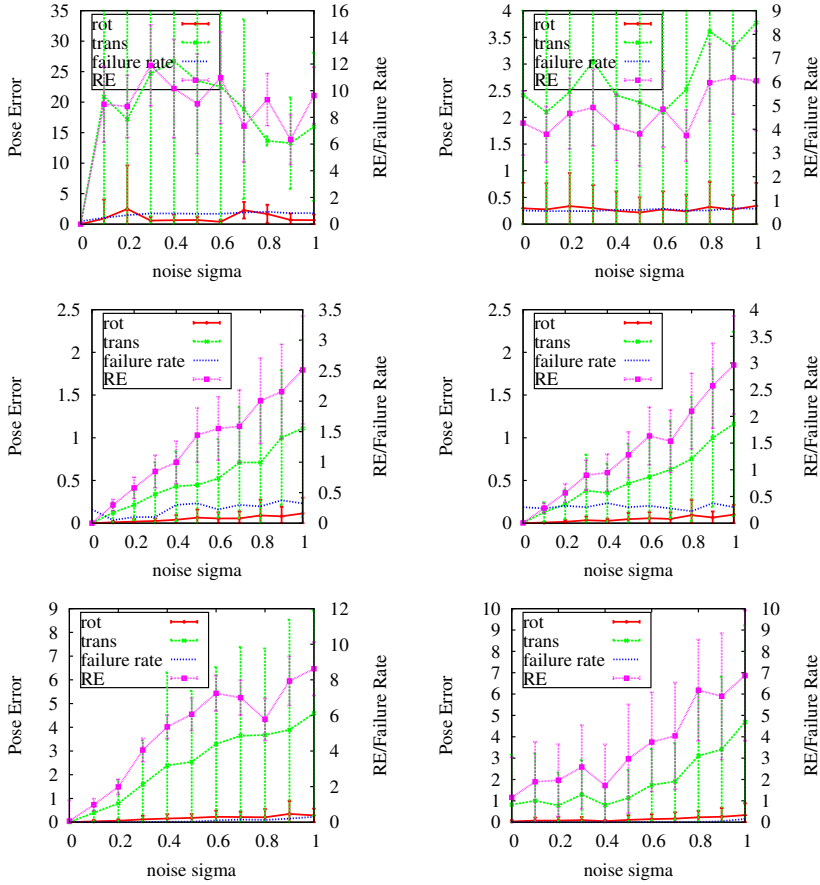

Figure 5.12. Results of refractive relative pose computation. From top to bottom are a linear method on noisy data, a linear method with non-linear optimization, and the RANSAC approach [FB81] using a linear method for generating sample solutions and a non-linear method for optimization. In the left column are results of the refractive linear method with ML optimization using the virtual camera error function. The right column shows results of using the iterative method with ML optimization using the virtual camera error.

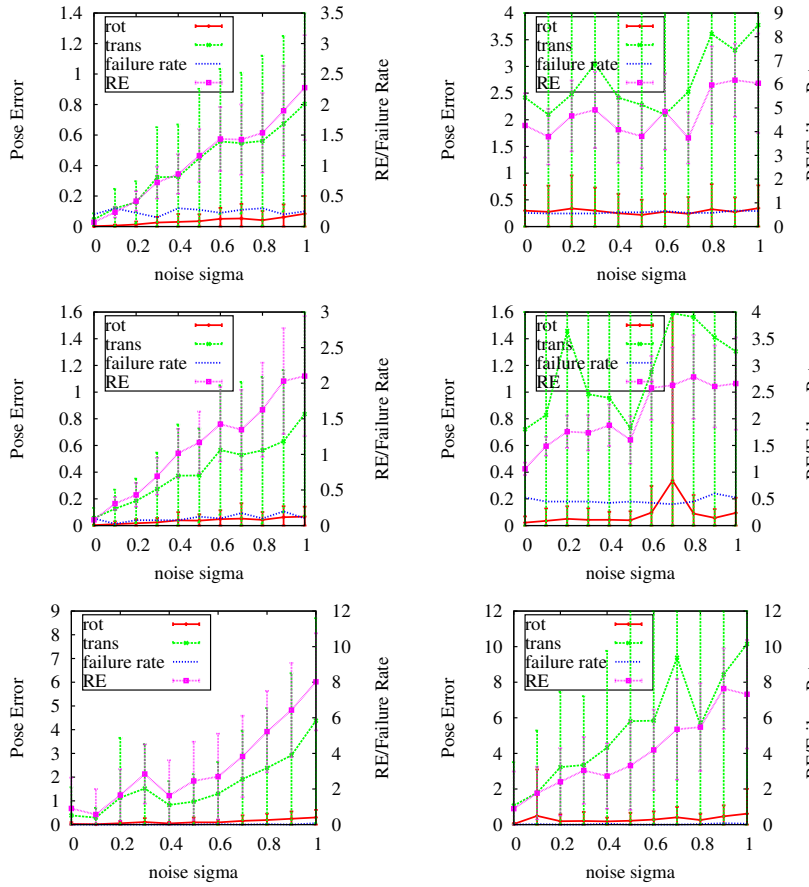

Figure 5.13. Results of refractive relative pose computation. From top to bottom are a linear method on noisy data, a linear method with non-linear optimization, and the RANSAC approach [FB81] using a linear method for generating sample solutions and a non-linear method for optimization. On the left are results of CMA-ES and virtual camera error for determining the initial solution, then the iterative method with CMA-ES optimization. The right column shows results for the iterative approach for the initial, linear solution combined with the angular error for optimization.

5.1.3 Absolute Pose

After computing the relative pose and triangulating 3D points for monocular image sequences, or in case of a calibrated stereo rig, 2D-3D correspondences can be matched, allowing to determine the absolute pose with respect to the 3D points directly. In the following, perspective and refractive methods for absolute pose will be discussed and compared in experiments.

Perspective Absolute Pose

For perspective camera models, the absolute pose problem has been considered in the computer vision community for several years and is known as PnP (Perspective-n-Point) problem. Hence, a large number of algorithms exists. In the Direct Linear Transform ([HZ04], p. 178) the projection of a 3D point $\mathbf{X}_k$ onto a 2D point $\mathbf{x}_k, k \in \{1, ..., K\}$ by a projection matrix $\mathbf{x}_k = \mathbf{P}\mathbf{X}_k$ is used to derive a set of linear equations in the unknowns of $\mathbf{P}$. It requires a minimal set of six correspondences. A widely used approach that does not estimate the pose directly, but iteratively, was proposed in 1995 by Dementhon and Davis [DD95]: in the POSIT (Pose from Orthography and Scaling with ITerations) algorithm, the pose is computed by iteratively refining an approximation that is based on a scaled orthographic camera model. Both methods cannot easily be adapted to the refractive camera model. Haralick et al. [HLON94] present an extensive analysis of the P3P problem with different methods by utilizing the basic insight that distances between 3D points are the same in the camera and the world coordinate systems (Figure 5.14). The distances and the angles between the corresponding rays in the camera coordinate system (red lines in Figure 5.14) allow to derive a set of equations that can be solved for the camera-point distances yielding 3D points in the camera coordinate system. Finally, the absolute orientation problem remains to be solved. Meaning rotation and translation between two sets of 3D-3D point correspondences need to be determined. [HLON94] describes a linear procedure as follows. Rotation matrix $\mathbf{R}$ and translation vector C are required to transform the points in the camera coordinate system into points in the world coordinate

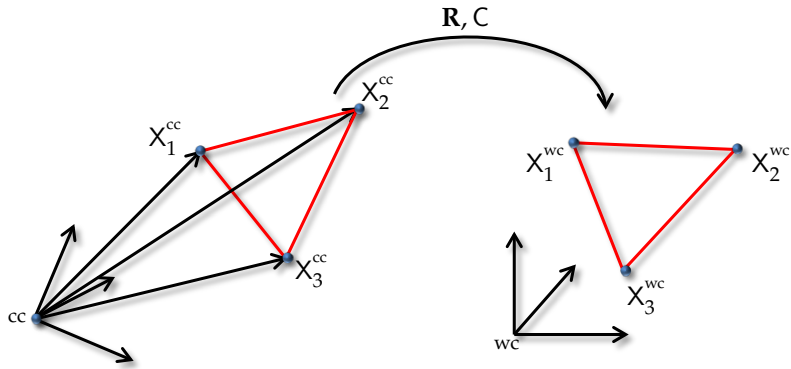

Figure 5.14. P3P pose estimation. The distances between the points (length of the red lines) are invariant to rotation and translation.

system:

$$X_k^{wc} = \mathbf{R}X_k^{cc} + C, \quad k \in \{1,2,3\}. \tag{5.1.32}$$

This problem cannot be solved linearly directly due to 12 unknowns and given only nine equations. However, the properties of the rotation matrix allow to compute the third column by using the first two columns:

$$r_3 = r_1 \times r_2. \tag{5.1.33}$$

In addition, the points in the camera coordinate system are coplanar, thus the transform can be computed for a set of points with zero z-component. Hence, the resulting equations are for $i \in \{1,2,3\}$:

$$\begin{aligned} r_{11}X_k^{cc} + r_{12}Y_k^{cc} + C_x &= X^{wc} \\ r_{21}X_k^{cc} + r_{22}Y_k^{cc} + C_y &= Y^{wc} \\ r_{31}X_k^{cc} + r_{32}Y_k^{cc} + C_z &= Z^{wc}. \end{aligned} \tag{5.1.34}$$

They can be utilized in a linear equation system and be solved for the first two columns of $\mathbf{R}$ and the translation vector C.

Refractive Linear Approach using FRC and POR

In [ARTC12], a method for camera calibration was proposed based on the Flat Refractive constraint (3.2.17) and the Plane of Refraction constraint (3.2.18). Since camera calibration usually involves the computation of the camera's pose, the idea in [ARTC12] for calibration can be adapted to absolute pose computation without calibrating the housing parameters:

$$\begin{aligned} \tilde{X}_{w_k} \times (\mathbf{R}'X_k + C' - X_{s_k}) &= 0 \\ (\mathbf{R}'X_k + C')^T(\tilde{n} \times X_{s_k}) &= 0, \end{aligned} \tag{5.1.35}$$

for each 2D-3D correspondence $k \in \{1, ..., K\}$. Both constraints are linear in the unknowns $\mathbf{R}'$ and C'. In Table A.8 the entries of the resulting linear system of equations can be found. Since the FRC is a constraint that basically determines an angle between two rays and the constraint based on the POR determines if a point is on a plane that extends from the camera center along its viewing ray, this method cannot robustly determine the correct camera translation in z-direction (compare to [ARTC12]). However, using the virtual camera error described above, the translation in z-direction can be optimized efficiently, using the Levenberg-Marquardt algorithm with one parameter. Afterwards, $\mathbf{R}'$ and C' need to be transformed into the global-to-local transform by: $\mathbf{R} = \mathbf{R}'^T$ and $C = -\mathbf{R}^T C'$.

Iterative Approach

Similar to the iterative method for computing relative pose as described above, an iterative method for absolute pose can be derived using a set of K 2D-3D correspondences. The constraint directly involves the 3D points:

$$X_k = \mathbf{R}X_{s_k} + C + \kappa_k \mathbf{R}\tilde{X}_{w_k} \quad \forall k \in \{1, ..., K\}, \tag{5.1.36}$$

where the matrix $\mathbf{R}$, and the translation C are unknown. Hence, (5.1.36) is non-linear in the unknowns and is therefore solved iteratively, by alternatively solving for the transformation and all $\kappa_k, k \in \{1, 2, ..., K\}$ in each iteration. The linear system of equations resulting from stacking (5.1.36) for all $k \in \{1, ..., K\}$ and keeping all κ_k constant is solved for $\mathbf{R}$ and C. Note that due to $r_3 = r_1 \times r_2$, the result for the z-coordinate can be determined

separately:

$$\begin{aligned} Z_k &= \kappa_k(\mathsf{r}_1 \times \mathsf{r}_2)^{\mathrm{T}}\tilde{\mathsf{X}}_{\mathrm{w}_k} + (\mathsf{r}_1 \times \mathsf{r}_2)^{\mathrm{T}}\mathsf{X}_{\mathrm{s}_k} + C_z \\ \Rightarrow C_z &= Z_k - (\kappa_k(\mathsf{r}_1 \times \mathsf{r}_2)^{\mathrm{T}}\tilde{\mathsf{X}}_{\mathrm{w}_k} + (\mathsf{r}_1 \times \mathsf{r}_2)^{\mathrm{T}}\mathsf{X}_{\mathrm{s}_k}). \end{aligned} \tag{5.1.37}$$

This leaves two equations per correspondence to be used for linear estimation of eight parameters.

$$\mathbf{A} = \begin{pmatrix} (\kappa_1\tilde{\mathsf{X}}_{\mathrm{w}_1} + \mathsf{X}_{\mathrm{s}_1})^{\mathrm{T}} & \mathsf{0}^{\mathrm{T}} & 1 & 0 \\ \mathsf{0}^{\mathrm{T}} & (\kappa_1\tilde{\mathsf{X}}_{\mathrm{w}_1} + \mathsf{X}_{\mathrm{s}_1})^{\mathrm{T}} & 0 & 1 \\ \vdots & & & \\ (\kappa_K\tilde{\mathsf{X}}_{\mathrm{w}_K} + \mathsf{X}_{\mathrm{s}_K})^{\mathrm{T}} & \mathsf{0}^{\mathrm{T}} & 1 & 0 \\ \mathsf{0}^{\mathrm{T}} & (\kappa_K\tilde{\mathsf{X}}_{\mathrm{w}_K} + \mathsf{X}_{\mathrm{s}_K})^{\mathrm{T}} & 0 & 1 \end{pmatrix}, \tag{5.1.38}$$

with $\mathbf{x} = (r_{11}r_{12}r_{13}r_{21}r_{22}r_{23}C_xC_y)^{\mathrm{T}}$ and $\mathbf{b} = (X_1Y_1...X_KY_K)^{\mathrm{T}}$. Then, all κ_k are updated using:

$$\kappa_k = \kappa_k + (\mathbf{R}\tilde{\mathsf{X}}_{\mathrm{w}_k})^{\mathrm{T}}(\mathsf{X}_k - \mathsf{C} - \mathbf{R}\mathsf{X}_{\mathrm{s}_k}) - \kappa_k = (\mathbf{R}\tilde{\mathsf{X}}_{\mathrm{w}_k})^{\mathrm{T}}(\mathsf{X}_k - \mathsf{C} - \mathbf{R}\mathsf{X}_{\mathrm{s}_k}), \tag{5.1.39}$$

which can be derived from Equation (5.1.36) due to $\tilde{\mathsf{X}}_{\mathrm{w}_k}$ having unit length and by applying the scalar product on both sides.

Sturm's Method

The method described by Sturm et al. [SRL06] has a similar idea to the method described in [HLON94], but can deal with refractive and also other general, ray-based camera models. The values for the κ_k described in the section above, can be determined without knowing the pose of the camera within the world coordinate system. This is done by utilizing the known distances between the 3D points, i. e., take a set of three 3D points in the world coordinate system and compute the pair-wise distances between them: d_{12}, d_{13}, and d_{23}. Then, the rays in the camera coordinate system $\mathsf{X}_{\mathrm{s}_k} + \kappa_k\tilde{\mathsf{X}}_{\mathrm{w}_k}$ must result in 3D points with the corresponding distances:

$$\begin{aligned} \| \mathsf{X}_{\mathrm{s}_1} + \kappa_1\tilde{\mathsf{X}}_{\mathrm{w}_1} - \mathsf{X}_{\mathrm{s}_2} - \kappa_2\tilde{\mathsf{X}}_{\mathrm{w}_2} \|_2^2 &= d_{12} \\ \| \mathsf{X}_{\mathrm{s}_1} + \kappa_1\tilde{\mathsf{X}}_{\mathrm{w}_1} - \mathsf{X}_{\mathrm{s}_3} - \kappa_3\tilde{\mathsf{X}}_{\mathrm{w}_3} \|_2^2 &= d_{13} \end{aligned} \tag{5.1.40}$$

$$\| \mathsf{X}_{s_2} + \kappa_2 \tilde{\mathsf{X}}_{w_2} - \mathsf{X}_{s_3} - \kappa_3 \tilde{\mathsf{X}}_{w_3} \|_2^2 = d_{23},$$

which is a non-linear system with three equations in three unknowns κ_k. It is possible to eliminate two unknowns and retrieve one equation with one unknown, which can be turned into an 8^{th} order polynomial in one variable (e. g., κ_3), for which the coefficients can be determined using a toolbox like Maxima or Matlab's symbolic toolbox. Up to eight solutions can be found for the polynomial, and hence up to eight solutions for the three κ_k can be determined. For each real solution, a rigid 3D-3D transformation needs to be estimated in a second step, which is done linearly (refer to [HLON94]). An optional fourth correspondence is used in the optimization in order to determine the correct solution robustly.

Nistér Approach

Nistér and Stewénius [NS07] propose a different method for solving the absolute pose problem with only three points for a general camera model. The solution is based on the idea of reducing the problem to the computation of intersections between a circle and a ruled quartic surface, which results in an 8^{th} degree polynomial of which the roots need to be found.

Maximum Likelihood Optimization

The initial poses computed by any of the methods described above need to be optimized in case of noise in the 2D-3D correspondences. As in case of the relative pose problem, a suitable error function is required for this. The above described reprojection error, the angular error, and the virtual camera error can be straightforwardly applied to the absolute pose problem as well. Any of these error functions can then be used within a Levenberg-Marquardt algorithm or a CMA-ES algorithm for optimizing the initial pose.

Experiments

The described absolute pose algorithms were tested on synthetic data in the same scenario described in the relative pose section. Only this time, a set of 3D points was used in order to get the required 2D-3D

correspondences. Figure 5.15 shows the results of the perspective absolute pose algorithm (here POSIT and a Levenberg-Marquardt optimization) on perspective data (left column) and of applying the perspective method to underwater data (right column). Figures 5.16 to 5.18 depict results of the refractive or general camera model methods on underwater data, starting with Sturm's method, followed by Nistér's, the proposed iterative approach with Levenberg-Marquardt optimization, and the proposed linear approach using the FRC and POR. In Figure 5.18, results are give for using the CMA-ES method with high initial deviation as an initial method, of the iterative method with CMA-ES as an optimization method, and for using the iterative method and CMA-ES within a RANSAC framework [FB81] on underwater data with outliers.

Applying the POSIT algorithm to perspective data shows the expected results. With growing noise on the 2D points, the pose error, but also the reprojection error increases. Using a non-linear optimization method on the initial POSIT estimate decreases mainly the pose error. Applying the RANSAC routine on data with outliers allows to estimate the pose robustly. More interesting is the case of applying the perspective methods to underwater data. In this case, the pose error caused by the model error is evident, this time, the 3D points cannot compensate for the error introduced by refraction. Note that absolute pose is often computed for a large number of views, consequently, the error made for each view accumulates over time. This problem can be avoided by applying a method for absolute pose estimation that explicitly incorporates refraction. As can be seen in the top row in Figure 5.16, Sturm's and Nistér's methods work if no noise is present in the data, but the performance quickly deteriorates if noise is added to the 2D image points. However, similar to the relative pose estimation, the non-linear optimization can be used to fix the large initial error. The same is true for initial estimation and non-linear optimization on data with outliers within a RANSAC framework. Figure 5.17 on the left shows results for the described iterative method as an initial estimator, followed by the Levenberg-Marquardt optimization with the virtual camera error as error function. Note that the results are very accurate even when compared with the POSIT results on perspective data. Strongly sensitive to noise is the linear method using the POR and FRC for absolute pose estimation (Figure 5.17 on the right). However, it

can still be used because the non-linear optimization can cure the problem. Figure 5.18 shows that by far the best initial results can be obtained by using CMA-ES. Combining the iterative approach with CMA-ES for optimization yields results of comparable accuracy and the combination of both within a RANASC framework yields the most accurate pose. However, using CMA-ES, especially in a RANSAC framework is very time consuming.

Usually, one strives to minimize the number of correspondences required for initial pose estimation. The reason for this is that during the RANSAC algorithm, the chances of drawing a sample without outliers are maximized. However, if the minimal solution, i. e., Nistér's or Sturm's methods, are very sensitive to noise, this may not always be advantageous. Especially, in case of a correspondences set with very few true outliers, but naturally noisy correspondences (degraded contrast, backscatter, marine snow, etc. in underwater images), methods less sensitive to noise can outperform the minimal solutions because they may require less samples during the RANSAC algorithm. Therefore, in the following, the iterative method will be used for initialization and the Levenberg-Marquardt optimization with the virtual camera error will be used for optimizing the initial solution.

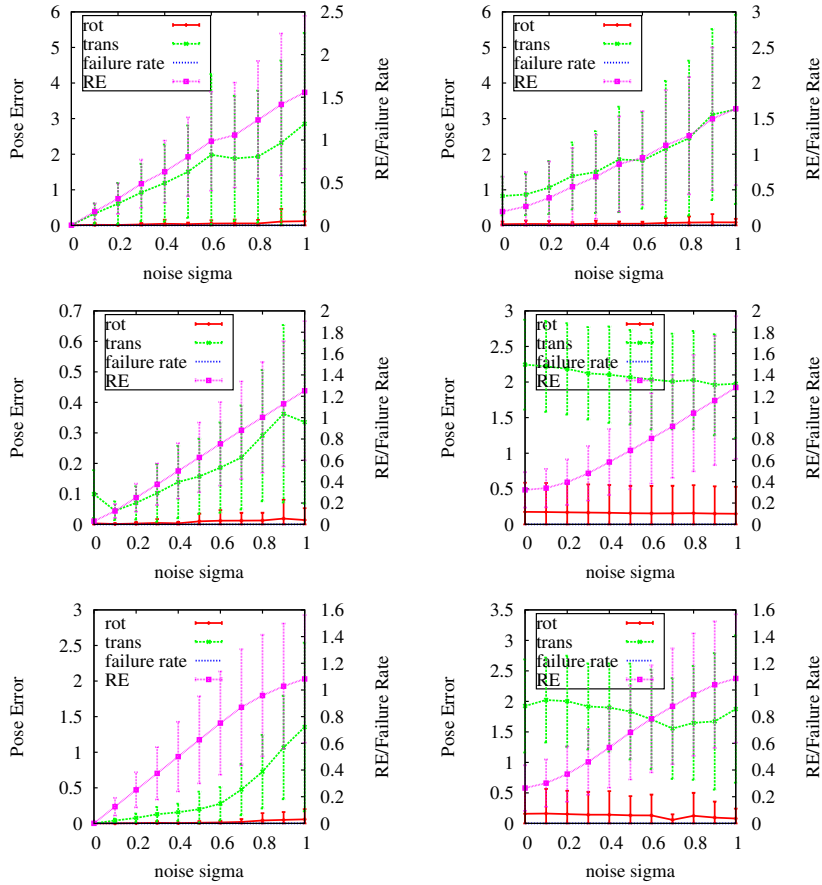

Figure 5.15. Perspective absolute pose estimation results. The top row shows linear estimation results, middle row the non-linear optimization results based on the top row, and the bottom row shows RANSAC results using the linear and the optimization method. Left column: perspective results on perspective data. Right column: perspective results on underwater data.

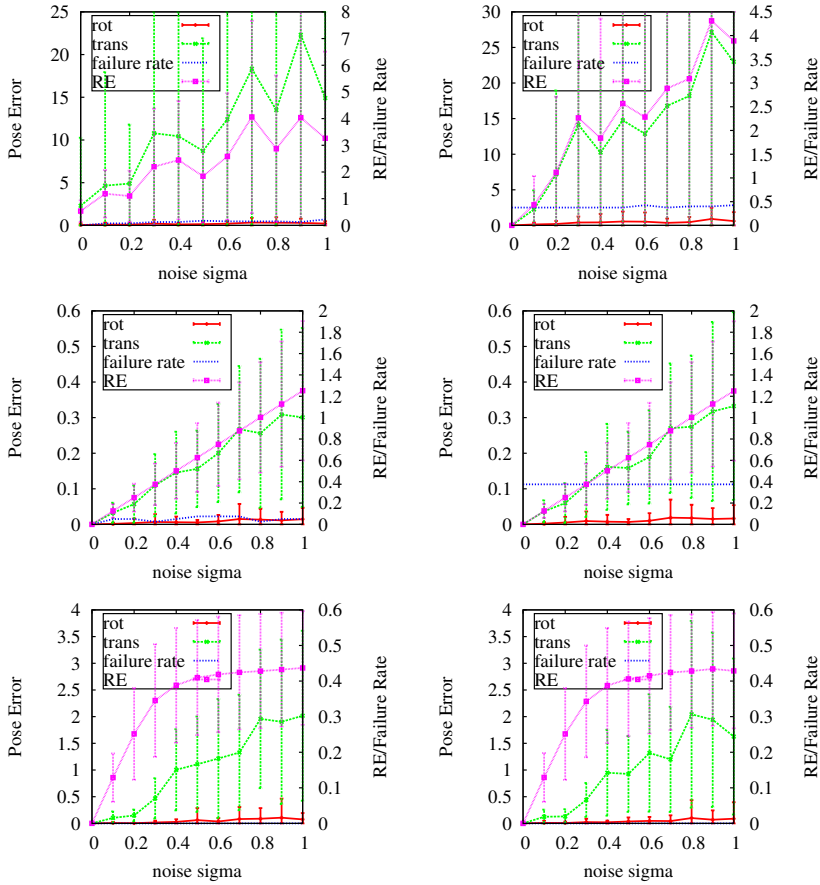

Figure 5.16. Refractive absolute pose estimation results. Top row shows linear estimation results, middle row the non-linear optimization results based on the top row, and the bottom row shows the RANSAC results using the linear and the optimization method. The left column depicts underwater results on underwater data with Sturm's method, the right column results for Nistér's method.

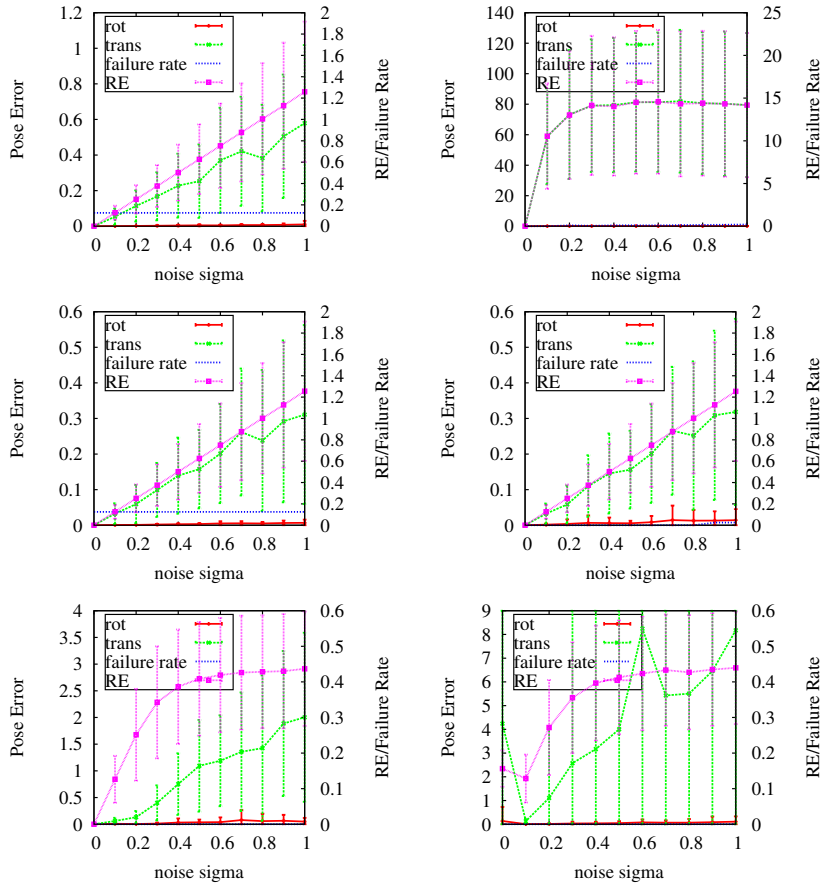

Figure 5.17. Refractive absolute pose estimation results with the proposed iterative method and the proposed linear method

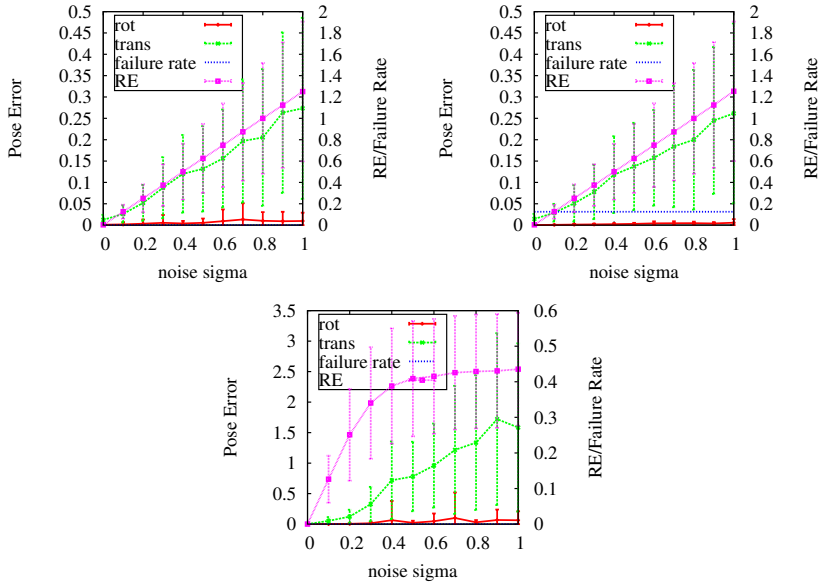

Figure 5.18. Refractive absolute pose estimation results using CMA-ES for initialization (top row on the left), followed by a combination of the iterative method and CMA-ES on the right. In the bottom row are the RANSAC results.

5.1.4 Bundle Adjustment

In the context of Structure-from-motion, bundle adjustment (BA) is a term used for a non-linear optimization of the whole reconstructed scene. Camera poses, 3D points, and possibly camera intrinsics or housing parameters are optimized simultaneously using the 2D observations. An introduction to bundle adjustment in general can be found in [TMHF00] and [McG04]. Overviews in the context of SfM are treated in [HZ04] and [Sze11]. In common perspective bundle adjustment scenarios, the reprojection error in all images is minimized, i. e., there can be thousands or even hundreds of thousands of parameters and many more observations. In order to simultaneously optimize all parameters, a squared error function is minimized, usually using the Levenberg-Marquardt algorithm. This requires the Hessian of the error function, hence, very large matrices need to be handled, which results in one of the major challenges of bundle adjustment: it is time and memory consuming and therefore requires special care when being implemented. Luckily, careful ordering of the parameters and observations causes the required matrices to be sparse and of block structure, which allows a multitude of optimizations in the implementation. The theoretical basics and some general implementation issues are addressed in Appendix A.4.

Bundle adjustment often runs for each newly added view, hence, it is probably the component in the SfM process that requires most of the run-time and its performance is therefore usually the bottleneck. Not surprisingly, still a lot of research is done to optimize bundle adjustment and to improve convergence. Recent works allow to deal with tens of thousands of images with accordingly large 3D point clouds [WACS11, JNS+10, JNS+12].

For less ambitious projects, different implementations are available, a commonly used example is [LA09]. However, it only supports explicit constraints of the type $f(p) = I$, with p being the vector of all parameters and I being the vector of all observations, and hence the library cannot be used to optimize the virtual camera error described above. Additionally, no parameters can be shared between all views, i. e., for a camera moving through a scene with constant intrinsics, those cannot be optimized.

Another recent work considering stereo rigs [KTS11] is concerned with

a general setup of current stereoscopic systems for capturing 3D movies. In case of rigidly coupled rigs of two or more cameras, the relative transform between master and slave cameras is assumed to be constant throughout the image sequence. Consequently, the transformation of the whole stereo rig $\mathbf{T}_i, i \in \{1, ..., N\}$ is applied to 3D points, then, points can be transformed into the local coordinate systems of the slave cameras by the relative slave transformation $\mathbf{T}_j, j \in \{1, ..., M\}$. Note that this does not destroy the sparse block structure of the matrices used.

As mentioned above, [LA09] can only deal with explicit constraints. This is a common in perspective scenarios, where the reprojection error is minimized, and hence only explicit constraints are required. Only very few works can be found in the literature, where implicit constraints need to be optimized. A recent work by Steffen et al. [SFF12] addresses the problem of optimizing scenes based on implicit trifocal constraints, which does not require approximated values for the 3D points.

In this thesis, a system is implemented for implicit constraints that supports parameters that all cameras have in common, i. e., intrinsic parameters in the perspective case or underwater housing parameters. In addition, constraints between parameters are supported, hence, quaternion unit length, interface normal unit length, or a fixed rig baseline length, are supported. Such a system requires applying the Gauss-Helmert model [McG04], which solves the following system of equations in each iteration:

$$\underbrace{\begin{bmatrix} \mathbf{A}_\mathrm{g}^\mathrm{T}(\mathbf{B}_\mathrm{g}\mathbf{C}_{ll}\mathbf{B}_\mathrm{g}^\mathrm{T})^{-1}\mathbf{A}_\mathrm{g} & \mathbf{H}_\mathrm{h}^\mathrm{T} \\ \mathbf{H}_\mathrm{h} & \mathbf{0} \end{bmatrix}}_{\mathbf{N}} \begin{bmatrix} \Delta\mathsf{p} \\ \mathsf{k}_\mathrm{h} \end{bmatrix} = \begin{bmatrix} -\mathbf{A}_\mathrm{g}^\mathrm{T}(\mathbf{B}_\mathrm{g}\mathbf{C}_{ll}\mathbf{B}_\mathrm{g}^\mathrm{T})^{-1}\mathrm{g}(\mathsf{p},\mathsf{l}) \\ -\mathrm{h}(\mathsf{p}) \end{bmatrix}, \quad (5.1.41)$$

where $\mathrm{g}(\mathsf{p},\mathsf{l}) = 0$ is the error function containing all observations depending on the parameter vector p and the observation vector l. $\mathbf{A}_\mathrm{g}$ is the Jacobian with respect to the parameters and $\mathbf{B}_\mathrm{g}$ is the Jacobian with respect to the observations. $\mathrm{h}(\mathsf{p}) = 0$ contains all constraints between parameters with $\mathbf{H}_\mathrm{h}$ being the corresponding Jacobian. $\mathbf{C}_{ll}$ comprises uncertainties of the observations and can be set to identity if those uncertainties are unknown. $\Delta\mathsf{p}$ is the update on the parameters resulting in the iteration and k_h contains a set of Lagrange Multipliers (for more detailed information see Appendix A.4).

Perspective Bundle Adjustment

In case of perspective bundle adjustment, the error function measures the reprojection error for all camera views $i \in \{1, ..., N\}$, i. e., the pair-wise distances between the set of 2D points measured $\check{\mathsf{x}}_{ijk}$ and projected points x_{ijk}. Each projected point is determined by the master pose parameters, quaternion $\tilde{\mathsf{q}}_i$ and translation C_i, for each $i \in \{1, ..., N\}$ and by the intrinsic parameters. In case of multi-camera rigs, those are extended by the relative rig transforms $\tilde{\mathsf{q}}_j$, C_j for each $j \in \{1, ..., M\}$ and corresponding sets of intrinsic parameters for each slave camera. As mentioned above, the perspective projection can be expressed by explicit constraints of the form:

$$\mathsf{f}_{ijk}(\mathsf{X}_k, \mathbf{P}_{ij}, \mathsf{d}_j) = \check{\mathsf{x}}_{ijk}, \tag{5.1.42}$$

with X_k being the 3D point, $\mathbf{P}_{ij}$ being the projection matrix combining intrinsics for rig camera j, slave extrinsics for camera j, and master extrinsics for camera i. Vector d_j contains lens distortion parameters for camera j. Note that the Jacobian with respect to the observations $\mathbf{B}_\mathrm{g}$ is the identity. It is also possible to keep the intrinsic parameters constant and only optimize 3D points and camera poses. This is applied in scenarios, where the camera is assumed to be calibrated with high accuracy beforehand. No matter which parameters are to be optimized, the Jacobian of the function f in Equation (5.1.42) needs to be computed using all the parameters. For this thesis, analytical derivatives for the monocular and stereo case were computed using the analytic toolbox Maxima[1]. When knowing this derivation, the bundle adjustment system can be implemented by solving (5.1.41) in each iteration. Rotations were parametrized using quaternions, which proved to be superior to incremental Euler angles.

Careful ordering of the parameters causes the matrix $\mathbf{N}$ in (5.1.41) to be sparse and of block structure. In Figure 5.19 only colored parts are non-zero. This is the case no matter if a stereo rig is optimized or if constant intrinsics across the whole sequence are improved.

[1] http://maxima.sourceforge.net/

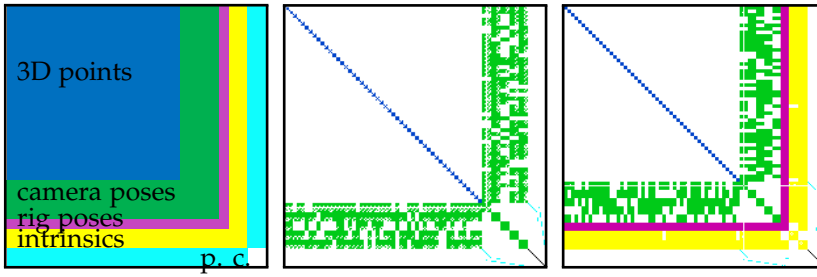

Figure 5.19. Sparse matrices **N** in (5.1.41) for perspective bundle adjustment. Left: general block structure with color coding. p.c. stands for parameter constraints, i. e., for constraints between parameters like quaternion unit length. Middle: sparse matrix for optimization of 3D points and camera poses (monocular). Note that only colored pixels stand for non-zero entries. Right: stereo sequence with pose and 3D point optimization and optimization of intrinsics for both cameras. Colors: 3D points blue, master poses green, rig transform magenta, intrinsics yellow, constraints between parameters cyan.

Refractive Bundle Adjustment

In case of refractive bundle adjustment, the error function cannot easily describe how a 3D point is projected onto a 2D image point because the 12^{th}-degree polynomial that needs to be solved for each observation causes the approach to be infeasible in terms of computation time. However, the above described virtual camera error can be computed efficiently, thus the error function is the distance between observation and projected 3D point compared in the virtual camera (5.1.28). Therefore, the implicit constraints to the bundle adjustment system are:

$$\mathsf{g}_{ijk}(\mathsf{X}_k, \mathbf{P}_{ij}, \mathsf{h}_j, \breve{\mathsf{x}}_{ijk}) = 0, \tag{5.1.43}$$

with h_j containing the housing interface parameters that are used to compute the projection into the virtual camera. In order to be able to minimize a system comprised of constraints of the type (5.1.43), the derivatives of function g need to be computed with respect to both: the parameters, but also the observation x. Only then can the matrices $\mathbf{A}_g$ and $\mathbf{B}_g$ in Equa-

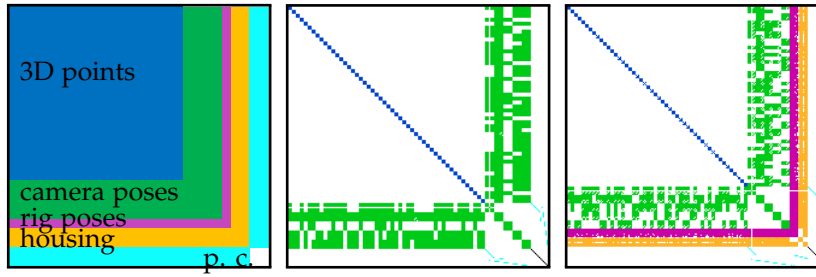

Figure 5.20. Sparse matrices for refractive bundle adjustment. Left: general block structure of sparse matrix N. Middle: optimization of 3D points and camera poses (monocular). Note that only colored parts are non-zero. Right: stereo optimization of housing parameters, 3D points, and camera poses. Colors: 3D points blue, master poses green, rig transform magenta, housing parameters orange, constraints between parameters cyan.

tion (5.1.41) be determined. Note that the interface parameters in h can be optimized or kept constant depending on the knowledge prior to running the system. As in the perspective case, the derivatives were computed using Maxima. However, the case of the interface normal coinciding with the optical axis, i. e., $\tilde{\mathsf{n}} = (0, 0, 1)^\mathsf{T}$, needs to be considered separately. If the housing parameters are to be optimized, each observation consists of the ray in air $\tilde{\mathsf{X}}_\mathsf{a}$ within the local camera coordinate system, which always starts at $(0, 0, 0)^\mathsf{T}$. In case the housing parameters are not part of the optimization, but considered to be known, each observation consists of the ray in water $\tilde{\mathsf{X}}_\mathsf{w}$ with starting point on the outer interface plane X_s and the corresponding intersection with the camera axis C_v, which is defined by the coordinate system origin and the interface normal (Figure 5.6). Rigs can be handled as described above.

Using the virtual camera error and the Gauss-Helmert model does not destroy the sparse block structure shown in the perspective case above. In fact if comparing the refractive sparse matrices in Figure 5.20 to the perspective sparse matrices in Figure 5.19, they do not exhibit any structural differences.

Table 5.1. Parameters and camera models in the eight different application scenarios.

case	cam. model	3D points	master poses	rig pose	intrinsics	housing
1	persp	yes	yes	-	no	-
2	persp	yes	yes	-	yes	-
3	persp	yes	yes	yes	no	-
4	persp	yes	yes	yes	yes	-
5	refr	yes	yes	-	-	no
6	refr	yes	yes	-	-	yes
7	refr	yes	yes	yes	-	no
8	refr	yes	yes	yes	-	yes

Experiments

In order to test the bundle adjustment implementation, eight different cases are investigated more closely. Table 5.1 summarizes the parameters optimized for each case. Constraints between parameters are summarized in Table 5.2. In Section 5.1.1, the experiments showed that in theory the absolute scale of a scene can be determined, however, that in case noise is added to the 2D image points, absolute scale estimation fails. For all five considered error functions, scale estimation does not work in the presence of noise, no matter if the scene contains two or 50 views. Therefore, the scene scale in the bundle adjustment implementation was always fixed.

For each of the eight adjustment scenarios, 50 tests using a scene with eight camera views and at least 500 points with a minimum trail length of three, i. e., each 3D point was seen by at least three views, were conducted. The initial values of rotations and translations were disturbed randomly. In case intrinsics and housing parameters were optimized, those initial values were randomly disturbed as well. Figures 5.21 to 5.23 summarize the resulting parameter errors after optimization.

Figure 5.21 contains rotation errors in degrees after optimization, Figure 5.22 gives camera translation errors and 3D point errors in mm on the left axis and the reprojection error in px on the right axis. Figure 5.23

Table 5.2. Constraints on parameters for the different scenarios.

case	$\|\|\tilde{\mathbf{q}}_{ij}\|\|_2 = 1$	fixed scale	fixed first camera	$\|\|\tilde{\mathbf{n}}_j\|\|_2 = 1$
1	yes	yes	yes	-
2	yes	yes	yes	-
3	yes	yes	yes	-
4	yes	yes	yes	-
5	yes	yes	yes	-
6	yes	yes	yes	yes
7	yes	yes	yes	-
8	yes	yes	yes	yes

depicts errors in the intrinsic parameters after optimization in the first two rows and errors of housing optimization in the bottom row. Note that in contrast to the experiments shown above, this time the perspective camera model is tested on perspective data only and the refractive camera model is tested on underwater data. There were no major differences in convergence between the two. Run times for the refractive camera model are a bit higher compared to the perspective camera model, however, due to the analytic derivatives for the virtual camera error function, refractive BA is not much slower than perspective BA. This is a great improvement compared to using the reprojection error or even the preliminary version of the virtual camera error based on caustic points, published in [SK11a]. In [SK11a] run-time for only eight images was in the order of several hours, while the newly proposed virtual camera error can be optimized in a matter of seconds. In terms of accuracy, the perspective BA outperforms the refractive BA especially in case of stereo camera rigs, where the housing parameters need to be optimized. However, as will be seen when comparing refractive SfM on underwater data to perspective SfM on underwater data, accuracy of refractive BA is very good. The implementation used for this thesis works well and reasonably fast. Its performance and scalability are not yet comparable to state-of-the-art systems like [WACS11, JNS+10, JNS+12]. However, it allows to efficiently optimize the virtual camera error, and thus to explicitly model refraction.

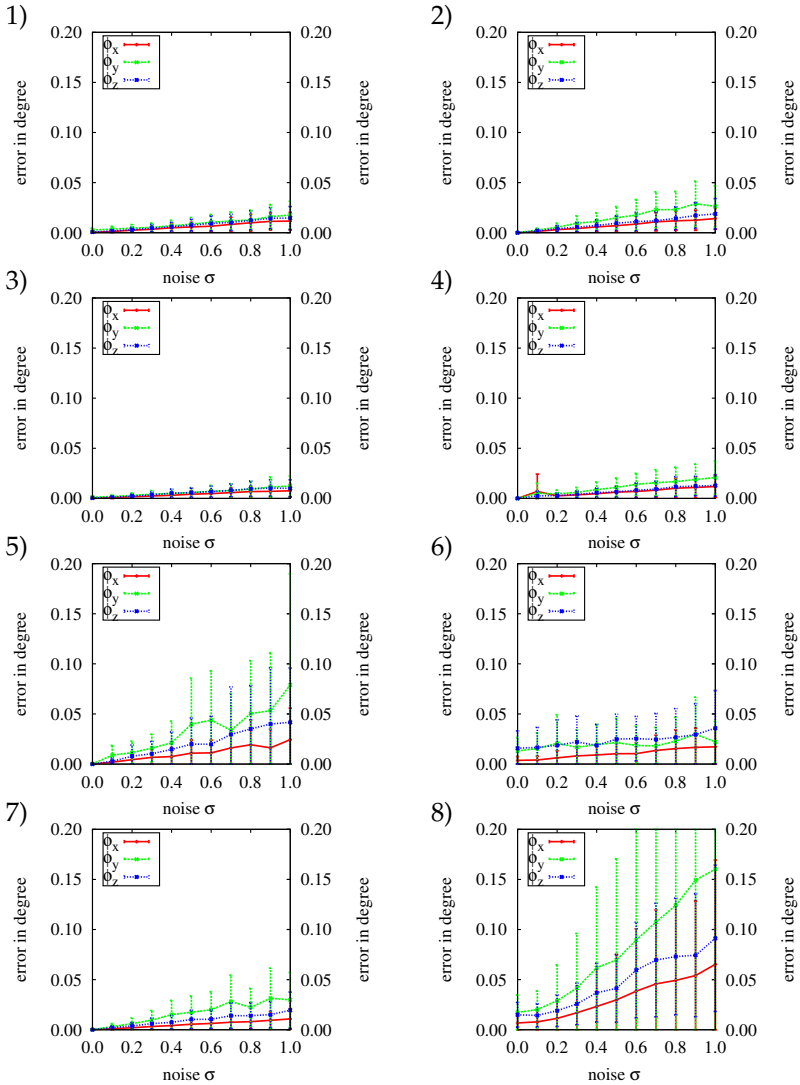

Figure 5.21. Rotation estimation results of bundle adjustment. Plotted are the resulting mean errors and standard deviation in brackets, initial errors and standard deviation were $0.22^\circ(0.15^\circ)$.

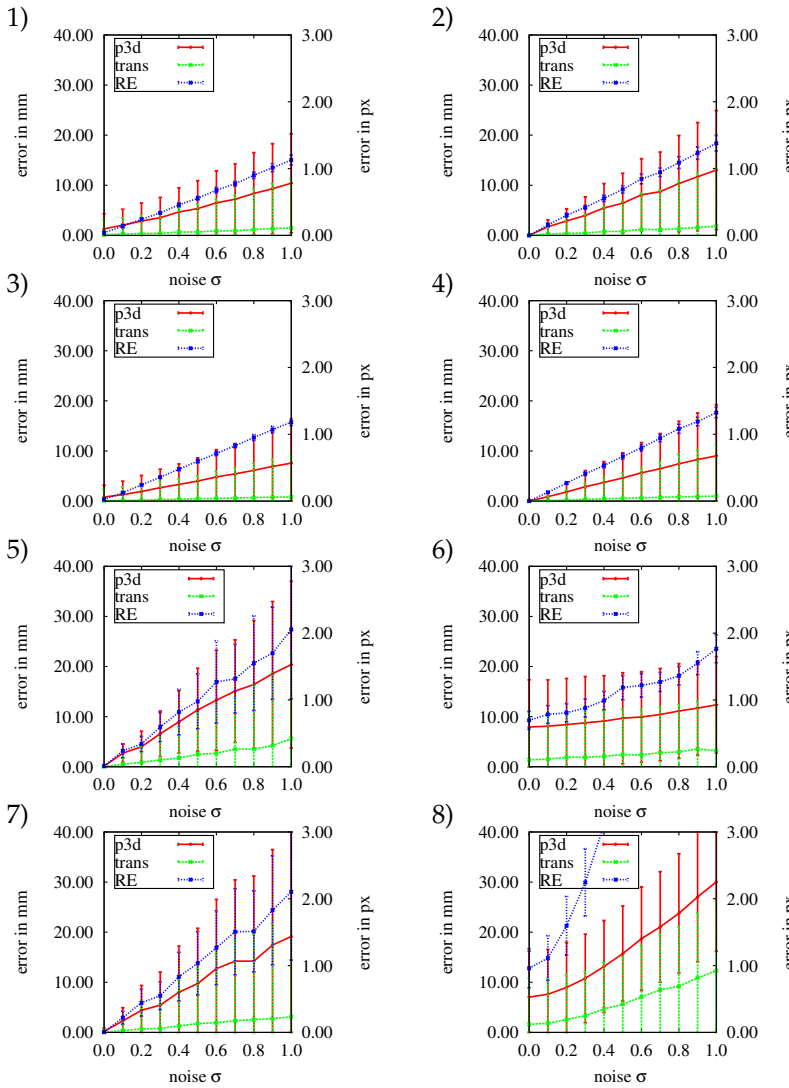

Figure 5.22. The left axis shows BA estimation results for translation and 3D points, the right axis for the reprojection error. Initial errors and standard deviation (in brackets) were 12 mm(10 mm) for translation, 3D point error: 48 mm(19 mm), reprojection error: (RE) 8.5 px(1.5 px).

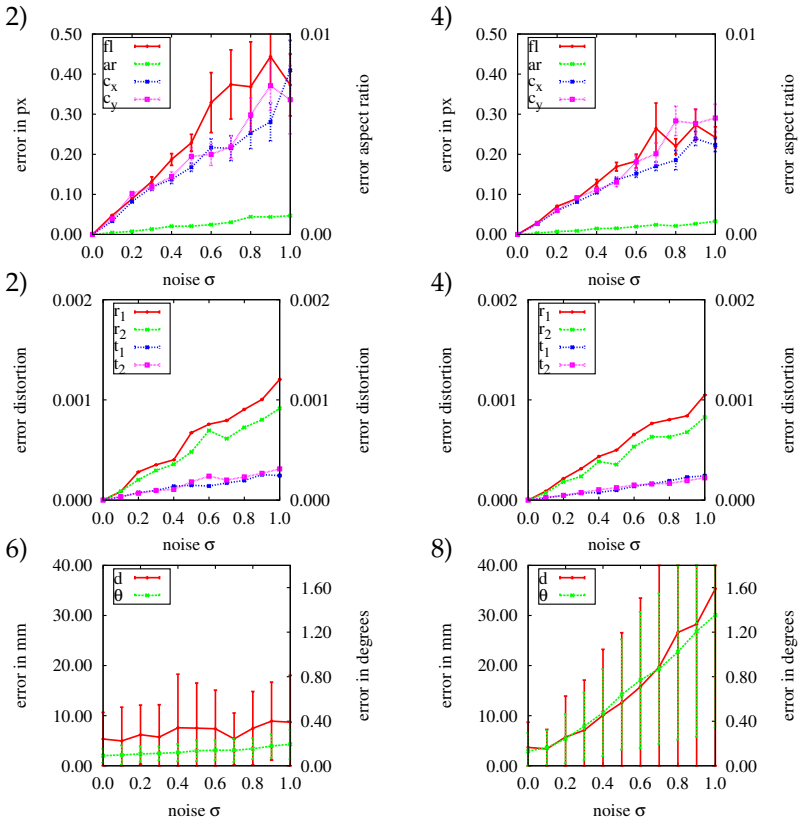

Figure 5.23. First row: BA estimation of intrinsic parameters, on the left axis are focal length and principal point in pixels. On the right axis is the aspect ratio. Second row: BA estimation of distortion. Note that distortion parameters were set to zero for initialization. Third row: results of interface distance and normal estimation (for initialization the interface distance was set to 10 mm in all cases and the interface normal was $\tilde{\mathbf{n}} = (0, 0, 1)^{\mathrm{T}}$.

5.1.5 Structure from Motion

Sequential Approach

During a sequential reconstruction, correspondences are matched between the first two input images using for example SIFT [Wu07]. Then, the relative pose of the second camera with respect to the first is estimated using those correspondences. Once the two camera poses are known, a sparse set of 3D points is triangulated. Even though a RANSAC framework is usually used for estimating relative pose, outliers should be detected and removed after triangulating points. Bundle adjustment is then applied to optimize the two-view scene. After this initialization of the reconstruction, other views are added sequentially, i. e., correspondences between the last view in the reconstruction and the new view are matched. Since 3D points exist for the last view in the reconstruction, this automatically yields a set of 2D-3D correspondences for the new view. Hence, the absolute pose can be used to determine the camera pose of the new view. Note that the error made at the subsequent steps accumulates during the reconstruction, sometimes causing the algorithm to fail entirely before all images have been added. Advantages of sequential SfM are:

- stereo rigs can be easily incorporated by omitting relative pose and directly triangulating 3D points using the two rig cameras and
- intrinsic parameters in the perspective case and housing parameters in the refractive case can be optimized using bundle adjustment at any time.

Hierarchical Approach

A disadvantage of the sequential SfM approach is that errors made in pose estimation tend to accumulate over time, causing the scene to drift. In order to prevent that, Farenza et al. and Gherardi et al. in [FFG09, GFF10] proposed a hierarchical approach. Here, correspondences are computed between all pairs of input images, thus the input images do not have to be ordered. For each image pair that has a certain number of shared 2D-2D correspondences, the relative pose of the second view is estimated using a

Algorithm 5.1 Sequential SfM

match 2D-2D correspondences between I_0 and I_1.
Init: set the camera for I_0 into origin of the world coordinate system
Init: compute relative pose of I_1 using RANSAC
Init: remove outliers
Init: triangulate 3D points using I_0 and I_1
for Images $I_i, 2 < i < n$ **do**
    match 2D-2D correspondences between I_{i-1} and I_i
    compute absolute pose of I_i using RANSAC
    remove outliers
    triangulate 3D points using I_{n-1} and I_n
    remove outliers
    (optional: run bundle adjustment)
end for
(optional: run final bundle adjustment)

RANSAC framework. Then, for each image pair a score called connectivity depending on the number of shared 2D-2D correspondences and the distribution of the correspondences across the images is computed. Image pairs are then ordered according to this score. Starting with the highest ranking image pairs, two-view clusters are formed by triangulating a set of 3D points and optimizing the two-view scene using bundle adjustment. Note that each image can only be in one cluster. After that, two different actions are possible: a new view can be added to an existing cluster using 2D-3D correspondences and absolute pose followed by triangulating new 3D points and bundle adjustment, or two existing clusters can be merged by computing a suitable transform with rotation, translation, and scale for the second cluster based on 3D-3D point correspondences. The algorithm is summarized in 5.2 Using such an hierarchical approach has several advantages compared to the sequential approach:

- the input images do not need to be ordered,
- closed loops in the camera path, i. e., cases were the camera intersects its former path, are automatically detected and contribute to a more stable reconstruction of the camera path with less drift, and

Algorithm 5.2 Hierarchical SfM

Init: compute 2D-2D correspondences between all image pairs
Init: compute relative pose for all pairs I_i, I_j using RANSAC
Init: remove outliers
Compute connectivity for all image pairs $(I_1, I_2)_i$
Compose ordered list L with all image pairs according to their connectivity
while $L \neq \varnothing$ **do**
  if I_1 and I_2 are no part of any cluster **then**
    create new cluster using relative pose results
  end if
  if I_1 part of cluster j **then**
    add I_2 to cluster j using absolute pose
  end if
  if I_2 part of cluster j **then**
    add I_1 to cluster j using absolute pose
  end if
  if I_1 part of cluster j and I_2 part of cluster k **then**
    merge clusters j and k
  end if
  remove outliers
  triangulate new 3D points
  run bundle adjustment
end while
(optional: run final bundle adjustment)

▷ by starting with the best fitting image pairs, instead of the first two images, a possible error made at the beginning of the reconstruction is as small as possible.

When considering scene scale in case of refractive reconstruction, a disadvantage of the hierarchical approach can be revealed. Due to the need to compute the relative pose between all suitable image pairs, scene scale needs to corrected in all those cases and therefore needs to be known before the reconstruction even starts. This can easily be achieved by utilizing navigation data, however, in case the absolute scale is determined

by a known distance in the scene and depends on accurate localization of the corresponding points in all images, fixing the baseline scales becomes more difficult. Therefore, in this thesis, the sequential approach will be used for reconstruction, even though the hierarchical approach is advantageous in general. In case of navigation data or a resolved scaling issue, it would be very interesting to experiment with the combination of the hierarchical approach and refractive geometry estimation.

Experiments

The last sections showed how a reconstruction of a set of input images can be computed using the classic perspective camera model, but also by explicitly modeling refraction of light at underwater housings. In Chapter 3, different existing approaches in the literature were discussed with the conclusion that most existing approaches do not model refraction explicitly, but use the perspective camera model to approximate the effect. In this section, both methods are therefore compared by applying them to image sets. This is done using synthetic images rendered using the simulator described in Section 3.3, as well as real images captured in a fairly well controlled lab environment.

Simulated Images

Two sets of synthetic images of different scenes were rendered with different housing configurations, where the interface distance was chosen from $d \in \{-5\,\text{mm}, 0\,\text{mm}, 5\,\text{mm}, 10\,\text{mm}, 20\,\text{mm}, 50\,\text{mm}, 100\,\text{mm}\}$ and the interface tilt was $\theta_2 \in \{0^\circ, 0.5^\circ, 1.0^\circ, 3.0^\circ\}$, resulting in a total of 28 configurations for testing. The error was determined by comparing the resulting 3D points and camera poses to ground truth geometry data and determining the average error over all pixels and images. In addition, another data set was rendered using a stereo rig with a denser sampling of interface distances and tilts ($d \in \{-10\,\text{mm}, 0\,\text{mm}, 10\,\text{mm}, 20\,\text{mm}, ..., 140\,\text{mm}\}$, $\theta_2 \in \{0^\circ, 1.0^\circ, 2.0^\circ, 3.0^\circ\}$) with 68 different configurations. Figure 5.24 summarizes rendering paths, scene structure, and camera-point distances for all three scenes. To all image sets, the perspective and the refractive SfM algorithm is applied, thus, the accuracy of estimating the camera

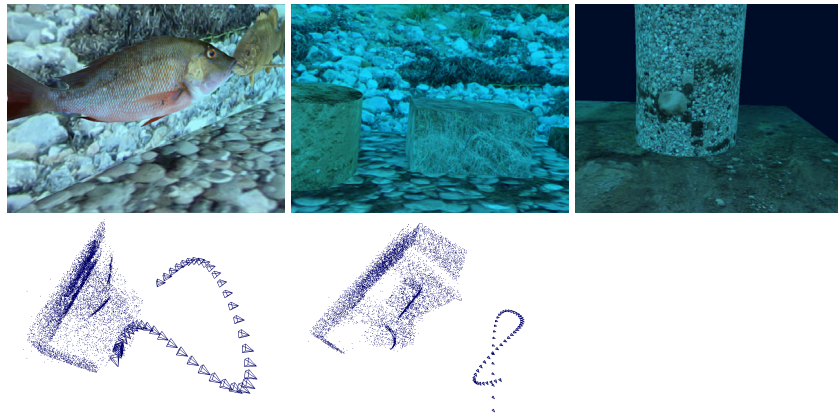

Figure 5.24. Top row: exemplary images from the rendered scenes. Bottom row: scene structure and camera trajectory. Note that the scenes differ not only in structure and camera path, but also in camera-object distance, i. e., in the first scene the closest views have camera-object distances between 550 mm and 1900 mm, while the furthest views have 1300 mm-2300 mm. The second scene is larger, such that the closest views have camera-object distances between 4600 mm and 9000 mm and the furthest have 7700 mm-12000 mm. In the third (stereo) scene, the camera was moved in an approximate orbit around the scene, hence the camera-object distances were almost constant for all views (3000 mm-6000 mm).

poses and 3D points can be investigated. Figures 5.25, 5.26, and 5.28 show the resulting errors depending on interface distance d and interface tilt θ. Depicted are the 3D error, measured in mm using the known ground truth depth maps, the camera translation error in mm and the reprojection error for the perspective case (left columns) and the refractive case (right columns). In all three cases, the systematic model error introduced by using the perspective camera model to approximate refractive effects is clearly increasing with increasing interface distance and tilt. This is in accordance with the results in Chapter 4, where it was demonstrated using the depicted stereo data set, how the caustic sizes increase with increasing interface distance and tilt, thus showing an increasing deviation from the single-view-point camera model. Figure 5.27 gives results for the refractive reconstruction on the box sequence without correction of

the scene scale after relative pose estimation. As can be seen, the errors are larger compared to the refractive reconstruction with correction of scale (Figure 5.26), however, the error is not systematic as in case of the perspective reconstruction and also lower. This is in accordance with the results depicted in Figure 5.9, where it was demonstrated that the different non-linear error functions cannot be used to estimate scale, but are not invariant against scale changes either. Note that for all other SfM results, scene scale was corrected after relative pose estimation.

Real Images

The refractive and perspective algorithms were also compared using real images captured in a fairly controlled lab environment. A fish tank of the size 500 mm × 1000 mm × 500 mm was filled with water and a pair of cameras (Point Grey firewire) were placed outside the tank viewing the inside, simulating an underwater housing. A disadvantage of this set-up is that the cameras are not allowed to move with respect to the glass, i. e., once calibrated, the scene inside the tank needs to be moved instead of moving the cameras around the scene. Therefore, the scene consisted of a model of the entry to the Abu Simbel temple in Egypt (Figure 5.29, top left), the size of which was approximately 380 mm × 280 mm × 180 mm. It was rotated around its vertical axis in front of the cameras at distances between 300 mm and 750 mm. As can be seen in Figure 5.29 in the upper left image, the scene was mirrored at the bottom of the tank. Additionally, the tank itself, but especially small gas bubbles at the tank walls violated the rigid scene assumption. Therefore, all input images needed to be roughly segmented.

Images were captured using seven different camera-glass configurations (compare to Figure 4.11). Corresponding results of refractive calibration have already been presented in Chapter 4. Figure 5.29 shows the results of the perspective (red) and refractive (blue) SfM algorithm. It is difficult to determine the ground truth camera poses and, therefore, measure an absolute error as in case of the synthetic data. However, it can be seen that in cases a) and f) the perspective reconstruction failed completely. Additionally, Table 5.3 shows the average differences and standard deviation in mm between refractive and perspective camera poses.

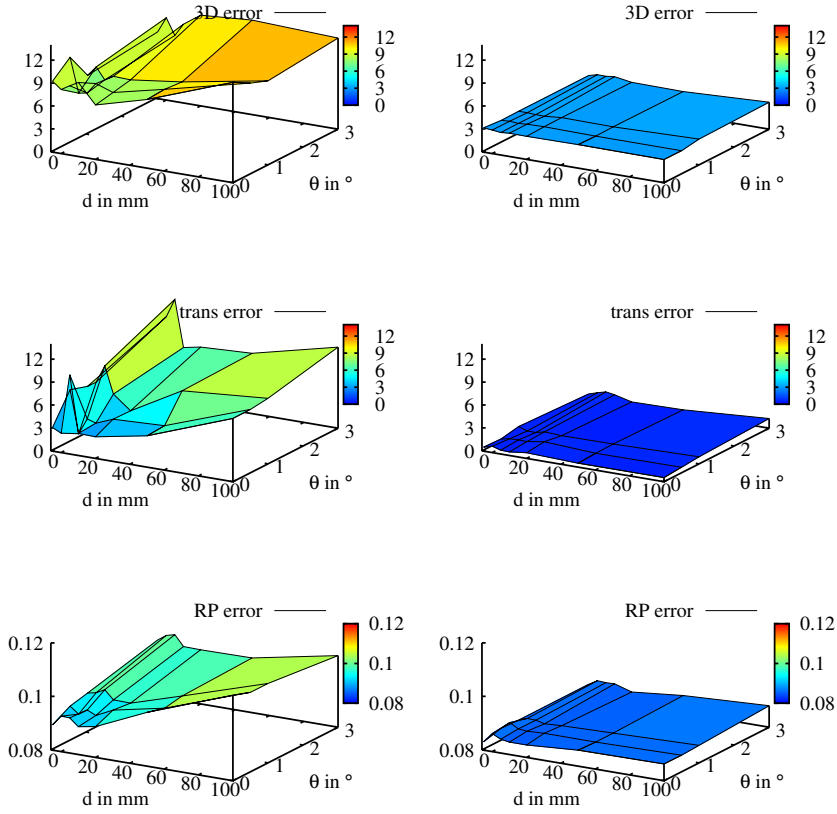

Figure 5.25. Results of monocular SfM on fish sequence. Left column: perspective camera model on underwater images. Right column: refractive camera model on underwater images. From top to bottom: 3D error in mm, camera translation error in mm, and reprojection error in px.

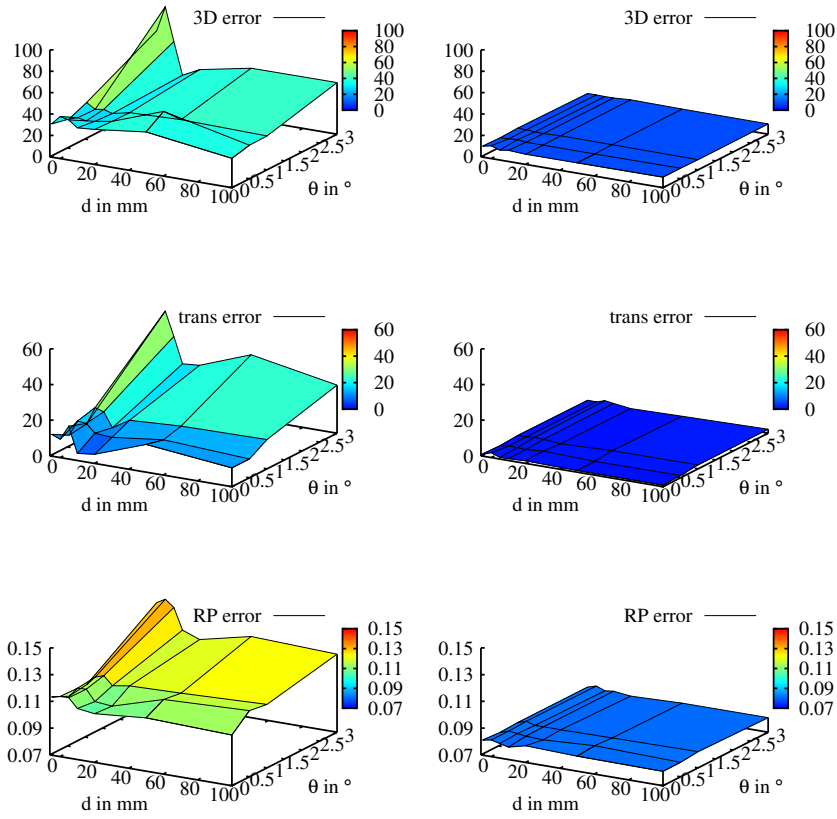

Figure 5.26. Results of monocular SfM on box sequence. Left column: perspective camera model on underwater images. Right column: refractive camera model on underwater images. From top to bottom: 3D error in mm, camera translation error in mm, and reprojection error in px.

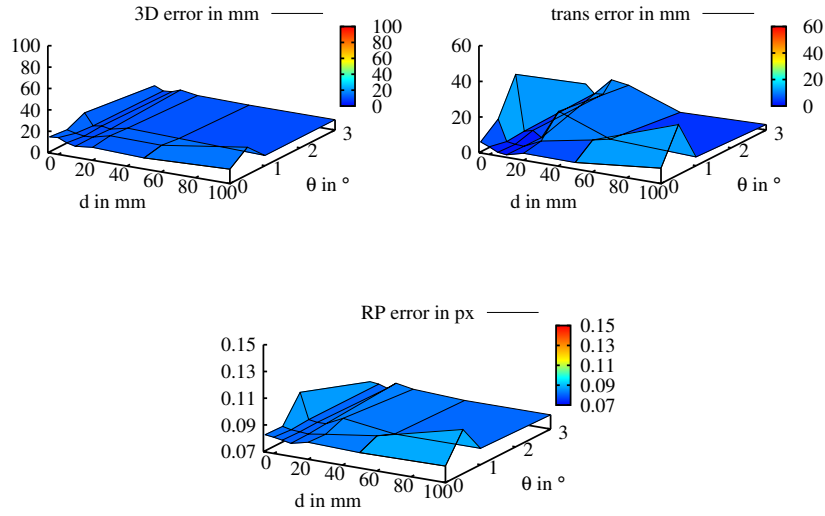

Figure 5.27. Refractive results of monocular SfM on box sequence without scene scale correction after relative pose estimation. Upper left: 3D error in mm, upper right: camera translation error in mm, and bottom: reprojection error in px. Compare these results to the right column in Figure 5.26

It can be seen that larger interface distances and tilt angles tend to cause larger differences in camera translation between perspective and refraction reconstructions, indicating the influence of the systematic model error.

In summary, it can be said that the use of the perspective camera model causes a systematic model error, which can be eliminated by modeling refraction explicitly. Even though the errors demonstrated here do not seem to be large, one has to keep in mind that usually longer camera trajectories are reconstructed, which causes the model error to accumulate over time. Additionally, the failures of the perspective reconstruction on Abu Simbel data demonstrates that perspective reconstruction on refractive data is not always possible.

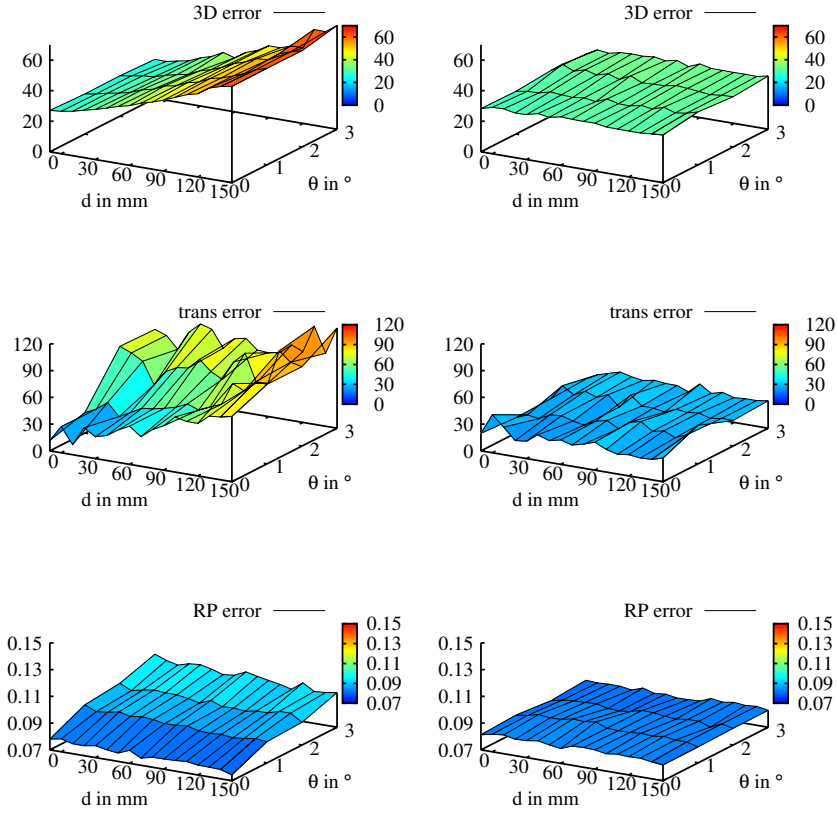

Figure 5.28. Results of monocular SfM on orbit sequence. Left column: perspective camera model on underwater images. Right column: refractive camera model on underwater images. From top to bottom: 3D error in mm, camera translation error in mm, and reprojection error in px.

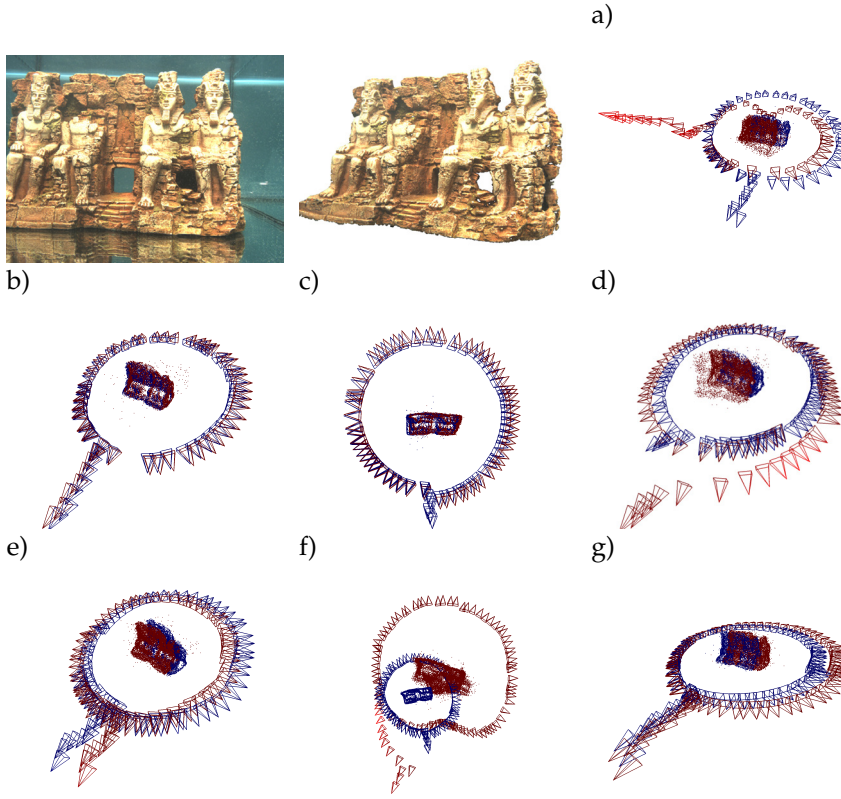

Figure 5.29. The first image shows an exemplary input image. Due to the mirrored scene in the tank bottom and other features like small air bubbles on the tank walls, all input images have been roughly segmented (second image). Images a) to g) show reconstruction results for seven different camera-glass configurations (Figure 4.11). Blue is the camera trajectory and point cloud from the refractive reconstruction, red is from perspective reconstruction (refer to Table 5.3 for differences in mm between perspective and refractive results).

Table 5.3. Average distance and standard deviation between perspective and refractive camera translations for seven different camera-glass configurations. Note the large differences in trials a) and f), which are cases, where the perspective reconstruction failed (compare to Figure 5.29).

Trial	#imgs	d in mm	θ in $^\circ$	$\varnothing in$mm	sd in mm
a	46	7.88	0.34	350.879	312.876
b	52	10.60	0.25	24.791	4.423
c	67	51.95	0.29	26.4426	14.4046
d	65	61.47	7.36	186.571	82.5256
e	76	76.96	29.29	115.596	31.4714
f	87	95.45	0.12	609.384	194.478
g	79	149.39	0.12	79.5105	37.9085

5.2 Multi-View-Stereo and 3D Model Generation

Once the SfM algorithm determined a sparse 3D point cloud and all camera poses, a dense 3D model can be created. One approach to that is an adaptation of PMVS (Patch-based Multi-view Stereo) [FP10] to explicitly incorporate refraction, which has been done in [KWY12b] for two-view scenes. In order to determine a dense scene, features are detected and matched and for each successful match, a 3D patch is created. Holes between the patches are then filled by searching more 2D-2D correspondences between the images, along the epipolar lines in the perspective case. However, in the refractive case, searching along the epipolar lines is not possible. Therefore, the whole image space needs to be searched or epipolar curves need to be determined. Additionally, the method requires a lot of projections of 3D points into the images, thus the method is infeasible for more than ten images. In this thesis, it is therefore proposed to compute dense depth maps for each image as a reference image, which are then merged into a 3D model. This approach to compute dense depth maps has already been published in [JSJK13].

5.2.1 Refractive Plane Sweep

In order to compute dense depth maps for each image as a reference image with explicit incorporation of refraction, several constraints need to be considered for algorithm design. First of all, epipolar geometry is invalid, hence, images cannot be rectified (refer to [Sze11]), which eliminates all established methods relying on rectification for multi-view stereo, e. g., [Hir01]. As described in Chapter 3, Gedge et al. [GGY11] showed how to approximate refractive epipolar curves piecewise linearly, however, in the course of their method many projections of 3D points into refractive images are required, making the method infeasible for long image sequences.

Therefore, the proposed method is a refractive plane sweep (refer to [Sze11] for an introduction to the classic Plane Sweep algorithm), which does not rely on rectified images or epipolar geometry. The idea of the plane sweep algorithm is to sweep a set of hypothesis planes through space in front of the reference camera for which the dense depth map is to be computed (Figure 5.30). For each pixel in the reference camera, a patch around the pixel is compared to the corresponding patch from other images determined by the depth hypothesis, thus yielding a cost value for each depth hypothesis. The lowest cost determines the final depth. In case of perspective cameras, a homography is used to efficiently warp entire images from one image into another using the depth hypothesis plane. This allows to quickly compare all image patches. However, due to the refractive camera being an nSVP camera, such homographies for warping are invalid. Additionally, Section 3.2.5 showed that the projection of 3D points into images is very time-consuming in case of refractive cameras. Therefore, replacing homography warping by a combination of back-projection and projection for each pixel in each image and for each hypothesis plane in case of refractive cameras is infeasible.

Considering all these constraints, the following method is proposed. Instead of comparing patches in the reference image, comparison is done by projecting images on the hypothesis planes, which can be achieved by the following steps for each hypothesis plane:

- project the image corners of the refractive image onto the hypothesis plane, thereby defining the 3D image corners of what will be called *plane image* in the following,

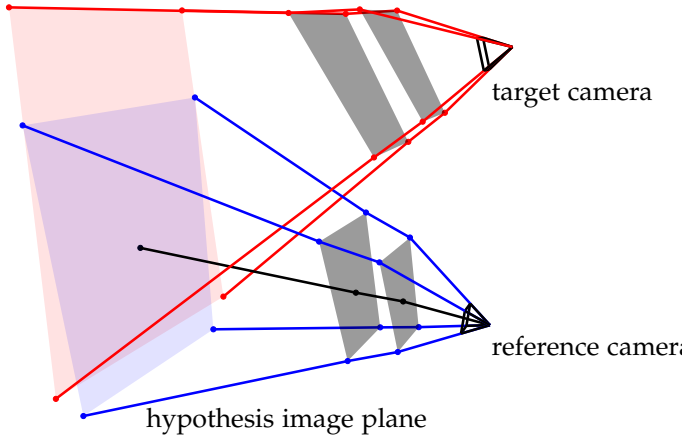

Figure 5.30. In the proposed plane sweep, the images from reference and target view are mapped onto the hypothesis plane, where the corners of the reference view (blue) determine image boundaries. Adapted from [JSJK13].

- apply forward-mapping from the reference image to the plane image, by projecting all pixel positions onto the plane image in order to determine colors for each plane image pixel,
- apply forward-mapping to the target view by projecting all pixel positions of the second image onto the plane image defined for the reference image, hence determining the image color for each pixel of the second plane image, and
- for each pixel in the reference image, compare the two corresponding patches in the two plane images.

Since the described plane image computation is a forward-mapping, the resulting plane images can be incomplete, i. e., contain holes. In order to get an efficient implementation of the proposed algorithm, it is implemented on the GPU, which allows not only to efficiently compute all back-projections using shaders, but also to fill all resulting holes by interpolation. GPU shaders are then used to compare corresponding image patches. In a first sweep, Normalized Cross Correlation (NCC) is used to measure similarity. Is invariant against changes in lighting caused by

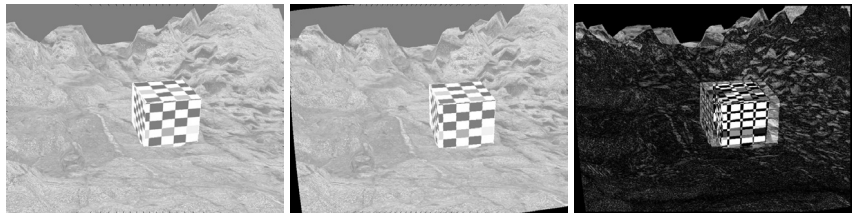

Figure 5.31. Exemplary plane images of a rendered scene. From left to right: plane image of reference image, plane image of the target image, and difference image between the two plane images.

the water. Then, in a second sweep, the depth hypothesis can be used to correct image colors in the warped images using the model for color degradation in underwater images (3.1.19) and the Sum of Absolute Differences (SAD) can be used in a shiftable window approach to compute similarity. Good results from the first NCC-sweep are used as weights for the matching costs in the second sweep and improve robustness. In case NCC cannot find a clear minimum in the cost volume, SAD can distinguish finer structures, because it can be implemented as a separable filter and, therefore, can run efficiently in a shiftable window approach. Consequently, the second sweep is used to improve results from the first sweep and fill possible holes in the dense depth map. In order to get more accurate and robust results, the described plane sweep was used on three images simultaneously, i. e., the middle image was the reference image and the other two were used to determine depth. Thus, a three-view refractive plane sweep scheme is defined that can be used to determine accurate dense depth maps for each camera with known pose. The required minimal and maximal depth values for the sweeping distance are acquired from the sparse 3D point cloud result from SfM. Note that after computing starting point and direction for each pixel, the underlying refractive camera model is not utilized in the remainder of the method. This makes it possible to apply the proposed method to other, general camera models as long as a ray with starting point and direction can be computed for each pixel.

Once all refractive depth maps have been computed, they are fused into a 3D surface model that can be textured. In order to compute the depth

map fusion, 3D points need to be projected into the images for each pixel in each image, an operation that is inefficient with the refractive camera model. However, as described in Section 2.2.3, distortions of nSVP cameras can be corrected if depth is known. Consequently, using each refractive dense depth map, it is possible to determine a corresponding perspective dense depth, where refractive distortions are corrected. Note that the resulting perspective depth map and texture are not an approximation, but that due to known scene depth for each pixel, an exact, but perspective depth map and texture can be computed. The set of perspective depth maps and textures for all cameras can be used to create the final 3D model. All depth maps are fused by detecting outliers by the voting scheme proposed in [PDH+97]. Then, the Marching Cubes algorithm [LC87] can be applied to build a triangle mesh, which can be textured to get the final 3D model. The transformation of refractive depth maps to perspective depth maps does not change the resulting model, but eliminates the need of costly refractive projections of 3D points during depth map fusion, thus preventing the approach from becoming infeasible.

5.2.2 Experiments

The described plane sweep algorithm has been tested on synthetically rendered data and on real images captured in an the above described water tank.

Synthetic Data

In order to test the described plane sweep algorithms on synthetic data, the two first data sets that have already been used for experiments with the refractive SfM are used.

Figures 5.32 and 5.33 show results of the described three-view plane sweep algorithm on synthetic images with known ground truth camera poses. The proposed method performs better than using the perspective camera model on underwater images and is completely invariant against changes in the underwater housing configuration.

The extent of the systematic error introduced by the perspective camera model can be observed in Figure 5.34, where the resulting depth maps are

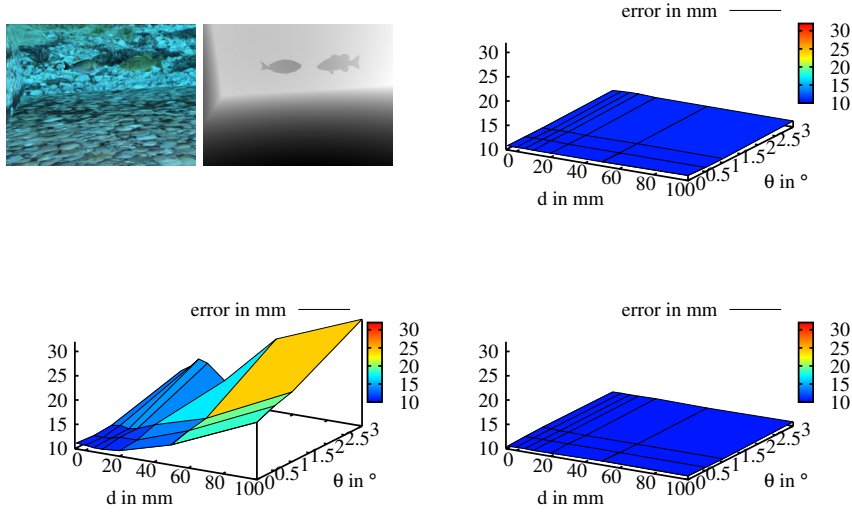

Figure 5.32. Refractive plane sweep results. Results for a close scene with distances up to 4000 mm. From left to right, top to bottom: exemplary input image and ground truth depth map of scene, results of perspective model on perspective images, results of perspective camera on underwater images, and results of refractive camera on underwater images. Reprint from [JSJK13].

shown in the top row. The result for using the perspective camera model on underwater images is clearly less accurate than the result for using the refractive camera model. The extent of the error becomes particularly clear, when observing the bottom row, where the pixel-wise difference maps with the ground truth depth map are shown. The planar surface in the back ground was reconstructed with a systematic error of up to 200 mm in case of the perspective plane sweep and about 20 mm in the refractive case. Clearly, the accuracy of the perspective method depends on the pixel position in the image and the camera-object distance.

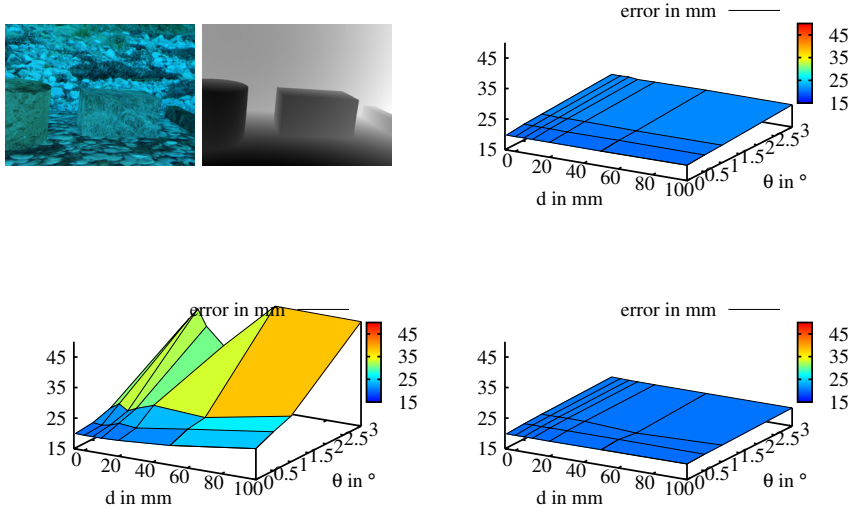

Figure 5.33. Refractive plane sweep results. Results of a scene with distances between 4000 mm-9000 mm. From left to right, top to bottom: exemplary input image and ground truth depth map of scene, results of perspective model on perspective images, results of perspective camera on underwater images, and results of refractive camera on underwater images. Reprint from [JSJK13].

Real Data

Figure 5.35 shows some exemplary results of depth maps created using the described refractive plane sweep algorithm. As can be seen, the depth maps for both cases, the refractive and the perspective accurately show the scene. Due to the missing ground truth data, the camera poses used for the plane sweep algorithm are the results from SfM. Consequently, they already contain the systematic model error introduced by using the perspective camera model on underwater data. Unfortunately, it is not possible to quantitatively analyze the error, however, the last column in Figure 5.35 shows difference images between the refractive and the perspective depth maps and it is clear that the differences increase with

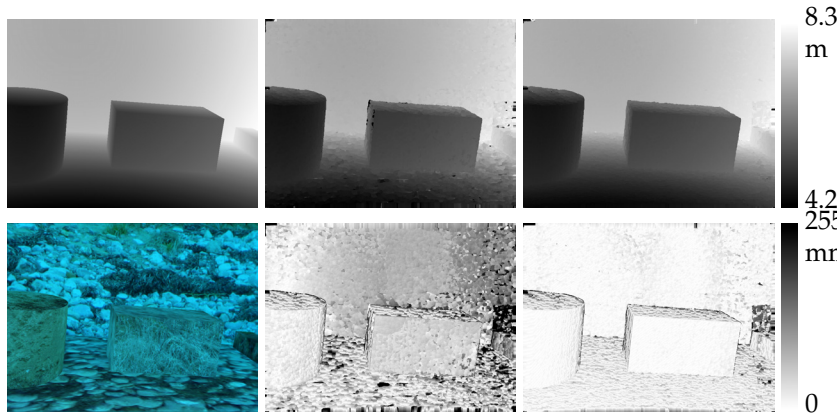

Figure 5.34. Exemplary result of refractive plane sweep with housing configuration $d = 100\,\text{mm}$ and $\theta_2 = 3°$. Top row: ground truth depth map, resulting depth map using the perspective model, and resulting depth map using the refractive model. Bottom row: input image, pixel-wise difference to ground truth for perspective result, and pixel-wise difference to ground truth for refractive result. reprint from [JSJK13].

increasing camera-object distance.

After determining the depth maps, a textured 3D model can be computed. Figure 5.36 shows the perspectively (red) and refractively (blue) computed 3D models for test case g). Note that both camera models yield plausible reconstructions (top row) with seemingly valid geometry. However, when rendering both models at the same time (bottom row), it becomes clear that they have a slightly different size and different positions in space even though the first cameras in both cases are placed at the world coordinate system origin.

5.3 Summary

In this chapter, the main contribution of this thesis was presented, a complete system for refractive 3D reconstruction. Components that are required to be adapted to refraction are relative and absolute pose, non-

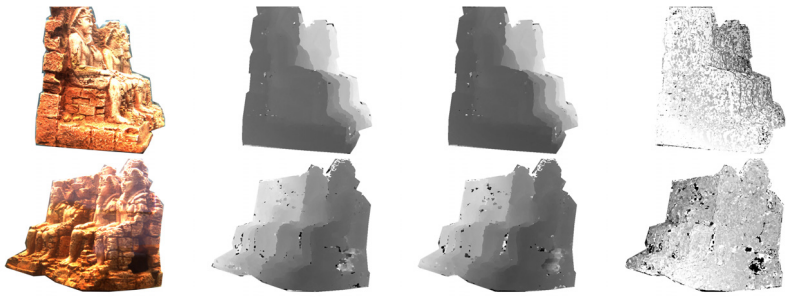

Figure 5.35. Depth estimation results for Abu Simbel model. Top: results for sequence one. Botton: results for sequence two. From left to right: input image, resulting perspective depth map, resulting refractive depth map, negated, pixel wise, difference between perspective and refractive depth maps with differences between 25 mm - 33 mm for the first sequence and differences on the model between 15 mm - 27 mm for the second sequence. Reprint from [JSJK13].

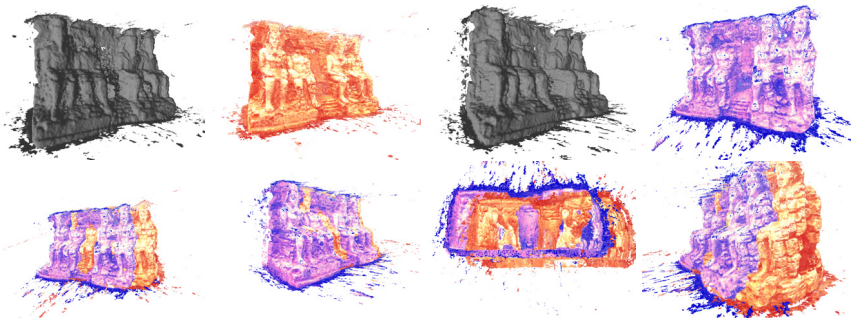

Figure 5.36. Abu Simbel model for case g) (refer to Figure 5.29). Top row from left to right: untextured perspective model, textured perspective model (in red), untextured refractive model, textured refractive model (in blue). Bottom row: different views of both models rendered together (red: perspective model, blue: refractive model).

linear optimization, especially in the context of bundle adjustment, and dense depth estimation. In order to develop a time-efficient system, no 3D-to-2D projections can be used in any component. Experiments compared different methods for relative and absolute pose and a combination with a good accuracy and run-time relation was chosen for the final system. Further experiments with the final system demonstrated that the perspective algorithm suffers indeed from a systematic model error, when being applied to underwater images, which can be eliminated successfully by explicitly modeling refraction during SfM. The same was shown for the proposed refractive plane sweep algorithm.

Chapter 6

Applications

The described methods and concepts for dealing with refraction of light at underwater camera housings have various different applications. This is mainly due to the increasing popularity and availability of underwater cameras, but also because of the increasing interest in resources being mined on or below the seafloor. Additionally, many objects of scientific interest can be found in the water or on the seafloor. In this chapter, it will be demonstrated how the methods proposed in Chapters 4 and 5 can be applied in the areas of Geology and Archaeology.

6.1 Geology

In the area of Geology or more specifically Volcanology, it is of interest how underwater volcanoes came into existence, for example a specific volcano that was found near the Cape Verdes [KHDK13]. Unfortunately, the volcano was found at a depth of 3500 m. Thus, the ROV Kiel 6000 was used to examine it by capturing a dense HDTV video sequence and a set of images using a stereo rig of SLR cameras. As already mentioned in the introduction, having to rely on such image data greatly hampers the way geologists usually do their fieldwork. It is very difficult to get an impression of the whole volcano when studying hours of video, where the ROV had to navigate along the volcano flank at a very close distance. By applying the reconstruction methods proposed in this thesis, parts of the volcano can be modeled. This allows to detect geological structures like joints [KHDK13] due to the possibility to interactively view the model, e. g., to veer away from the closely captured flank in order to gain a better overview. Input images, captured using the ROV's HDTV camera and the final result are shown in Figure 6.1. The underwater camera used for

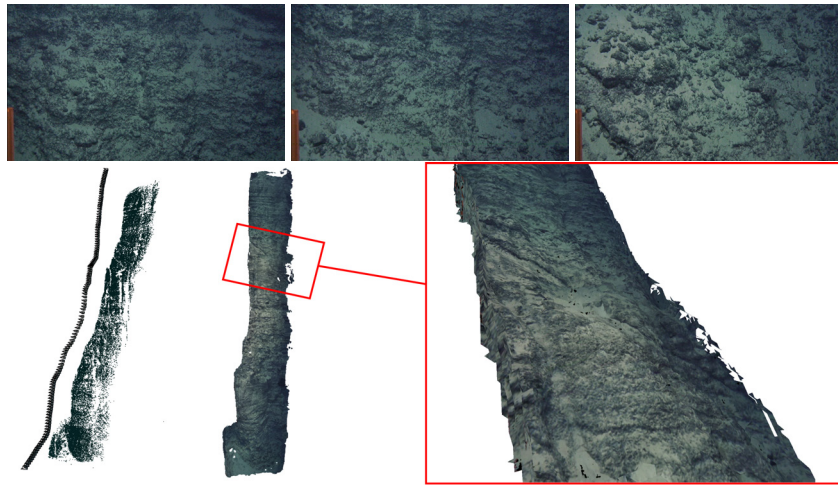

Figure 6.1. Top row: exemplary input images chosen as example from 2700 images captured of an underwater volcano near the Cape Verdes at approximately 3500 m water depth. Every 25^{th} image was used for the reconstruction. Bottom row from left to right: reconstructed point cloud with camera path, reconstructed 3D model with exemplary detected geological feature, here a joint (within marked in red rectangle) according to [KHDK13]. Input images from Geomar Helmholtz Centre for Ocean Research.

capturing the images was a deep sea HDTV video camera, i. e., the housing glass was several centimeters thick. Unfortunately, it cannot be removed from its housing, thus, calibration in air without the glass is impossible. In order to reconstruct the model, the camera was calibrated perspectively using checkerboard images captured in air, thus, the thick glass was approximated using radial distortion. Then, based on underwater images, the camera was calibrated refractively. Additionally, grain sizes can be measured.

Figure 6.2 shows reconstruction results of parts of the volcano that have been reconstructed refractively using the camera calibration shown in Chapter 4. As can be seen in the input images, a lot of image regions exhibit poor contrast due to bad lighting or homogeneous sand, where the depth maps cannot be computed reliably. This problem can possibly be

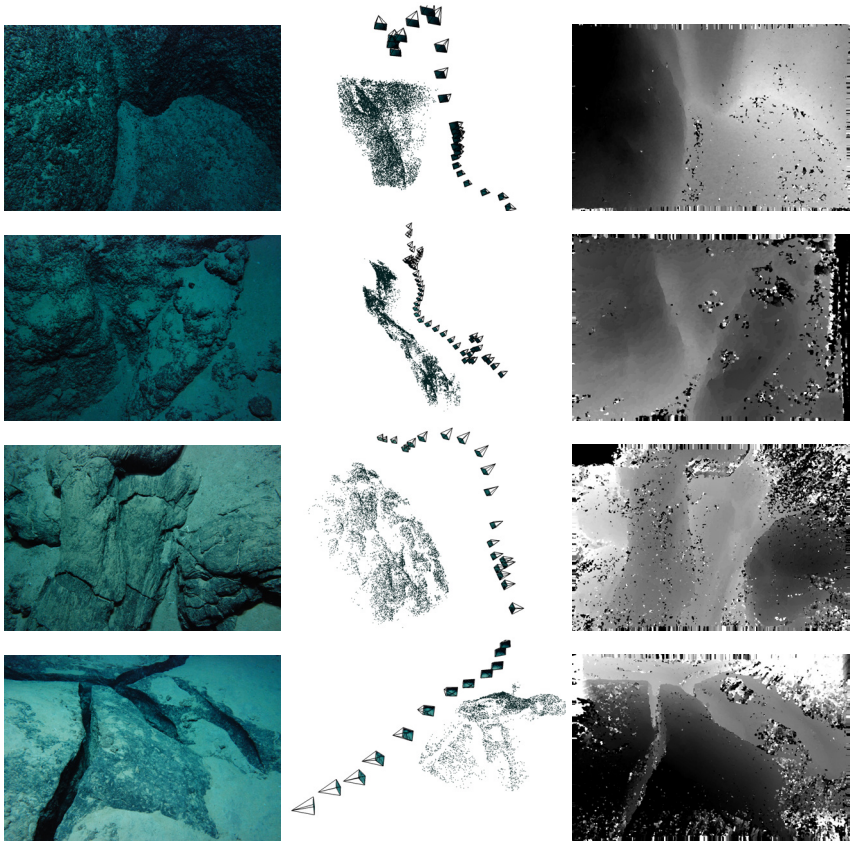

Figure 6.2. Depicted are an exemplary input image, the 3D point cloud and camera path, and the depth map for four sequences showing parts of the Cape Verdes underwater volcano. Note that there are image regions with very low contrast due to darkness and sand, where the depth maps cannot be computed reliably. Input images from Geomar Helmholtz Centre for Ocean Research.

solved by using a more robust method for finding the minimum depth in the cost volume than the applied local patch comparison.

A second example can be found in the study of black smokers, hydrothermal vents in the ocean floor. Black smokers can for example be

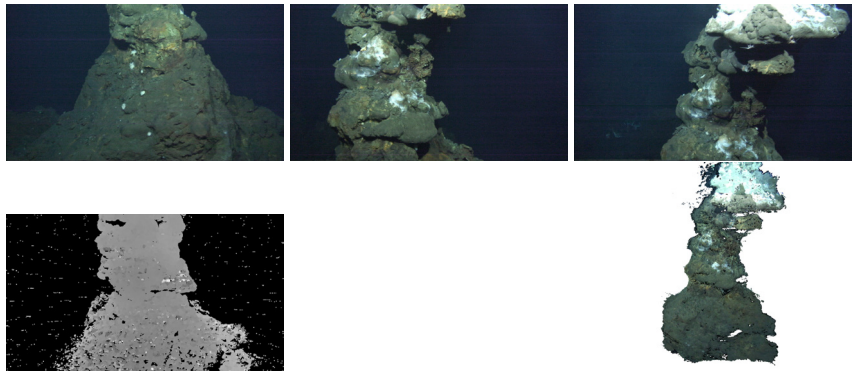

Figure 6.3. Black smoker reconstruction results. Top row: exemplary input images. Bottom row from left to right: exemplary dense depth map, 3D point cloud with camera path, and textured 3D model. Input images from Geomar Helmholtz Centre for Ocean Research.

found at the middle Atlantic ridge, where new oceanic crust is produced and magma chambers lie shallow beneath the seafloor. Above such magma reservoirs, seawater percolates through the heated crust and is heated in turn, causing an exchange of minerals and metals with the surrounding country rock. When the water enters the sea through hydrothermal vents, it has been heated to several hundred degrees Celcius, but is still liquid because of the high water pressure. Due to the contact with the cold seawater, the minerals and metals spontaneously precipitate ("black smoke") and slowly accumulate around the seepage: a black smoker develops. Black smokers are of great interest to Geologists for several reasons. They are explored as possible deposits of minerals and metals, they serve as a unique habitat for chemosynthetic communities despite the great water pressure, and they are surprisingly common: on average, one such vent field can be found every one hundred kilometers along the axis of the global mid-oceanic ridge system [BG04]. 3D models allow to measure their size, but also to determine their volume, a characteristic, which is subject to constant change. Figure 6.3 shows exemplary reconstruction results of a black smoker. The input images were captured using the ROV's Kiel 6000 HDTV video camera.

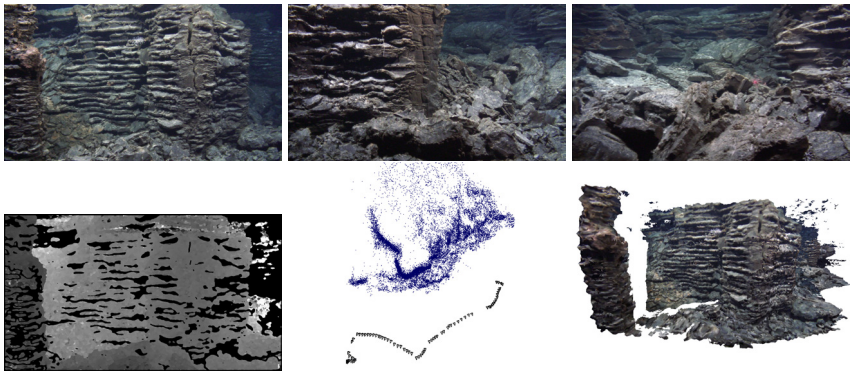

Figure 6.4. Reconstruction of an inside wall of a lava lake. Top row: exemplary input images. Bottom row from left to right: exemplary dense depth map, 3D point cloud with camera path viewed from the top, and textured 3D model. Input images from Geomar Helmholtz Centre for Ocean Research.

Figure 6.4 shows the reconstruction of the inside wall of a lava lake, where decreasing lava levels caused the edges to tear horizontally. It was also found at the middle Atlantic ridge at at $4^\circ 48'\mathrm{S}, 12^\circ 22.5'\mathrm{W}$.

6.2 Archaeology

Typically, the process of archaeological excavations involves a complex documentation routine because the scientists need to dig to deeper layers, which requires to remove everything that has been found in the current layer from the scene. Even in this scenario (in air), the computation of 3D reconstructions is gaining popularity [WSK13] due to the possibility to measure object sizes even long after the excavation and to document the finds. Whenever such a find is below the water surface on the seafloor or lake bottoms, archaeological excavations become far more difficult. Depending on the water depth, either specially trained, scientific divers or ROVs are required and in both cases diving time is limited. Especially in case of larger finds, it is very difficult to convey the find's significance and layout to other scientists or even to the general public. For all of

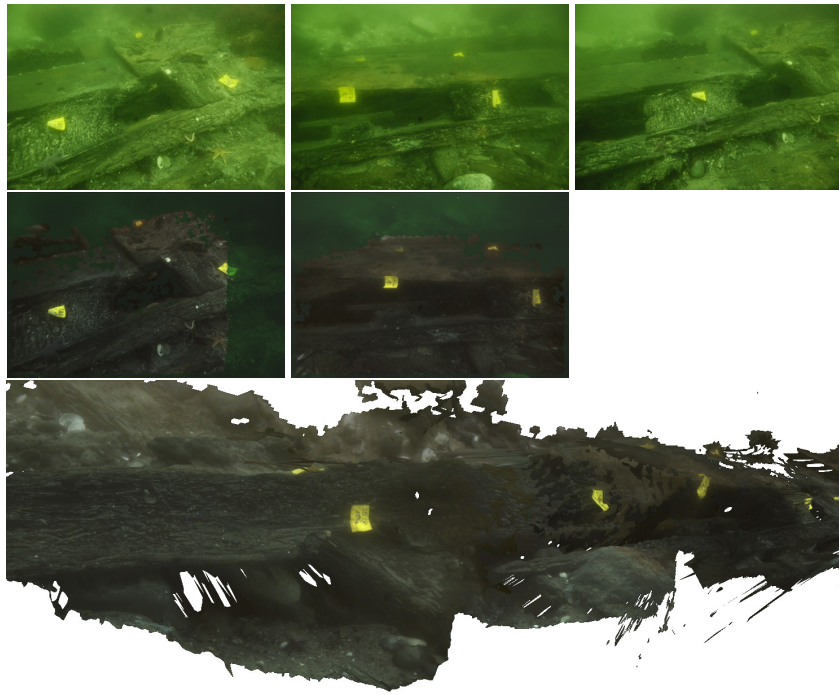

Figure 6.5. Reconstruction of the Hedvig Sophia shipwreck. Top: exemplary input images. Middle: color corrected images and depth map. Bottom: resulting model with color correction. Input images by Florian Huber.

these reasons, the computation of digital 3D models of the finds or even of different layers during the excavation is of great interest. An example is the shipwreck of the Hedvig Sophia, which was found in the Baltic Sea in the Kiel fjord, Germany. In this case, the water depth was about 5 m, so divers were able to examine the wreck in person. The images were captured with an SLR camera, confined in a hand-held underwater housing with a dome port. Since dome ports cause less or even no refractive effects in ideal settings, no algorithm with explicit consideration of refraction at dome ports was developed in this thesis. Therefore, the ship wreck was reconstructed using the traditional perspective approach. However,

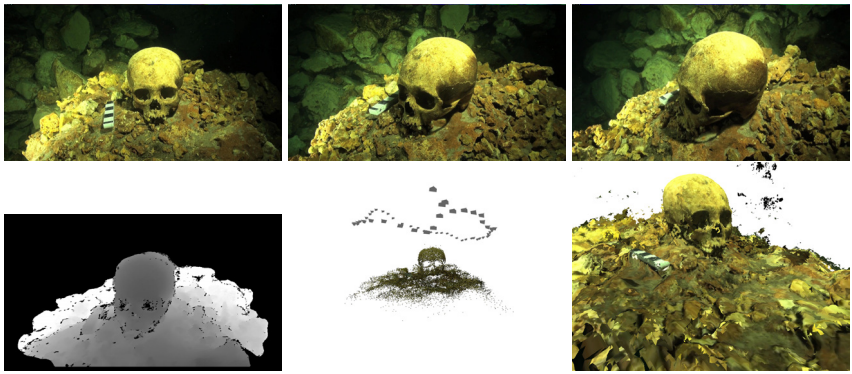

Figure 6.6. Reconstruction results for a human skull. Top row: exemplary input images. Bottom row from left to right: exemplary dense depth map, 3D point cloud with reconstructed camera path, and screenshot of textured 3D model. Input images by Christian Howe.

Table 6.1. Calibration results for dome port camera with the dome being a lens that corrects the refractive effect.

	air	water
focal length	605.92	616.75
princ. point	$(484.29, 258.41)^{\mathrm{T}}$	$(489.78, 290.86)^{\mathrm{T}}$
rad. dist	-0.1331, 0.0276	-0.0899, -0.0529

interesting results concerning image color correction can be presented. In this case, the viewing distance in the Baltic Sea was less than 1 m. The images show a strong green hue. Even though, texture colors on the final model (Figure 6.5) were corrected successfully using the simple model for color correction introduced in Section 3.1.3 in combination with the proposed calibration routine (Chapter 4).

The input images of the skull and the sloth in Figures 6.6 and 6.7 were captured in even more difficult circumstances deep inside a cave system in Yucatan, Mexico. In this case, the scientific divers needed a lot of time, and hence gas, to even reach their objects of interest. They were not allowed to remove any finds from the cave system and did not have enough time for a

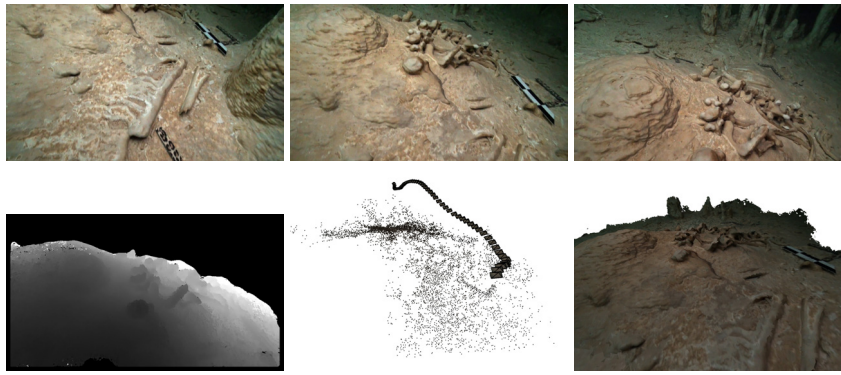

Figure 6.7. Results of sloth bones reconstruction. Top row: exemplary input images. Bottom row from left to right: dense depth map, 3D point cloud with camera path, and textured 3D model. Input images by Christian Howe.

detailed documentation and investigation. Apart from having to reach the find in a labyrinthine cave system and to find the way back out, staying at the scene of interest for longer than a few minutes caused the exhaled gas to congregate at the cave roof, which in turn caused sediments to trickle down, and hence to deteriorate visibility. Consequently, the possibility to reconstruct 3D models of the find were a huge asset in the documentation process, allowing for example Anthropologists to examine the bones in detail even though they were not able to dive in person.

Note that in both cases the camera also had a dome port, this time with the dome being a lens and thus correcting most of the refractive effect such that the perspective model was not required to even compensate refraction with focal length, radial distortion and principal point. Thus, the perspective model can be used for reconstruction, theoretically based on a calibration using checkerboard images captured in air. Calibration results are shown in Table 6.1. The underlying checkerboard images were captured in air and below water and calibrated perspectively. All intrinsic calibration results, excepting the principal point in y-direction are very similar. Note that both models were reconstructed using the hierarchical method described in Section 5.1.5.

6.3 Summary

This chapter pointed out a set of different applications for the methods and concepts introduced in this thesis. The 3D reconstruction capabilities can be utilized for different applications in the areas of Geology and Archaeology.

Chapter 7

Conclusion and Future Work

Due to readily available off-the-shelf underwater camera systems, but also custom made systems in the area of deep-sea robotics, increasing amounts of images or video footage are captured underwater. These images are utilized in a multitude of applications that often profit from capabilities in the area of computer vision. Therefore, this thesis investigated how water affects image formation and, thus, methods from the area of computer vision, in this case Structure-from-Motion (SfM) and dense stereo. The two major effects on image formation, color attenuation and refraction, were described along with the current state-of-the art in the literature. Especially the geometric effects of refraction at the underwater housing were found to require adaptions of classic computer vision methods like SfM and dense stereo. This is due to the perspective camera model being invalid, when capturing underwater images, i. e., causing a systematic model error. The main contribution of this thesis is therefore the elimination of this systematic model error, by explicitly modeling refraction in a 3D reconstruction approach. Color attenuation is a wavelength – and depth – depending function and can be corrected if scene distance is known. Chapter 3 discusses this topic. Chapter 4 is concerned with a refractive calibration approach, allowing to calibrate interface distance and a possible interface tilt of underwater camera housings. A detailed analysis demonstrates the accuracy of the proposed method, but also reveals how and to what extent the perspective camera model can absorb part of the model error introduced by refraction. However, synthetic tests with a stereo camera rig also show that the perspective approximation is unsatisfactory for underwater distance measurements.

Refractive housing calibration is an essential part of refractive SfM, which is developed in detail in Chapter 5. Proposed algorithms include

methods for estimating relative and absolute pose and are compared to methods previously published in the literature. Particularly important is the fact that the projection of 3D points into images is computationally very expensive using the refractive camera model. Therefore, a new error function for non-linear optimization was proposed that eliminates the need for point projection, thus allowing to efficiently optimize the scenes to be reconstructed. Experiments show for the first time that not only does a systematic model error exist when using the perspective method, but that explicitly modeling refraction eliminates the systematic error.

Finally, a method for a refractive plane sweep is introduced, in order to determine dense depth maps allowing to compute textured 3D models. Experiments showed again that a considerable systematic model error exists when computing dense depth maps perspectively and that the error can be eliminated by explicitly modeling refraction. Note that the proposed method can also be applied to other nSVP cameras like catadioptric camera systems.

The relevance of researching methods for underwater reconstruction was pointed out using examples from the fields of Archaeology and Geology.

Future Work It would be interesting to further address the question of how to correctly estimate absolute scene scale. In this thesis, it was shown that the correct scene scale can theoretically be retrieved due to the calibrated underwater housing interface, however, it fails in presence of noise in the correspondences. Secondly, the description of the method for bundle adjustment showed that the camera housing interface can be optimized if an initial solution is known. However, detailed experiments that investigate if this method eliminates the need of having to calibrate the interface using checkerboard images have not been conducted yet. This would greatly improve the method's applicability in real world scenarios, where the capture of checkerboard images in the local water body is at best impractical. The proposed refractive plane sweep algorithm has the great advantage that it is applicable not only to refractive cameras, but also to other general camera models as long as a ray with starting point and direction can be computed for each pixel. However, so far finding the best depth for each pixel in the cost volume is only implemented

as a local method. It would be interesting to extend the algorithm to global methods and try to improve the robustness of the results. Another challenge is scalability in the sense of the need to process more than a few images. For example for one black smoker shown in Chapter 6, several image sequences each of which contains several thousand images were captured. The system described and implemented for this thesis can handle image sequences of several hundred, but not yet thousands of images. In addition, those image sequences were captured using the ROV Kiel 6000, which has a set of navigation instruments. Another useful extension to the described approach is to explicitly use the navigation data and the resulting camera trajectory in a real time approach to aid robot navigation. Finally, it would be interesting to see if refractive optical and acoustic methods can be coupled in a rig, allowing to make use of the advantages each method has.

Appendix A

Appendix

A.1 Radiometric Quantities and Local Illumination

The following section gives an introduction to radiometric quantities, which are required for studying underwater light propagation. The introduction is based on [EeK96], [SSC02], [Der92], and [Mob94].

Light can be considered to be a stream photons, i. e., little quantized packages of energy where each photon has the energy Q measured in Joule. Photons can interact with matter on a molecular basis, e. g., can be absorbed or scattered. However, light can also be seen as a wave with frequency v. Let $c = 2.998\mathrm{e}^{8}\mathrm{ms}^{-1}$ be the speed of light and $h = 6.62517\mathrm{e}^{-34}\mathrm{Js}$ be Planck's constant. A a photon's energy can then be computed by:

$$Q = hv = \frac{hc}{\lambda}, \tag{A.1.1}$$

thus relating the energy to the photon's wavelength λ.

In order to model light propagation, transformations between polar and Euclidean coordinates and the solid angle measured in steradian (sr) are utilized. Let the three-dimensional vectors $(\mathsf{E}_1, \mathsf{E}_2, \mathsf{E}_3)$ define a common Euclidean coordinate system with V being a unit vector in this coordinate system. Let $\mathsf{V} = V_1\mathsf{E}_1 + V_2\mathsf{E}_2 + V_3\mathsf{E}_3$. Then, V can be equivalently rewritten in polar coordinates as follows:

$$\mathsf{V}(\theta, \varphi) = \begin{pmatrix} \cos^{-1} V_3 \\ \tan^{-1} \frac{V_2}{V_1} \end{pmatrix}. \tag{A.1.2}$$

Table A.1. Radiometric quantities.

Quantity	Symbol	Unit	Description
Radiant Energy	Q	joules $[J] = [\frac{kg}{m^2 s^2}]$	quantity of energy transferred independently of direction and time
Radiant Power (radiant flux)	$F = \frac{dQ}{dt}$	$[J/s] = [W]$	quantity of energy per second
Radiant Intensity (Light Intensity)	$I(\theta, \varphi) = \frac{dF(\theta,\varphi)}{d\Omega(\theta,\varphi)}$	$[W/sr]$	a portion of the radiant flux that is propagated through a very small solid angle around the direction $d\Omega(\theta, \varphi)$ (this system works for point sources only – now a surface emitting light is considered to be made up of infinitesimal point sources)
Radiance (Light Flux)	$L(\theta, \varphi) = \frac{dI(\theta,\varphi)}{dAcos\theta}$	$\left[\frac{W}{m^2 sr}\right]$	light intensity per surface area dA depending on the angle θ between the plane normal and the direction of the light
Irradiance	$E = \frac{dF}{dA}$	$\left[\frac{W}{m^2}\right]$	Radiant flux being incident upon a unit area of a surface (integrated over hemisphere).
Light Intensity 2	$I' = \frac{dF}{dAcos\theta}$	$\frac{W}{m^2}$	the flux F transfers a directed beam of light perpendicularly through an element of its cross-section $dAcos\theta$.

It follows:

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \end{pmatrix} = \begin{pmatrix} \sin\theta\cos\varphi \\ \sin\theta\sin\varphi \\ \cos\theta \end{pmatrix}. \tag{A.1.3}$$

The solid angle is a projection of a surface onto the unit sphere. Imagine a surface A with distance r from a sphere's center. The solid angle can then be calculated by $\Omega = \frac{A}{r^2}$. Hence, the solid angle of the whole sphere is the surface of the unit sphere, $\Omega = 4\pi$ and the solid angle of the hemispheres is $\Omega_{\text{hemisphere}} = 2\pi = \int_0^{2\pi}\int_0^{\frac{\pi}{2}} \sin\theta d\theta d\varphi = 2\pi \int_0^{\frac{\pi}{2}} \sin\theta d\theta = 2\pi[-\cos\theta]_0^{\frac{\pi}{2}}$.

Starting with the radiant energy measured in Joule J, radiometry defines the physical quantities depicted in Table A.1.

Light emission can be computed using the radiance L and integrating over the hemisphere:

$$\begin{aligned} E &= \int_{\text{hemisphere}} L(\varphi,\theta)\cos\theta d\Omega \\ &= \int_{\varphi=0}^{2\pi}\int_{\theta=0}^{\frac{\pi}{2}} L(\varphi,\theta)\cos\theta\sin\theta d\theta d\varphi. \end{aligned} \tag{A.1.4}$$

In case L is not depending on the direction, i. e., a Lambert light source is considered, this can be solved:

$$\begin{aligned} E &= -L\int_{\varphi=0}^{2\pi}\int_{\theta=0}^{\frac{\pi}{2}} \cos\theta\sin\theta d\theta d\varphi \\ &= -2\pi L\int_{\theta=0}^{\frac{\pi}{2}} \cos\theta\sin\theta d\theta \\ &= -2\pi L\left[-\frac{1}{2}\cos^2\theta\right]_0^{\frac{\pi}{2}} \\ &= -2\pi L(0-\frac{1}{2}) \\ &= L\pi \end{aligned} \tag{A.1.5}$$

From the definition of radiance follows that the emission of a Lambert light source is only depending on the angle between the surface normal and the light source direction.

Note that in Chapter 3 where, the simulator for underwater light propagation is described, point light sources are used, which can be attached to the ROV. Point light sources are assumed to emit light isotropically in all directions with a quadratic fall-off in intensity with growing distance from the light source. Thus, the irradiance, which is incident upon a surface point and comes from a point light source, is $E = \frac{I}{r^2}\cos\theta$ with r being the distance and θ being the angle between surface normal and the direction of the incoming light.

In order to model reflectance [Sze11], the Bidirectional Reflectance Distribution Function (BRDF) describes how light is reflected by surfaces in the scene. It depends on incoming θ_i and ϕ_i and reflected angles θ_r, ϕ_r,

but also on the light's wavelength λ:

$$f_r(\theta_i, \phi_i, \theta_r, \phi_r; \lambda). \tag{A.1.6}$$

On isotropic surfaces, this can be simplified to:

$$f_r(v_i, v_r, \tilde{n}; \lambda), \tag{A.1.7}$$

where v_i and v_r are incoming and reflected rays and $\tilde{n}$ is the normal of the surface. Reflectance is thus modeled by the incoming irradiance and the reflectance function f_r:

$$L(v_r, \lambda) = f_r(v_i, v_r, \tilde{n}; \lambda) E(v_i, \lambda). \tag{A.1.8}$$

This describes the emitted radiance of a surface due to reflection. However, to determine the overall radiance emitted by a surface, integration over the contribution of all light sources is needed:

$$L_r(v_r, \lambda) = \int L_i(v_i, \lambda) f_r(v_i, v_r, \tilde{n}; \lambda) \cos^+ \theta_i \, dv_i, \tag{A.1.9}$$

with $\cos^+ \theta_i = \max(\cos \theta_i, 0)$. In case of a finite number of light sources, the integral is replaced by a sum.

A.1.1 Phong Model

The concepts explained above are utilized in the area of computer graphics in order to compute artificial illumination of scenes. A common model for this is the Phong model (introduced in 1975), where reflection at an object surface is modeled in three components, diffuse, specular, and ambient parts of light, hence, the BRDF is split into three different functions [Sze11]:

diffuse reflection from surfaces is modeled by a constant function, depending only on wavelength: $f_d(v_i, v_r, \tilde{n}; \lambda) = f_d(\lambda)$. Here, the light is reflected into all directions.

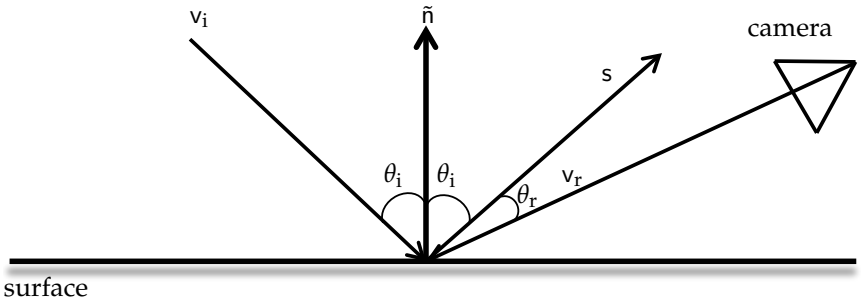

Figure A.1. Phong model specular part.

specular reflection from shiny surfaces is modeled by:

$$f_s(v_i, v_r, \tilde{n}; \lambda) = k_s(\lambda) \cos^k \theta_r \tag{A.1.10}$$

In this case the light is mainly reflected at the same angle under which it hit the surface. The light reaching the camera is therefore depending on the angle between the reflected light and the camera ray (Figure A.1).

ambient light is the third component of the Phong model. It describes light that has been scattered, reflected, and refracted so often, that it is perceived as surrounding light, which cannot be assigned to any particular light source:

$$f_a = k_a L_a(\lambda). \tag{A.1.11}$$

Combining all three components for a finite number of light sources yields:

$$L_r(v_r, \lambda) = k_a L_a(\lambda) + k_d(\lambda) \sum_i L_i(\lambda)(v_i^T \tilde{n})^+ + k_s(\lambda) \sum_i L_i(\lambda)(v_r^T s)^k. \tag{A.1.12}$$

In Chapter 3, the simulator uses only the diffuse part of the model. This is due to the assumption that the underwater scenes are assumed to not have a lot of materials that have strong specular reflections. The ambient light is replaced by explicitly modeling backscatter, which allows to compute a specialized ambient light that is not completely uniform but stronger closer to the light sources.

A.2 Camera Optics

In general, it can be said that images capture radiance (Section A.1) coming from the scene in front of the camera. In order to model that effect, different illumination models exist, e. g., the Phong model (Section A.1.1) [Sze11]. The entrance pupil model described in Section 2.2.2 already showed that several light rays are actually captured by one pixel and that if the scene is not exactly in focus, blobs are imaged instead of sharp images. This depends on the distance and explains the depth-of-field effect. To make things worse, the use of lenses causes the light to be refracted, which is a wavelength-dependent effect called chromatic aberration, i. e., different colors are focused at slightly different distances. Compound lenses with different materials and therefore different refractive indices can help to minimize this effect in modern cameras [Sze11]. Another effect causes the image brightness to fall off with increasing distance from the center of the image and is called vignetting. It can have multiple causes, among them the lens system and the aperture. The fundamental radiometric relation describes how the radiance L coming from the scene is related to the irradiance E (refer to Section A.1), which is incident upon the sensor [Sze11]:

$$E = L\frac{\pi}{4}\left(\frac{d}{f}\right)^2 cos^4\theta \quad [\mathrm{Wm}^{-2}], \tag{A.2.1}$$

with d being the aperture diameter, f being the focal length, and θ being the angle between the main incoming ray and the optical axis.

A.3 Singular Value Decomposition

The Singular Value Decomposition (SVD) is a useful method of decomposing matrices, which has a variety of applications especially in geometry estimation ([PVT+02], [Sch05], [BS99], and [HZ04]). It can be used to determine the null-space of a matrix, to solve homogeneous and inhomogeneous systems of linear equations, to enforce rank constraints on matrices, and to compute pseudo-inverses of matrices.

Definition Let $\mathbf{A}$ be an $m \times n$ matrix with $m \geqslant n$. Then, the Singular Value Decomposition of $\mathbf{A}$ is:

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}} \tag{A.3.1}$$

with $\mathbf{S}$ being an $n \times n$ diagonal matrix with singular values ordered by size (smallest one in the last row), $\mathbf{U}$ being an $m \times n$ matrix with orthonormal columns ($\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}_{n\times n}$, but $\mathbf{U}\mathbf{U}^{\mathrm{T}} \neq \mathbf{I}_{m\times m}$), and $\mathbf{V}$ being an $n \times n$ orthonormal matrix. There is a connection between singular values and eigenvalues for the $m \times n$ matrix $\mathbf{A}$:

$$\mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{V}\mathbf{S}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}} = \mathbf{V}\mathbf{S}^{2}\mathbf{V}^{\mathrm{T}} \tag{A.3.2}$$

$$\mathbf{A}\mathbf{A}^{\mathrm{T}} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}\mathbf{V}\mathbf{S}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}} = \mathbf{U}\begin{bmatrix} \mathbf{S}^{2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\mathbf{U}^{\mathrm{T}}, \tag{A.3.3}$$

Thus, the singular values are the square roots of the eigenvalues of the matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$. In order to determine the null- or image-space, let $\mathbf{A}$ be an $m \times n$ matrix or linear function for vector $x \in \mathbb{R}^{n}$:

$$\mathbf{A}x = y, \quad y \in \mathbb{R}^{m}. \tag{A.3.4}$$

Then, the null- and image spaces of $\mathbf{A}$ can be found by:

- for all singular values = 0, the corresponding columns in $\mathbf{V}^{\mathrm{T}}$ span the null space of $\mathbf{A}$ and
- for all singular values $\neq 0$, the corresponding columns in $\mathbf{U}$ span the image space of $\mathbf{A}$.

In addition, the SVD can be used to enforce rank constraints on matrices. Let $\mathbf{A}$ be a 3×3 matrix that has been determined for example by the eight-point algorithm for computing the Fundamental matrix from noisy 2D correspondences. The rank-two constraints can be enforced by the following steps:

- compute SVD: $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}$,
- set smallest singular value in $\mathbf{S}$ to 0 and get $\mathbf{S}'$, and

▷ compute $\mathbf{A}' = \mathbf{U}\mathbf{S}'\mathbf{V}^{\mathrm{T}}$.

Another application of the SVD is solving linear systems of equations. For example, let $\mathbf{A}$ be an $m \times n$ matrix containing measurements that have recorded data in each row for n variables. Then, the following system can be solved by computing the SVD of $\mathbf{A}$:

$$\mathbf{A}\mathsf{x} = 0. \tag{A.3.5}$$

The singular values are then an indicator to the number of possible solutions:

▷ if no singular value equals 0, no exact solution exists (see Linear-Least-Squares problems),

▷ if one singular value equals 0, the corresponding column in $\mathbf{V}$ yields the up-to-scale solution, and

▷ if more than one singular value equals 0, a multi-dimensional space of solutions exist. It is spanned by the corresponding column vectors in $\mathbf{V}$.

In case of the linear system of equations being inhomogeneous, the pseudo inverse can be computed using the SVD, by inverting the $m \times n$ matrix $\mathbf{A}$:

$$\mathbf{A}^{-1} = (\mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}})^{-1} = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^{\mathrm{T}}. \tag{A.3.6}$$

In case $\mathbf{A}$ is diagonal, inverting the matrix is trivial if all singular values > 0. If not, all non-zero elements are inverted, the others remain zero. The pseudo inverse can then be used to solve inhomogeneous systems of linear equations of the form:

$$\mathbf{A}\mathsf{x} = \mathsf{b}, \tag{A.3.7}$$

with $\mathsf{b} \neq 0$. In practical problems, the data is often noisy, e. g., measurements. Then, the linear system of equations does not have an exact solution and the SVD can be used to find the best solution in a Linear-Least-Squares sense. Find:

$$\mathsf{x}, \text{ such that } r \cong |\mathbf{A}\mathsf{x} - \mathsf{b}|^2 \text{ is minimized.} \tag{A.3.8}$$

r is the residuum of the solution, which equals the smallest singular value in case of homogeneous systems of equations.

A.4 Parameter Estimation using Numerical Optimization

In this section, two different approaches to parameter optimization are introduced. The objective in both cases is to use a set of observations and find a set of parameters describing a model that fits the observations as closely as possible. The classic case for this is a set of cameras each parametrized by an individual camera pose, but sharing intrinsic parameters common for all cameras. In addition, a set of 3D points in space exists that has been reconstructed using the cameras. 3D points, camera poses, and intrinsics are the parameters to be optimized. The model is the perspective projection, thus the detected feature points in the images are the observations. Classical bundle adjustment is the process of optimizing the scene description such that the average (squared) reprojection error in all images is minimized. This is achieved by applying a method for non-linear optimization, usually by locally minimizing a linearized version of the error function. First, possibilities for such an optimization are described. Note that in case of camera calibration with known 3D points (checkerboard) this can be applied as well. Then, a global method that does not require the computation of derivatives is introduced.

Notations and assumptions:

- parameter vector $\mathsf{p} \in \mathbb{R}^n$ with initialization p^0
- observation vector $\mathsf{l} \in \mathbb{R}^m$, $\mathbf{C}_{ll} \in \mathbb{R}^{m \times m}$ observation covariance matrix
- in general $m > n$ is assumed
- model can be described by:

1. Explicit model constraints of the form:

$$\mathsf{f} : \mathbb{R}^n \to \mathbb{R}^m, \mathsf{f}(\mathsf{p}) = \mathsf{l} \qquad \text{with} \quad \mathbf{A}_\mathsf{f} = \frac{\partial \mathsf{f}}{\partial \mathsf{p}} \tag{A.4.1}$$

2. Implicit model constraints e. g., refractive checkerboard error:

$$\mathrm{g} : \mathbb{R}^n \to \mathbb{R}^k, \mathrm{g}(\mathsf{p}, \mathsf{l}) = 0 \quad \text{with} \quad \mathbf{A}_\mathrm{g} = \frac{\partial \mathrm{g}}{\partial \mathsf{p}} \in \mathbb{R}^{k \times n} \quad \mathbf{B}_\mathrm{g} = \frac{\partial \mathrm{g}}{\partial \mathsf{l}} \in \mathbb{R}^{k \times m} \tag{A.4.2}$$

3. Parameter constraints e. g., unit length of normal or quaternions:

$$\mathrm{h} : \mathbb{R}^n \to \mathbb{R}^l, \mathrm{h}(\mathsf{p}) = 0 \ \text{ with } \ \mathbf{H}_\mathrm{h} = \frac{\partial \mathrm{h}}{\partial \mathsf{p}} \in \mathbb{R}^{l \times n} \tag{A.4.3}$$

▻ Functions f, g, h are not necessarily linear.

When parametrizing rotations and 3D points, one has to take care not to over-parametrize, which yields correlations between parameters, but also to avoid discontinuities in the parameter space because the approach is to use an iterative approach on a linearized version of the model. Triggs et al. [TMHF00] have some suggestions as to how parametrize rotations and 3D points and how to deal with outliers:

rotations use quaternions with extra h-constraint for unit length or incremental Euler angles $\mathbf{R}\delta\mathbf{R} = \mathbf{R}(\mathbf{I} + [\delta r]_x)$. In this thesis, quaternions are used for parametrizing rotation during SfM and incremental Euler angles in the described calibration approach.

3D points if 3D points can be close to infinity (far away points, outliers, etc.) they can come close to or even move across the plane of infinity during optimization. In this case it is better to parametrize them as homogeneous points instead of euclidean vectors. Due to limited visibility underwater, 3D points are not assumed to be close to infinity and are hence parametrized as euclidean vectors.

outliers in the input data are not a problem as long as they are reflected as uncertain in the weight matrix and as long as the input data weight matrix is used in an ML estimation. Better yet is to derive an optimization scheme that makes explicit use of a robust error function.

In general, it can be summarized that the parameters space must not have singularities in the relevant area, it needs to be continuous, and differentiable [HZ04].

Optimizing a system as described above can be achieved using several different approaches, two of which will now be introduced starting with least squares using derivatives of the error functions, followed by a derivative-free, global, evolutionary method.

A.4.1 Least Squares

This section is based on [TMHF00], [McG04], and [HZ04].

Linear Least Squares

In case f is linear [HZ04], a matrix $\mathbf{A}_f$ exists such that $\mathbf{A}_f p = l + v$, with $v = \mathbf{A}_f p - l$ being the residual that needs to be minimized. $\mathbf{A}_f$ is an $n \times m$ matrix and is assumed to have full column rank i. e., $\text{rank}(\mathbf{A}_f) = n$. In order to optimize the problem using the least squares approach, the following function is minimized:

$$\text{argmin}_p \Phi = v^T v = (\mathbf{A}_f p - l)^T (\mathbf{A}_f p - l) = (\mathbf{A}_f p)^T (\mathbf{A}_f p) - 2(\mathbf{A}_f p)^T l + l^T l. \tag{A.4.4}$$

Setting the derivative of Φ to zero yields the solution:

$$\frac{\partial \Phi}{\partial p} = 2\mathbf{A}_f^T \mathbf{A}_f p - 2\mathbf{A}_f^T l = 0 \tag{A.4.5}$$

$$\Rightarrow \mathbf{A}_f^T \mathbf{A}_f p = \mathbf{A}_f^T l \tag{A.4.6}$$

$$\Rightarrow p = (\mathbf{A}_f^T \mathbf{A}_f)^{-1} \mathbf{A}_f^T l, \tag{A.4.7}$$

where either the normal equation system (A.4.6) can be solved or the SVD can be used to compute the pseudo inverse in (A.4.7).

Linear Least Squares with Consideration of Observation Weights

In case an observation covariance matrix exists (usually diagonal or block diagonal) [TMHF00], the observations can be weighted according to their uncertainty. The function to be minimized from (A.4.4) is changed to:

$$\text{argmin}_p \Phi = v^T \mathbf{C}_{ll}^{-1} v = (\mathbf{A}_f p - l)^T \mathbf{C}_{ll}^{-1} (\mathbf{A}_f p - l) \tag{A.4.8}$$

$$= (\mathbf{A}_\mathrm{f}\mathsf{p})^\mathrm{T}(\mathbf{C}_{ll}^{-1}\mathbf{A}_\mathrm{f}\mathsf{p}) - 2(\mathbf{A}_\mathrm{f}\mathsf{p})^\mathrm{T}\mathbf{C}_{ll}^{-1}\mathsf{l} + \mathsf{l}^\mathrm{T}\mathbf{C}_{ll}^{-1}\mathsf{l}.$$

Following the derivation as in the linear least squares case above, this leads to:

$$\begin{aligned} \mathbf{A}_\mathrm{f}^\mathrm{T}\mathbf{C}_{ll}^{-1}\mathbf{A}_\mathrm{f}\mathsf{p} &= \mathbf{A}_\mathrm{f}^\mathrm{T}\mathbf{C}_{ll}^{-1}\mathsf{l} \\ \mathsf{p} &= (\mathbf{A}_\mathrm{f}^\mathrm{T}\mathbf{C}_{ll}^{-1}\mathbf{A}_\mathrm{f})^{-1}\mathbf{A}_\mathrm{f}^\mathrm{T}\mathbf{C}_{ll}^{-1}\mathsf{l}. \end{aligned} \tag{A.4.9}$$

According to the Gauss-Markov theorem, if p and l are random variables, $rank(\mathbf{A}_\mathrm{f}) = n$ and the error vector $\mathsf{v} = \mathbf{A}_\mathrm{f}\mathsf{p} - \mathsf{l}$ has zero mean and zero correlations, then $\mathsf{p} = (\mathbf{A}_\mathrm{f}^\mathrm{T}\mathbf{C}_{ll}^{-1}\mathbf{A}_\mathrm{f})^{-1}\mathbf{A}^\mathrm{T}\mathbf{C}_{ll}^{-1}\mathsf{l}$ is the best linear unbiased estimator (BLUE) for the parameter vector and the covariance matrix of the parameters is given by $\mathrm{cov}(\mathsf{p}) = \sigma^2(\mathbf{A}_\mathrm{f}^\mathrm{T}\mathbf{C}_{ll}^{-1}\mathbf{A}_\mathrm{f})^{-1}$ with $\sigma^2 = \frac{1}{m-n}(\mathsf{l} - \mathbf{A}_\mathrm{f}\mathsf{p})^\mathrm{T}\mathbf{C}_{ll}^{-1}(\mathsf{l} - \mathbf{A}_\mathrm{f}\mathsf{p})$. $R = m - n$ is called redundancy, σ^2 is the variance factor.

In addition, it is possible to derive the covariance matrix of the residuals:

$$\mathbf{C}_{vv} = \sigma^2(\mathbf{C}_{ll} - \mathbf{A}_\mathrm{f}\mathrm{cov}(\mathsf{p})\mathbf{A}_\mathrm{f}^\mathrm{T}), \tag{A.4.10}$$

allowing to evaluate the result (compare to [McG04]).

Linear Least Squares with Constraints Between Parameters

In case linear constraints between parameters of the form $\mathrm{h}(\mathsf{p}) = \mathbf{H}_\mathrm{h}\mathsf{p} + \mathsf{b} = 0$ exist, the linear least squares approach described above needs to be extended using Lagrange Multipliers $\mathsf{k} \in \mathbb{R}^n$ [TMHF00]:

$$\mathrm{argmin}_\mathsf{p}\Phi = \mathsf{v}^\mathrm{T}\mathsf{v} + 2 \underbrace{\mathsf{k}^\mathrm{T}(\mathbf{H}_\mathrm{h}\mathsf{p} + \mathsf{b})}_{\text{commutative scalar product}}. \tag{A.4.11}$$

In this function, the vector p and the Lagrange Multipliers k^T are unknown:

$$\begin{aligned} \frac{\partial\Phi}{\partial\mathsf{p}} &= 2\mathbf{A}_\mathrm{f}^\mathrm{T}\mathbf{A}_\mathrm{f}\mathsf{p} - 2\mathbf{A}_\mathrm{f}^\mathrm{T}\mathsf{l} + 2\mathbf{H}_\mathrm{h}^\mathrm{T}\mathsf{k} = 0 \\ \frac{\partial\Phi}{\partial\mathsf{k}^\mathrm{T}} &= \mathbf{H}_\mathrm{h}\mathsf{p} + \mathsf{b} = 0, \end{aligned} \tag{A.4.12}$$

which yields the linear system of equations:

$$\begin{bmatrix} \mathbf{A}_f^T \mathbf{A}_f & \mathbf{H}_h^T \\ \mathbf{H}_h & \mathbf{0} \end{bmatrix} \begin{bmatrix} p \\ k \end{bmatrix} = \begin{bmatrix} \mathbf{A}_f^T l \\ -b \end{bmatrix}. \tag{A.4.13}$$

Solving the system of equations yields the parameter vector p while at the same time full-filling the constraints h.

Gradient Descent for Non-Linear Functions

In case of non-linear functions, the solution is found iteratively, starting with an initialization of the parameter vector. $\operatorname{argmin}_p \Phi = v^T v = (f(p) - l)^T (f(p) - l)$ is the sum of least squares functions that needs to be minimized iteratively with respect to p. When optimizing parameters p from a starting point p^0 it is possible to determine the gradient of the function in p^0 and using it to determine the direction of steepest descent along the error function and use this as an update for the parameters [HZ04]:

$$\begin{aligned} \lambda \Delta p &= -\frac{\partial \Phi}{\partial p^\nu} \quad \lambda \in \mathbb{R} \\ p^{\nu+1} &= p^\nu + \lambda \Delta p, \end{aligned} \tag{A.4.14}$$

λ is a damping factor determining the step size to be taken in the current iteration. It can for example be determined using line search. i. e., starting with $\lambda = 1$, it is determined if the parameter update leads to a decrease in the error function. If not, half the step size is tried. Note that gradient descent sometimes converges only slowly. An often better choice are the Newton-Iterations, which can be derived using the same scheme as described in the linear least squares scheme above.

Non-Linear Least Squares (Gauss-Newton Iterations and Levenberg-Marquardt Algorithm)

As opposed to the assumption in the linear least squares case, the function f is now non-linear, resulting in a similar computation only this time an initial solution p^0 needs to be known and is then updated iteratively in

order to find the best fitting parameter vector p. This is achieved by using a Taylor approximation of the error function:

$$\begin{aligned}
&\mathsf{f}(\mathsf{p}+\Delta\mathsf{p}) \approx \mathsf{f}(\mathsf{p}) + \mathbf{A}_\mathsf{f}\Delta\mathsf{p} \qquad (A.4.15)\\
&\mathsf{v} = \mathsf{f}(\mathsf{p}) + \mathbf{A}_\mathsf{f}\Delta\mathsf{p} - \mathsf{I}\\
&\Rightarrow \operatorname{argmin}_\mathsf{p}\Phi = \mathsf{v}^\mathsf{T}\mathsf{v}\\
&= (\mathsf{f}(\mathsf{p}) + \mathbf{A}_\mathsf{f}\Delta\mathsf{p} - \mathsf{I})^\mathsf{T}(\mathsf{f}(\mathsf{p}) + \mathbf{A}_\mathsf{f}\Delta\mathsf{p} - \mathsf{I})\\
&= \mathsf{f}(\mathsf{p})^\mathsf{T}\mathsf{f}(\mathsf{p}) + 2(\mathbf{A}_\mathsf{f}\Delta\mathsf{p})^\mathsf{T}\mathsf{f}(\mathsf{p}) + (\mathbf{A}_\mathsf{f}\Delta\mathsf{p})^\mathsf{T}(\mathbf{A}_\mathsf{f}\Delta\mathsf{p}) - 2\mathsf{f}(\mathsf{p})^\mathsf{T}\mathsf{I} - 2(\mathbf{A}_\mathsf{f}\Delta\mathsf{p})^\mathsf{T}\mathsf{I} + \mathsf{I}^\mathsf{T}\mathsf{I},
\end{aligned}$$

which is again derived in the direction of $\Delta\mathsf{p}$ and set to zero:

$$\frac{\partial\Phi}{\partial\Delta\mathsf{p}} = 2\mathbf{A}_\mathsf{f}^\mathsf{T}\mathsf{f}(\mathsf{p}) + 2\mathbf{A}_\mathsf{f}^\mathsf{T}\mathbf{A}_\mathsf{f}\Delta\mathsf{p} - 2\mathbf{A}_\mathsf{f}^\mathsf{T}\mathsf{I} = 0, \qquad (A.4.16)$$

and leads to the following iterative updating scheme:

$$\begin{aligned}
\mathbf{A}_\mathsf{f}^\mathsf{T}\mathbf{A}_\mathsf{f}\Delta\mathsf{p} &= -\mathbf{A}_\mathsf{f}^\mathsf{T}(\mathsf{f}(\mathsf{p}^\nu) - \mathsf{I}) \qquad (A.4.17)\\
\mathsf{p}^{\nu+1} &= \mathsf{p}^\nu + \Delta\mathsf{p}.
\end{aligned}$$

Note that at this point $\mathbf{A}_\mathsf{f}^\mathsf{T}\mathbf{A}_\mathsf{f}$ is the same approximation of the Hessian as used in Gauss-Newton iterations (compare to [HZ04]), the advantage of which is that the second derivatives do not have to be computed explicitly and the approximation is usually positive definite.

In order to stabilize convergence, the Levenberg-Marquardt algorithm combines gradient descent and Gauss-Newton iterations. This is achieved by using the augmented normal equations [HZ04]:

$$\left(\mathbf{A}_\mathsf{f}^\mathsf{T}\mathbf{A}_\mathsf{f} + \lambda\mathbf{I}\right)\Delta\mathsf{p} = -\mathbf{A}_\mathsf{f}^\mathsf{T}\mathsf{f}(\mathsf{p}^\nu), \qquad (A.4.18)$$

with $\lambda = 0.001 \in \mathbb{R}$. In case the error function increases, λ is multiplied by a fixed factor, if it decreases, λ is divided by the same factor [HZ04]. Thus, for large λ, the iterations are essentially a gradient descent, while for small λ, Gauss-Newton iterations are computed. According to [ESN06] the identity matrix $\mathbf{I}$ can also be substituted by $\mathbf{N} = \operatorname{diag}(\mathbf{A}_\mathsf{f}^\mathsf{T}\mathbf{A}_\mathsf{f})$, yielding a set of normal equations with even better convergence.

Non-Linear Least Squares with Parameter Constraints and Observation Weights

Combining all cases derived so far yields an optimization scheme that can deal with non-linear, explicit error functions, weights for observations, and additional (non-linear) constraints between parameters. First, the error function and parameter constraints need to be linearized:

$$\begin{aligned} &f(p + \Delta p) \approx f(p) + \mathbf{A}_f \Delta p \\ &h(p + \Delta p) \approx h(p) + \mathbf{H}_h \Delta p \\ &v = f(p) + \mathbf{A}_f \Delta p - l \\ &\Rightarrow \operatorname{argmin}_p \Phi = v^T \mathbf{C}_{ll}^{-1} v = 2k^T (h(p) + \mathbf{H}_h \Delta p), \end{aligned} \tag{A.4.19}$$

where k is a set of Lagrange-Multipliers. Computing derivatives of Φ in direction of Δp and k^T and setting them to zero yields:

$$\begin{aligned} \frac{\partial \Phi}{\partial \Delta p} &= 2\mathbf{A}_f^T \mathbf{C}_{ll}^{-1} f(p) + 2\mathbf{A}_f^T \mathbf{C}_{ll}^{-1} \mathbf{A}_f \Delta p - 2\mathbf{A}_f^T \mathbf{C}_{ll}^{-1} l + 2\mathbf{H}_h^T k = 0 \\ \frac{\partial \Phi}{\partial k^T} &= 2(h(p) + \mathbf{H}_h \Delta p) = 0, \end{aligned} \tag{A.4.20}$$

which directly yields the following linear system of equations:

$$\begin{bmatrix} \mathbf{A}_f^T \mathbf{C}_{ll}^{-1} \mathbf{A}_f & \mathbf{H}_h^T \\ \mathbf{H}_h & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta p \\ k \end{bmatrix} = \begin{bmatrix} -\mathbf{A}_f^T \mathbf{C}_{ll}^{-1} (f(p) - l) \\ -h(p) \end{bmatrix}. \tag{A.4.21}$$

Non-Linear Least Squares with Implicit Constraints

Up until now, only explicit constraints of the form $f(p) = l$ were considered. Sometimes, e. g., when calibrating underwater housings, formulating explicit constraints is not possible. Therefore, non-linear, implicit constraints of the form $g(p, l) = 0$ are now considered:

$$\begin{aligned} &g(p + \Delta p, l + v) \approx g(p, l) + \mathbf{A}_g \Delta p + \mathbf{B}_g v = 0 \\ &v = -\mathbf{B}_g^{-1} g(p, l) - \mathbf{B}_g^{-1} \mathbf{A}_g \Delta p \\ &\Rightarrow \operatorname{argmin}_p \Phi = v^T v. \end{aligned} \tag{A.4.22}$$

As before, the derivative in direction of Δp is computed and set to zero:

$$\begin{aligned}
&\frac{\partial \Phi}{\partial \Delta p} = 2(\mathbf{B}_g^{-1}\mathbf{A}_g)^T(\mathbf{B}_g^{-1}g(p,l)) + 2(\mathbf{B}_g^{-1}\mathbf{A}_g)^T(\mathbf{B}_g^{-1}\mathbf{A}_g)\Delta p = 0 \quad \text{(A.4.23)}\\
&\Rightarrow \mathbf{A}_g^T(\mathbf{B}_g\mathbf{B}_g^T)^{-1}\mathbf{A}_g g(p,l) + \mathbf{A}_g^T(\mathbf{B}_g\mathbf{B}_g^T)^{-1}\mathbf{A}_g\Delta p = 0\\
&\Rightarrow \mathbf{A}_g^T(\mathbf{B}_g\mathbf{B}_g^T)^{-1}\mathbf{A}_g\Delta p = -\mathbf{A}_g^T(\mathbf{B}_g\mathbf{B}_g^T)^{-1}\mathbf{A}_g g(p,l).
\end{aligned}$$

Full Iterative Scheme with Implicit Error Function, Observation Weights, and Parameter Constraints

In this section, all of the above options are combined in order to describe the most general model with implicit error function, observation uncertainty, and constraints between parameters. In [McG04], this corresponds to the Gauss-Helmert model for optimization. This time, the error function is formulated using Lagrange Multipliers for the linearized versions of g and h.

$$\begin{aligned}
&g(p+\Delta p, l+v) \approx g(p,l) + \mathbf{A}_g\Delta p + \mathbf{B}_g v = 0 \quad \text{(A.4.24)}\\
&h(p+\Delta p) \approx h(p) + \mathbf{H}_h\Delta p = 0\\
&\Rightarrow \operatorname{argmin}_p \Phi = v^T\mathbf{C}_{ll}^{-1}v + 2k_g^T(g(p,l) + \mathbf{A}_g\Delta p + \mathbf{B}_g v) + 2k_h^T(h(p) + \mathbf{H}_h\Delta p),
\end{aligned}$$

where k_g are the Lagrange Multipliers for g and k_h are the Lagrange Multipliers for h. In order to minimize Φ, derivatives in the directions v, Δp, k_g^T, and k_h^T are computed and set to zero:

$$\begin{aligned}
&\frac{1}{2}\frac{\partial \Phi}{\partial v} = v^T\mathbf{C}_{ll}^{-1} + k_g^T\mathbf{B}_g = 0 \quad \Rightarrow \quad v = -\mathbf{C}_{ll}\mathbf{B}_g^T k_g && \text{(A.4.25)}\\
&\frac{1}{2}\frac{\partial \Phi}{\partial \Delta p} = k_g^T\mathbf{A}_g + k_h^T\mathbf{H}_h = 0 && \text{(A.4.26)}\\
&\frac{1}{2}\frac{\partial \Phi}{\partial k_g^T} = (g(p,h) + \mathbf{A}_g\Delta p + \mathbf{B}_g v) = 0 && \text{(A.4.27)}\\
&\Rightarrow \quad k_g = (\mathbf{B}_g\mathbf{C}_{ll}\mathbf{B}_g^T)^{-1}(g(p,l) + \mathbf{A}_g\Delta p) && \text{(A.4.28)}\\
&\frac{1}{2}\frac{\partial \Phi}{\partial k_h^T} = h(p) + \mathbf{H}_h\Delta p = 0. && \text{(A.4.29)}
\end{aligned}$$

Algorithm A.1 Gauss-Helmert Model

Initialization l, p^0, $\mathbf{C}_{ll}$, step size $\lambda \in \mathbb{R}$
while not converged **do**
$\mathbf{A}_\mathrm{g} = \frac{\partial \mathrm{g}}{\partial \mathsf{p}^i}$
$\mathbf{B}_\mathrm{g} = \frac{\partial \mathrm{g}}{\partial \mathsf{l}^i}$
$\mathbf{H}_\mathrm{h} = \frac{\partial \mathrm{h}}{\partial \mathsf{p}^i}$
solve (A.4.30) for $\Delta\mathsf{p}$ and invert $\mathbf{N} = \mathbf{A}_\mathrm{g}^\mathrm{T}(\mathbf{B}_\mathrm{g}\mathbf{C}_{ll}\mathbf{B}_\mathrm{g}^\mathrm{T})^{-1}\mathbf{A}_\mathrm{g}$
$\mathsf{v} = -\mathbf{C}_{ll}\mathbf{B}_\mathrm{g}^\mathrm{T}(\mathbf{B}_\mathrm{g}\mathbf{C}_{ll}\mathbf{B}_\mathrm{g}^\mathrm{T})^{-1}(\mathrm{g}(\mathsf{p},\mathsf{l}) - \mathbf{A}_\mathrm{g}\Delta\mathsf{p})$ (A.4.25, A.4.28)
$\Omega = \mathsf{v}^\mathrm{T}\mathbf{C}_{ll}^{-1}\mathsf{v} = (\mathrm{g}(\mathsf{p},\mathsf{l}) - \mathbf{A}_\mathrm{g}\Delta\mathsf{p})^\mathrm{T}(\mathbf{B}_\mathrm{g}\mathbf{C}_{ll}\mathbf{B}_\mathrm{g}^\mathrm{T})^{-1}(\mathrm{g}(\mathsf{p},\mathsf{l}) - \mathbf{A}_\mathrm{g}\Delta\mathsf{p})$
$R = m + \mathrm{size}(\mathrm{h}) - n$
$\sigma_0 = \frac{\Omega}{R}$
$\mathsf{r} = \lambda\mathsf{r}$
$\Delta\mathsf{p} = \lambda\Delta\mathsf{p}$
$\mathsf{p}^{i+1} = \mathsf{p}^i + \Delta\mathsf{p}$
convergence if $\frac{\Delta\mathsf{p}_j}{\mathbf{N}_{jj}^{-1}} < \mathit{thres} \quad \forall j < n$
end while

After substituting k_g in Equation (A.4.26), (A.4.26) and (A.4.29) form the linear system of equations to be solved in each iteration:

$$\begin{bmatrix} \mathbf{A}_\mathrm{g}^\mathrm{T}(\mathbf{B}_\mathrm{g}\mathbf{C}_{ll}\mathbf{B}_\mathrm{g}^\mathrm{T})^{-1}\mathbf{A}_\mathrm{g} & \mathbf{H}_\mathrm{h}^\mathrm{T} \\ \mathbf{H}_\mathrm{h} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathsf{p} \\ \mathsf{k}_\mathrm{h} \end{bmatrix} = \begin{bmatrix} -\mathbf{A}_\mathrm{g}^\mathrm{T}(\mathbf{B}_\mathrm{g}\mathbf{C}_{ll}\mathbf{B}_\mathrm{g}^\mathrm{T})^{-1}\mathrm{g}(\mathsf{p},\mathsf{l}) \\ -\mathrm{h}(\mathsf{p}) \end{bmatrix} \quad \text{(A.4.30)}$$

Equation (A.4.30) leads to the iterative scheme for estimating the parameters p described in Algorithm A.1. Note that the step size λ and parameter update can also be computed as described for the Levenberg-Marquardt algorithm above or using a line search routine.

Implementation Strategies for Non-Linear-Least Squares

When implementing the above described system for non-linear optimization, usually huge matrices need to be expected because in case of bundle adjustment, there can easily be thousands or even hundreds of thousands

of parameters and even more observations. Therefore, running such a system is time and memory consuming. However, the large matrices are usually sparse and of block structure and in the literature, several methods have been found to deal with those challenges, e. g.,:

- ▻ Inversion of matrix with constraints (see [TMHF00] 4.4)
- ▻ Schur complement (see [TMHF00] 6.1)
- ▻ Using sparseness/block structure of Hessian for example in top-down methods (see [TMHF00] 6.3.2)
- ▻ Multi-Core strategies (see [WACS11])

Compressed Column Form for Sparse Matrices Bundle adjustment involves the handling of very large, but also very sparse matrices, i. e., only few entries are non-zero. Such sparse matrices can be handled efficiently [Dav06]. In order to do that, the matrices are saved in the compressed column form, which allows rapid access of columns, but is time-consuming when accessing rows. SuiteSparse[1] is a software library that was used in the implementation of bundle adjustment in this thesis. It offers basic matrix operations as well as the data structure itself.

Schur Complement The sparseness and special block structure of matrices in bundle adjustment problems can be utilized for fast inversion of $\mathbf{N}$ and/or solving the normal equation system. For this purpose, the block structure of matrix $\mathbf{N}$ is depicted in example cases with perspective and refractive, monocular and stereo cameras and 3D points in Figure 5.19 and Figure 5.20. The upper left block is diagonal and contains the entries concerning the 3D points. The not-sparse upper middle block contains entries concerning the cameras, the far right block contains entries arising from parameter constraints. The overwhelming part of the matrix is thus comprised of the 3D point block matrix. This upper left part can be inverted very efficiently. The Schur complement method (see also [TMHF00]) allows to decompose the matrix into blocks and invert them separately or reduce the size of the linear system of equations to be solved.

[1] http://www.cise.ufl.edu/research/sparse/SuiteSparse/

Note that matrix $\mathbf{N}$ is symmetric. This fact will be utilized here, although it is not necessary for the Schur complement method. Let:

$$\mathbf{N} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^{\mathrm{T}} & \mathbf{D} \end{bmatrix} \quad \bar{\mathbf{D}} = \mathbf{D} - \mathbf{B}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{B}, \tag{A.4.31}$$

then $\mathbf{N}$ can be inverted by:

$$\mathbf{N}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}\bar{\mathbf{D}}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}\bar{\mathbf{D}}^{-1} \\ -\bar{\mathbf{D}}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{A}^{-1} & \bar{\mathbf{D}}^{-1} \end{bmatrix}, \tag{A.4.32}$$

with symmetry of $\bar{\mathbf{D}}$ and $\mathbf{A}$, it follows:

$$\mathbf{N}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}\bar{\mathbf{D}}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}\bar{\mathbf{D}}^{-1} \\ -(\mathbf{A}^{-1}\mathbf{B}\bar{\mathbf{D}}^{-1})^{\mathrm{T}} & \bar{\mathbf{D}}^{-1} \end{bmatrix}. \tag{A.4.33}$$

The last pages gave an introduction into optimization using the Gauss-Helmert model and pointed out some implementation issues. In addition to implementing the system itself, the derivatives of the error functions need to be computed as well. Ideally, analytic derivatives are used, but for example finding the analytic derivative of the virtual camera error can be involved. Therefore, the analytic derivatives for this thesis were computed using Maxima[2], a software that can manipulate and especially differentiate symbolic expressions.

A.4.2 Global Evolutionary Optimization (CMA-ES)

The method for optimization in case of classic bundle adjustment as described above often fails or delivers slightly incorrect results, in case of non-convex error functions. A well known example is the ambiguity between rotation and translation in case of camera calibration. The error function then has local minima, which the optimization can present as the final result. In order to circumvent this problem with non-convex error functions, different optimization strategies are required. In addition, it is often not possible or at least very involved to derive analytic derivations of the error function. In this case, either numeric derivations or

[2] http://maxima.sourceforge.net/

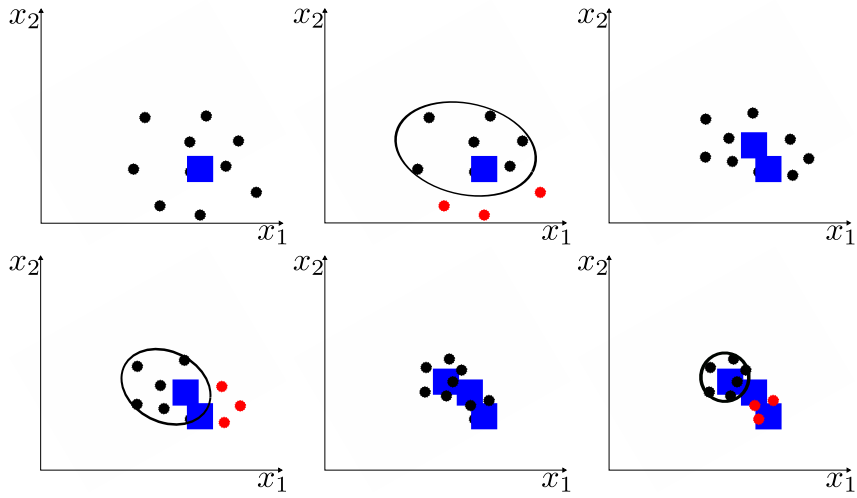

Figure A.2. CMA-ES algorithm. The green ellipse depicts the error function with the minimum being the center. The blue square is the initial solution and the black dots are the results of the fitness function of all samples of the first population, which are drawn according to the initial parameters variances. Then, the worst individuals of the population are discarded (red dots) and the parameter covariance is updated using the remaining samples. A new mean is computed (third image) and a new population is generated based on the updated covariance matrix. The images in the bottom row show how those steps are repeated and the mean gradually converges against the minimum (center of the green ellipse) while the parameter covariance is learned. Reprint from [JSK12].

derivative-free optimization strategies need to be applied. One example for an optimization algorithm fulfilling both criteria is CMA-ES, short for Covariance Matrix Adaptation Evolution Strategy described by Hansen and Ostermeier in [HO01]. It can optimize non-linear, non-convex error functions of the form $E : \mathbb{R}^n \rightarrow \mathbb{R}_{>0}$, given an initial solution $p \in \mathbb{R}^n$ and an expected initial deviation from this solution. Starting from this, the following steps are repeated for each generation. Based on current mean and deviation, a sample population of the size $\lambda = 4 + \text{floor}(3 * \log(n))$ is drawn and for each sample, the error function is evaluated. Figure A.2, top row on the left shows the current mean (blue rectangle) and the evaluated

fitness function values (black dots) for the different individuals of the current population. Then, in the second image, the worst individuals are discarded (red dots) and the covariance is updated (black circle). In addition, a new mean is computed (third image). The covariance and mean updates are based on current and older iterations. This is the reason why only so few samples are required in each iteration.

These steps are repeated until convergence, e. g., the mean fitness value does not change any more or a predefined certainty is reached for all parameters.

Note that by specifying the initial deviation and population size, the algorithm can be a global method (large deviations and population size) or a local method (small deviations and population size). In addition, the parameter covariance is learned more precisely over the generations, thus in the end, correlations between parameters and uncertainties of parameters are known and can be analyzed. In contrast to the derivative-based methods for non-linear least squares described above, the fitness function E used in CMA-ES can be far more general and especially functions with explicit and implicit constraints as described above can be used easily.

A.4.3 Robust Error Functions

In practical applications of the above non-linear least squares algorithm, the data usually contains outliers. In this case, a squared error function is known to weight outliers strongly. Therefore, it is often advisable to use error functions that are more robust than the least squares error function [Ste09, TMHF00]. If it is possible to assume the error in the observations to have a Gaussian distribution and if the variance of the single observations is known, the use a covariance matrix as described above already down-weights erroneous observations. However, in some cases it might help to replace the least squares error function with another, more robust error function that does not weigh the outliers as strongly.

In the algorithm described above, the error function was $\operatorname{argmin}_{\mathsf{p}} \Phi = \mathsf{v}^{\mathrm{T}} \mathbf{C}_{ll}^{-1} \mathsf{v}$, which can be replaced by another function from Table A.2.

A.4.4 Gauge Freedom

Triggs et al. [TMHF00] contains a chapter about gauge freedom, i. e., the different possibilities of fixing the reference frame, the global coordinate system for the reconstruction. That means that basically the bundle adjustment error function is invariant under similarity transformations, i. e., translation, rotation, and scale. Additionally, 3D points parametrized as homogeneous vectors can have an arbitrary scale, which can change without changing the error function in case of this being the perspective reprojection error. In order to prevent the optimization method to just move the scene around within the global coordinate system by similarity transformation, Gauge constraints are required. This can be achieved in different ways [McL00]. For this thesis, the first camera was simply fixed using h constraints as described above, thus fixing the scene in the world coordinate system. The scene scale was fixed by finding the longest distance between the first camera and the other camera poses and fixing its scale. In case of stereo rigs, the rig baseline was fixed to a certain distance.

Table A.2. Different error functions that can be used instead of the least squares error function.

Norm	Error Function	Weight Function	Plot
L_2	$\Phi(x) = \frac{1}{2}x^2$	$\Psi(x) = 1$	
L_1	$\Phi(x) = \lvert x \rvert$	$\Psi(x) = \frac{1}{\lvert x \rvert}$	
Brox [Bro05]	$\Phi(x) = \sqrt{x^2 + \epsilon^2}\ \epsilon \in \mathbb{R}_{>0}$		
Huber	$\Phi(x) = \begin{cases} \frac{x^2}{2} & \text{if } \lvert x \rvert \leqslant k \\ k\left(\lvert x \rvert - \frac{k}{2}\right) & \text{if } \lvert x \rvert > k \end{cases}$	$\Psi(x) = \begin{cases} 1 & \text{if } \lvert x \rvert \leqslant k \\ \frac{k}{\lvert x \rvert} & \text{if } \lvert x \rvert > k \end{cases}$	
Welch	$\Phi(x) = \frac{k^2}{2}\left(1 - \exp\left(\frac{-x^2}{k^2}\right)\right)$	$\Psi(x) = \exp\left(\frac{-x^2}{k^2}\right)$	
Tukey	$\Phi(x) = \begin{cases} \frac{k^2}{6}\left(1 - \left(1 - \frac{x^2}{k^2}\right)^3\right) & \text{if } \lvert x \rvert \leqslant k \\ \frac{k^2}{6} & \text{if } \lvert x \rvert > k \end{cases}$	$\Psi(x) = \begin{cases} \left(1 - \frac{x^2}{k^2}\right)^2 & \text{if } \lvert x \rvert \leqslant k \\ 0 & \text{if } \lvert x \rvert > k \end{cases}$	
Triggs [TMHF00]	$\Phi(x) = -\log\left(\exp(-\frac{1}{2}x^2) + \epsilon\right)$		

Table A.3. Calibration results for the camera's intrinsics calibrated using checkerboard images captured in air. The true camera baselines were about $(-60, 0, 0)^{\mathrm{T}}$ and $(-50, 0, 0)^{\mathrm{T}}$.

scenario	cam.	f_1	(p_x, p_y)	r_1	r_2	C_{rig}
c), e), f), g)	cam. 1	912.14	(404.97, 317.85)	-0.23	0.16	
	cam. 2	910.45	(396.85, 326.42)	-0.23	0.19	
	rig	912.14	(404.97, 317.85)	-0.23	0.16	
		910.45	(396.85, 326.42)	-0.23	0.19	$(-60.83, -4.25, -1.53)^{\mathrm{T}}$
a), b), d)	cam. 1	909.69	(406.54, 322.10)	-0.24	0.26	
	cam. 2	910.13	(385.88, 318.37)	-0.22	0.14	
	rig	909.69	(406.54, 322.10)	-0.24	0.26	
		910.13	(385.88, 318.37)	-0.22	0.14	$(-45.27, 0.74, -0.82)^{\mathrm{T}}$

A.5 Real Data Calibration Results

Chapter 4 showed calibration results for the seven different camera-interface configurations a) - g). Here more detailed results of calibrating perspectively are shown. First, Table A.3 gives results for calibrating the intrinsic parameters in air using the method described in [SBK08]. The first column shows for which of the seven scenarios (Figure 4.11) the intrinsic calibration is valid. The true baselines between the two cameras were about $(-60, 0, 0)^{\mathrm{T}}$ and $(-50, 0, 0)^{\mathrm{T}}$ in mm, thus the result in the second case was not very accurate. Table A.4 shows the results of calibrating the perspective camera model on underwater images for all seven image sets. The seven cases are ordered such that the interface distance increases. It is difficult to distinguish dependencies comparable to the results of the synthetic data, except for the principal point calibration in cases d and e, which both had a strongly tilted interface. The results on distortion cannot be compared to the synthetic case directly because the perspective cameras did not have zero distortion (Table A.3). However, r_1 increases with increasing interface distance.

Table A.4. Calibration results of perspective calibration on underwater images.

scenario	f_1	(p_x, p_y)	r_1	r_2	C_{rig}
a) camera 1	1216.93	(436.02, 328.62)	0.03	0.25	
a) camera 2	1219.70	(416.74, 336.31)	-0.04	0.46	
a) rig	1231.43	(427.68, 346.18)	0.02	0.03	
	1205.58	(429.14, 321.07)	-0.09	0.74	$(-49.10, -0.11, 12.70)^T$
b) camera 1	1215.59	(410.69, 327.21)	-0.09	0.28	
c) camera 1	1218.36	(393.30, 332.28)	-0.10	0.51	
c) camera 2	1211.17	(381.86, 332.36)	-0.05	0.18	
c) rig	1213.08	(411.48, 339.90)	-0.11	0.42	
	1209.05	(400.79, 339.72)	-0.06	0.22	$(-59.56, 0.18, -1.72)^T$
d) camera 1	1231.00	(**494.76**, 327.45)	-0.09	0.25	
d) camera 2	1221.19	(**475.96**, 326.94)	-0.03	0.07	
d) rig	1232.77	(**503.66**, 315.08)	-0.08	0.25	
	1231.14	(**489.49**, 305.63)	-0.06	0.24	$(-50.29, -0.88, 0.89)^T$
e) camera 1	1287.15	(**206.93**, 305.27)	-0.01	0.06	
e) camera 2	1264.86	(**190.41**, 317.01)	-0.11	0.18	
e) rig	1293.56	(**196.21**, 306.69)	-0.04	0.17	
	1263.17	(**199.32**, 307.82)	-0.12	0.18	$(-61.60, 0.14, 1.41)^T$
f) camera 1	1222.31	(395.65, 329.83)	-0.11	0.21	
f) camera 2	1218.09	(380.51, 333.25)	-0.13	0.34	
f) rig	1225.24	(390.54, 325.64)	-0.11	0.30	
	1223.16	(384.39, 331.06)	-0.12	0.27	$(-59.73, 0.13, -1.29)^T$
g) camera 1	1224.65	(389.66, 324.41)	-0.13	-0.02	
g) camera 2	1215.26	(371.14, 331.47)	-0.16	0.30	
g) rig	1223.32	(397.45, 324.04)	-0.14	0.13	
	1216.60	(378.27, 331.00)	-0.15	0.23	$(-59.61, -0.27, 0.36)^T$

A.6 Equation Systems

Table A.5. Coefficients for matrices $\mathbf{A}_\mathrm{E}$ and $\mathbf{A}_\mathrm{R}$ for relative pose using linear estimation.

variable	coefficient	variable	coefficient
e_{11}	$-\tilde{\mathsf{X}}_{\mathsf{w}_1}\tilde{\mathsf{X}'}_{\mathsf{w}_1}$	r_{11}	$\tilde{\mathsf{X}}_{\mathsf{w}_1}\mathsf{M}'_1 + \mathsf{M}_1\tilde{\mathsf{X}'}_{\mathsf{w}_1}$
e_{12}	$-\tilde{\mathsf{X}}_{\mathsf{w}_1}\tilde{\mathsf{X}'}_{\mathsf{w}_2}$	r_{12}	$\tilde{\mathsf{X}}_{\mathsf{w}_1}\mathsf{M}'_2 + \mathsf{M}_1\tilde{\mathsf{X}'}_{\mathsf{w}_2}$
e_{13}	$-\tilde{\mathsf{X}}_{\mathsf{w}_1}\tilde{\mathsf{X}'}_{\mathsf{w}_3}$	r_{13}	$\tilde{\mathsf{X}}_{\mathsf{w}_1}\mathsf{M}'_3 + \mathsf{M}_1\tilde{\mathsf{X}'}_{\mathsf{w}_3}$
e_{21}	$-\tilde{\mathsf{X}}_{\mathsf{w}_2}\tilde{\mathsf{X}'}_{\mathsf{w}_1}$	r_{21}	$\tilde{\mathsf{X}}_{\mathsf{w}_2}\mathsf{M}'_1 + \mathsf{M}_2\tilde{\mathsf{X}'}_{\mathsf{w}_1}$
e_{22}	$-\tilde{\mathsf{X}}_{\mathsf{w}_2}\tilde{\mathsf{X}'}_{\mathsf{w}_2}$	r_{22}	$\tilde{\mathsf{X}}_{\mathsf{w}_2}\mathsf{M}'_2 + \mathsf{M}_2\tilde{\mathsf{X}'}_{\mathsf{w}_2}$
e_{23}	$-\tilde{\mathsf{X}}_{\mathsf{w}_2}\tilde{\mathsf{X}'}_{\mathsf{w}_3}$	r_{23}	$\tilde{\mathsf{X}}_{\mathsf{w}_2}\mathsf{M}'_3 + \mathsf{M}_2\tilde{\mathsf{X}'}_{\mathsf{w}_3}$
e_{31}	$-\tilde{\mathsf{X}}_{\mathsf{w}_3}\tilde{\mathsf{X}'}_{\mathsf{w}_1}$	r_{31}	$\tilde{\mathsf{X}}_{\mathsf{w}_3}\mathsf{M}'_1 + \mathsf{M}_3\tilde{\mathsf{X}'}_{\mathsf{w}_1}$
e_{32}	$-\tilde{\mathsf{X}}_{\mathsf{w}_3}\tilde{\mathsf{X}'}_{\mathsf{w}_2}$	r_{32}	$\tilde{\mathsf{X}}_{\mathsf{w}_3}\mathsf{M}'_2 + \mathsf{M}_3\tilde{\mathsf{X}'}_{\mathsf{w}_2}$
e_{33}	$-\tilde{\mathsf{X}}_{\mathsf{w}_3}\tilde{\mathsf{X}'}_{\mathsf{w}_3}$	r_{33}	$\tilde{\mathsf{X}}_{\mathsf{w}_3}\mathsf{M}'_1 + \mathsf{M}_3\tilde{\mathsf{X}'}_{\mathsf{w}_3}$

Table A.6. Coefficients in matrix $\mathbf{A}_C$ for retrieving C after determining the generalized essential matrix.

variable	coefficient
C_1	$-\tilde{\mathsf{X}}_{\mathsf{w}_2}\tilde{\mathsf{X}'}_{\mathsf{w}_1}r_{31} - \tilde{\mathsf{X}}_{\mathsf{w}_2}\tilde{\mathsf{X}'}_{\mathsf{w}_2}r_{32} - \tilde{\mathsf{X}}_{\mathsf{w}_2}\tilde{\mathsf{X}'}_{\mathsf{w}_3}r_{33} + \tilde{\mathsf{X}}_{\mathsf{w}_3}\tilde{\mathsf{X}'}_{\mathsf{w}_1}r_{21} + \tilde{\mathsf{X}}_{\mathsf{w}_3}\tilde{\mathsf{X}'}_{\mathsf{w}_2}r_{22} + \tilde{\mathsf{X}}_{\mathsf{w}_3}\tilde{\mathsf{X}'}_{\mathsf{w}_3}r_{23}$
C_2	$\tilde{\mathsf{X}}_{\mathsf{w}_1}\tilde{\mathsf{X}'}_{\mathsf{w}_1}r_{31} + \tilde{\mathsf{X}}_{\mathsf{w}_1}\tilde{\mathsf{X}'}_{\mathsf{w}_2}r_{32} + \tilde{\mathsf{X}}_{\mathsf{w}_1}\tilde{\mathsf{X}'}_{\mathsf{w}_3}r_{33} - \tilde{\mathsf{X}}_{\mathsf{w}_3}\tilde{\mathsf{X}'}_{\mathsf{w}_1}r_{11} - \tilde{\mathsf{X}}_{\mathsf{w}_3}\tilde{\mathsf{X}'}_{\mathsf{w}_2}r_{12} - \tilde{\mathsf{X}}_{\mathsf{w}_3}\tilde{\mathsf{X}'}_{\mathsf{w}_3}r_{13}$
C_3	$-\tilde{\mathsf{X}}_{\mathsf{w}_1}\tilde{\mathsf{X}'}_{\mathsf{w}_1}r_{21} - \tilde{\mathsf{X}}_{\mathsf{w}_1}\tilde{\mathsf{X}'}_{\mathsf{w}_2}r_{22} - \tilde{\mathsf{X}}_{\mathsf{w}_1}\tilde{\mathsf{X}'}_{\mathsf{w}_3}r_{23} + \tilde{\mathsf{X}}_{\mathsf{w}_2}\tilde{\mathsf{X}'}_{\mathsf{w}_1}r_{11} + \tilde{\mathsf{X}}_{\mathsf{w}_2}\tilde{\mathsf{X}'}_{\mathsf{w}_2}r_{12} + \tilde{\mathsf{X}}_{\mathsf{w}_2}\tilde{\mathsf{X}'}_{\mathsf{w}_3}r_{13}$
b_1	$\tilde{\mathsf{X}}_{\mathsf{w}_1}(r_{11}\mathsf{M}'_1 + r_{12}\mathsf{M}'_2 + r_{13}\mathsf{M}'_3) + \mathsf{M}_1(r_{11}\tilde{\mathsf{X}'}_{\mathsf{w}_1} + r_{12}\tilde{\mathsf{X}'}_{\mathsf{w}_2} + r_{13}\tilde{\mathsf{X}'}_{\mathsf{w}_3})$
b_2	$\tilde{\mathsf{X}}_{\mathsf{w}_2}(r_{21}\mathsf{M}'_1 + r_{22}\mathsf{M}'_2 + r_{23}\mathsf{M}'_3) + \mathsf{M}_2(r_{21}\tilde{\mathsf{X}'}_{\mathsf{w}_1} + r_{22}\tilde{\mathsf{X}'}_{\mathsf{w}_2} + r_{23}\tilde{\mathsf{X}'}_{\mathsf{w}_3})$
b_3	$\tilde{\mathsf{X}}_{\mathsf{w}_3}(r_{31}\mathsf{M}'_1 + r_{32}\mathsf{M}'_2 + r_{33}\mathsf{M}'_3) + \mathsf{M}_3(r_{31}\tilde{\mathsf{X}'}_{\mathsf{w}_1} + r_{32}\tilde{\mathsf{X}'}_{\mathsf{w}_2} + r_{33}\tilde{\mathsf{X}'}_{\mathsf{w}_3})$

Table A.7. Coefficients for building matrix for iterative approach for relative pose estimation using Equation (5.1.17).

variable	coefficient equation 1	coefficient equation 2
r_{11}	0	$-{X'}_{s_1}\tilde{X}_{w_3} - \kappa'{\tilde{X}'}_{w_1}\tilde{X}_{w_3}$
r_{12}	0	$-{X'}_{s_2}\tilde{X}_{w_3} - \kappa'{\tilde{X}'}_{w_2}\tilde{X}_{w_3}$
r_{13}	0	$-{X'}_{s_3}\tilde{X}_{w_3} - \kappa'{\tilde{X}'}_{w_3}\tilde{X}_{w_3}$
r_{21}	${X'}_{s_1}\tilde{X}_{w_3} + \kappa'{\tilde{X}'}_{w_1}\tilde{X}_{w_3}$	0
r_{22}	${X'}_{s_2}\tilde{X}_{w_3} + \kappa'{\tilde{X}'}_{w_2}\tilde{X}_{w_3}$	0
r_{23}	${X'}_{s_3}\tilde{X}_{w_3} + \kappa'{\tilde{X}'}_{w_3}\tilde{X}_{w_3}$	0
r_{31}	$-{X'}_{s_1}\tilde{X}_{w_2} - \kappa'{\tilde{X}'}_{w_1}\tilde{X}_{w_2}$	${X'}_{s_1}\tilde{X}_{w_1} + \kappa'{\tilde{X}'}_{w_1}\tilde{X}_{w_1}$
r_{32}	$-{X'}_{s_2}\tilde{X}_{w_2} - \kappa'{\tilde{X}'}_{w_2}\tilde{X}_{w_2}$	${X'}_{s_2}\tilde{X}_{w_1} + \kappa'{\tilde{X}'}_{w_2}\tilde{X}_{w_1}$
r_{33}	$-{X'}_{s_3}\tilde{X}_{w_2} - \kappa'{\tilde{X}'}_{w_3}\tilde{X}_{w_2}$	${X'}_{s_3}\tilde{X}_{w_1} + \kappa'{\tilde{X}'}_{w_3}\tilde{X}_{w_1}$
C_1	0	$-\tilde{X}_{w_3}$
C_2	$\tilde{X}_{w_3}$	0
C_3	$-\tilde{X}_{w_2}$	$\tilde{X}_{w_1}$

Table A.8. Coefficients for matrix **A** based on constraints on FRC and POR for absolute pose estimation.

variable	FRC equation 1	FRC equation 2	FRC equation 3	POR equation 1
r_{11}	0	$\tilde{\mathrm{X}}_{\mathrm{w}_3} X$	$-\tilde{\mathrm{X}}_{\mathrm{w}_2} X$	$X(\tilde{\mathrm{n}} \times \tilde{\mathrm{X}}_{\mathrm{w}})_1$
r_{12}	0	$\tilde{\mathrm{X}}_{\mathrm{w}_3} Y$	$-\tilde{\mathrm{X}}_{\mathrm{w}_2} Y$	$Y(\tilde{\mathrm{n}} \times \tilde{\mathrm{X}}_{\mathrm{w}})_1$
r_{13}	0	$\tilde{\mathrm{X}}_{\mathrm{w}_3} Z$	$-\tilde{\mathrm{X}}_{\mathrm{w}_2} Z$	$Z(\tilde{\mathrm{n}} \times \tilde{\mathrm{X}}_{\mathrm{w}})_1$
r_{21}	$-\tilde{\mathrm{X}}_{\mathrm{w}_3} X$	0	$\tilde{\mathrm{X}}_{\mathrm{w}_1} Z$	$X(\tilde{\mathrm{n}} \times \tilde{\mathrm{X}}_{\mathrm{w}})_2$
r_{22}	$-\tilde{\mathrm{X}}_{\mathrm{w}_3} Y$	0	$\tilde{\mathrm{X}}_{\mathrm{w}_1} Z$	$Y(\tilde{\mathrm{n}} \times \tilde{\mathrm{X}}_{\mathrm{w}})_2$
r_{23}	$-\tilde{\mathrm{X}}_{\mathrm{w}_3} Z$	0	$\tilde{\mathrm{X}}_{\mathrm{w}_1} Z$	$Z(\tilde{\mathrm{n}} \times \tilde{\mathrm{X}}_{\mathrm{w}})_2$
r_{31}	$\tilde{\mathrm{X}}_{\mathrm{w}_2} X$	$-\tilde{\mathrm{X}}_{\mathrm{w}_1} X$	0	$X(\tilde{\mathrm{n}} \times \tilde{\mathrm{X}}_{\mathrm{w}})_3$
r_{32}	$\tilde{\mathrm{X}}_{\mathrm{w}_2} Y$	$-\tilde{\mathrm{X}}_{\mathrm{w}_1} Y$	0	$Y(\tilde{\mathrm{n}} \times \tilde{\mathrm{X}}_{\mathrm{w}})_3$
r_{33}	$\tilde{\mathrm{X}}_{\mathrm{w}_2} Z$	$-\tilde{\mathrm{X}}_{\mathrm{w}_1} Z$	0	$Z(\tilde{\mathrm{n}} \times \tilde{\mathrm{X}}_{\mathrm{w}})_3$
C_1	0	$\tilde{\mathrm{X}}_{\mathrm{w}_3}$	$-\tilde{\mathrm{X}}_{\mathrm{w}_1}$	$(\tilde{\mathrm{n}} \times \tilde{\mathrm{X}}_{\mathrm{w}})_1$
C_2	$-\tilde{\mathrm{X}}_{\mathrm{w}_3}$	0	$\tilde{\mathrm{X}}_{\mathrm{w}_1}$	$(\tilde{\mathrm{n}} \times \tilde{\mathrm{X}}_{\mathrm{w}})_2$
C_3	$\tilde{\mathrm{X}}_{\mathrm{w}_2}$	$-\tilde{\mathrm{X}}_{\mathrm{w}_1}$	0	$(\tilde{\mathrm{n}} \times \tilde{\mathrm{X}}_{\mathrm{w}})_3$
b	$\tilde{\mathrm{X}}_{\mathrm{w}_2} \mathrm{X}_{\mathrm{s}_3} - \tilde{\mathrm{X}}_{\mathrm{w}_3} \mathrm{X}_{\mathrm{s}_2}$	$\tilde{\mathrm{X}}_{\mathrm{w}_3} \mathrm{X}_{\mathrm{s}_1} - \tilde{\mathrm{X}}_{\mathrm{w}_1} \mathrm{X}_{\mathrm{s}_3}$	$\tilde{\mathrm{X}}_{\mathrm{w}_1} \mathrm{X}_{\mathrm{s}_2} - \tilde{\mathrm{X}}_{\mathrm{w}_2} \mathrm{X}_{\mathrm{s}_3}$	0

Bibliography

[AA02] M. Aggarwal and N. Ahuja. A pupil-centric model of image formation. *International Journal of Computer Vision*, 48(3):195–214, 2002.

[ARTC12] A. Agrawal, S. Ramalingam, Y. Taguchi, and V. Chari. A theory of multi-layer flat refractive geometry. In *CVPR*, 2012.

[BAP+07] V. Brandou, A. G. Allais, M. Perrier, E. Malis, P. Rives, J. Sarrazin, and P. M. Sarradin. 3D reconstruction of natural underwater scenes using the stereovision system iris. In *Proc. OCEANS 2007 - Europe*, pages 1–6, 2007.

[BBD08] M. Brückner, F. Bajramovic, and J. Denzler. Experimental evaluation of relative pose estimation algorithms. In *VISAPP (2)*, pages 431–438, 2008.

[BFS+10] B. Bingham, B. Foley, H. Singh, R. Camilli, K. Delaporta, R. Eustice, A. Mallios, D. Mindell, C. Roman, and D. Sakellariou. Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle. *J. Field Robotics*, 27:702–717, 2010.

[BG04] E. T. Baker and C. R. German. On the global distribution of mid-ocean ridge hydrothermal vent-fields. *American Geophysical Union Geophysical Monograph*, 148:245–266, 2004.

[BLID10] C. Beall, B. J. Lawrence, V. Ila, and F. Dellaert. 3D reconstruction of underwater structures. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4418–4423, 2010.

[BQJM06] S. Bazeille, I. Quidu, L. Jaulin, and J.-P. Malkasse. Automatic underwater image pre-processing. In *Proceedings of the*

Charactersation du Milieu Marin (CMM06), pages 16–19, 10 2006.

[Bro71] D. C. Brown. Close-range camera calibration. champ, 1971.

[Bro05] T. Brox. *From pixels to regions: partial differential equations in image analysis*. PhD thesis, Faculty of Mathematics and Computer Science, Saarland University, Germany, 4 2005.

[BS99] I. N. Bronstein and K. A. Semendjajew. *Taschenbuch der Mathematik*. 1999.

[BWAZ00] M. Bryant, D. Wettergreen, S. Abdallah, and A. Zelinsky. Robust camera calibration for an autonomous underwater vehicle. In *Australian Conference on Robotics and Automation (ACRA 2000)*, 8 2000.

[CC11] Y.-J. Chang and T. Chen. Multi-view 3D reconstruction for scenes under the refractive plane with known vertical direction. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[CLC+06] C. Costa, A. Loy, S. Cataudella, D. Davis, and M. Scardi. Extracting fish size using dual underwater cameras. *Aquacultural Engineering*, 35(3):218–227, 2006.

[CRGN03] M. Carreras, P. Ridao, R. Garcia, and T. Nicosevici. Vision-based localization of an underwater robot in a structured environment. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 1, pages 971–976 vol.1, 9 2003.

[CS02] E. Cerezo and F. J. Serón. Rendering natural waters: merging computer graphics with physics and biology. In *in: Proceedings of Computer Graphics international CGI'02, 2002*, pages 481–498, 2002.

[CS09] V. Chari and P. Sturm. Multiple-view geometry of the refractive plane. In *Proceedings of the 20th British Machine Vision Conference, London, UK*, 9 2009.

[Dav06] T.A. Davis. *Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms 2)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.

[DCGG11] E. Darles, B. Crespin, D. Ghazanfarpour, and J.-C. Gonzato. A survey of ocean simulation and rendering techniques in computer graphics. *Comput. Graph. Forum*, 30(1):43–60, 2011.

[DD95] D. F. Dementhon and L. S. Davis. Model-based object pose in 25 lines of code. *Int. J. Comput. Vision*, 15:123–141, 6 1995.

[Der92] J. Dera. *Marine Physics*. Elsevier Oceanography Series, 1992.

[EeK96] J. Encarançao, W. Straßer, and R. Klein. *Graphische Datenverarbeitung 1*. Oldenbourg Verlag München Wien, 1996.

[ESH00] R. Eustice, H. Singh, and J. Howland. Image registration underwater for fluid flow measurements and mosaicking. In *OCEANS 2000 MTS/IEEE Conference and Exhibition*, volume 3, pages 1529–1534 vol.3, 2000.

[ESN06] C. Engels, H. Stewénius, and D. Nistér. Bundle adjustment rules. In *Photogrammetric Computer Vision (PCV)*. ISPRS, September 2006.

[FB81] M. Fischler and R. Bolles. RANdom SAmpling Consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 6 1981.

[FCS05] R. Ferreira, J.p. Costeira, and J.A. Santos. Stereo reconstruction of a submerged scene. In J. Marques, N. Pérez de la Blanca, and P. Pina, editors, *Pattern Recognition and Image Analysis*, volume 3522 of *Lecture Notes in Computer Science*, pages 102–109. Springer Berlin / Heidelberg, 2005.

[FF86] J. G. Fryer and C. S. Fraser. On the calibration of underwater cameras. *The Photogrammetric Record*, 12:73–85, 1986.

[FFG09] M. Farenzena, A. Fusiello, and R. Gherardi. Structure-and-motion pipeline on a hierarchical cluster tree. In *3DIM09*, pages 1489–1496, 2009.

[FP10] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

[FW00] W. Förstner and K. Wolff. Exploiting the multi view geometry for automatic surfaces reconstruction using feature based matching in multi media photogrammetry. In *Proceedings of the 19th ISPRS Congress*, pages 5B 900–907, 2000.

[GBCA01] R. Garcia, J. Batlle., X. Cufi, and J. Amat. Positioning an underwater vehicle through image mosaicking. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2779–2784 vol.3, 2001.

[GFF10] R. Gherardi, M. Farenzena, and A. Fusiello. Improving the efficiency of hierarchical structure-and-motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, 2010.

[GGY11] J. Gedge, M. Gong, and Y.-H. Yang. Refractive epipolar geometry for underwater stereo matching. In *Computer and Robot Vision (CRV), 2011 Canadian Conference on*, pages 146–152, 2011.

[Gla94] A. S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994.

[GN05] M. D. Grossberg and S. K. Nayar. The raxel imaging model and ray-based calibration. *International Journal of Computer Vision*, 61(2):119–137, 2005.

[GS00] G. Glaeser and H.-P. Schröcker. Reflections on refractions. *Journal for Geometry and Graphics (JGG)*, 4:1–18, 2000.

[GSMA08] D. Gutierrez, F. Seron, A. Munoz, and O. Anson. Visualizing underwater ocean optics. *Computer Graphics Forum (Proc. of EUROGRAPHICS)*, 27(2):547–556, 2008.

[GSV00] N. Gracias and J. Santos-Victor. Underwater video mosaics as visual navigation maps. *Computer Vision and Image Understanding, Journal of (CVIU)*, 79(1):66–91, 7 2000.

[GvdZBSV03] N. R. Gracias, S. van der Zwaan, A. Bernardino, and J. Santos-Victor. Mosaic-based navigation for autonomous underwater vehicles. *Oceanic Engineering, IEEE Journal of*, 28(4):609–624, oct. 2003.

[Hec05] E. Hecht. *Optik*. Oldenburg Verlag München Wien, 2005.

[HGJ07] A. Hogue, A. German, and M. Jenkin. Underwater environment reconstruction using stereo and inertial data. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pages 2372–2377, 7-10 2007.

[HGWA08] W. Hou, D. J. Gray, A. D. Weidemann, and R. A. Arnone. Comparison and validation of point spread models for imaging in natural waters. *Opt. Express*, 16(13):9958–9965, 2008.

[HGZJ06] A. Hogue, A. German, J. Zacher, and M. Jenkin. Underwater 3D mapping: Experiences and lessons learned. In *Computer and Robot Vision, 2006. The 3rd Canadian Conference on*, 2006.

[Hir01] H. Hirschmueller. Improvements in real-time correlation-based stereo vision. In *Proc. of IEEE Workshop on Stereo and Multi-Baseline Vision, Kauai, Hawaii*, 2001.

[HLON94] B. M. Haralick, C.-N. Lee, K. Ottenberg, and M. Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356, 12 1994.

[HMS99] E. Hering, R. Martin, and M. Stohrer. *Physik für Ingenieure*. Springer-Verlag, 1999.

[HO01] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[HS97a] R. I. Hartley and P. F. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.

[HS97b] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1106–1112, 1997.

[HS98] E. S. Harvey and M. R. Shortis. Calibration stability of an underwater stereo-video system : Implications for measurement accuracy and precision. *Marine Technology Society journal*, 32:3–17, 1998.

[HSXS13] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *Cybernetics, IEEE Transactions on*, PP(99):1318 – 1334, 2013.

[HWGF07] W. Hou, A. D. Weidemann, D. J. Gray, and G. R. Fournier. Imagery-derived modulation transfer function and its applications for underwater imaging. In A. G. Tescher, editor, *Proc. of SPIE*, volume 6696, page 669622. SPIE, 2007.

[HZ04] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision (Second Edition)*. Cambridge University Press, second edition, 2004.

[IDN02] K. Iwasaki, Y. Dobashi, and T. Nishita. An efficient method for rendering underwater optical effects using graphics hardware. *Computer Graphics Forum*, 21(4):701–711, 2002.

[ISOT07] K. Iqbal, R. A. Salam, A. Osman, and A. Z. Talib. Underwater image enhancement using an integrated colour model. *IAENG International Journal of Computer Science*, 34:2, 2007.

[ISVR12] G. Inglis, C. Smart, I. Vaughn, and C. Roman. A pipeline for structured light bathymetric mapping. In *Intelligent Robots*

and Systems (IROS), 2012 IEEE/RSJ International Conference on, pages 4425–4432, 2012.

[Jaf90] J. S. Jaffe. Computer modeling and the design of optimal underwater imaging systems. *IEEE Journal of Oceanic Engineering*, 15(2):101–111, 1990.

[JC98] H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scences with participating media using photon maps. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 311–320, New York, NY, USA, 1998. ACM.

[JDL12] P. Jasiobedzki, C. F. Dimas, and D. Lim. Underwater 3D modelling and photosynthetic life detection. In *Oceans, 2012*, pages 1–9, 2012.

[Jer76] N. G. Jerlov. *Marine Optics*. Elsevier Scientific Publishing Company, 1976.

[JHG99] B. Jähne, H. W. Haussecker, and P. Geissler. *Handbook of Computer Vision and Applications. 1. Sensors and Imaging*. Academic Press, 1999.

[JK11] A. Jordt and R. Koch. Fast tracking of deformable objects in depth and colour video. In S. McKenna, J. Hoey, and M. Trucco, editors, *Proceedings of the British Machine Vision Conference, BMVC 2011*. British Machine Vision Association, 2011.

[JNS+10] Y. Jeong, D. Nistér, D. Steedly, R. Szeliski, and I.-S. Kweon. Pushing the envelope of modern methods for bundle adjustment. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1474–1481, 2010.

[JNS+12] Y. Jeong, D. Nistér., D. Steedly, R. Szeliski, and I.-S. Kweon. Pushing the envelope of modern methods for bundle adjustment. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(8):1605–1617, 2012.

[JRPWM10] M. Johnson-Roberson, O. Pizarro, S. B. Williams, and I. J. Mahon. Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27, 2010.

[JSBJ08] P. Jasiobedzki, S. Se, M. Bondy, and R. Jakola. Underwater 3D mapping and pose estimation for rov operations. In *OCEANS 2008*, pages 1 –6, 15-18 2008.

[JSJK13] A. Jordt-Sedlazeck, D. Jung, and R. Koch. Refractive plane sweep for underwater images. In J. Weickert, M. Hein, and B. Schiele, editors, *Pattern Recognition*, volume 8142 of *Lecture Notes in Computer Science*, pages 333–342. Springer Berlin Heidelberg, 2013.

[JSK12] A. Jordt-Sedlazeck and R. Koch. Refractive calibration of underwater cameras. In A. Fitzgibbon, S. Lazebnik, P. Pietro, Y. Sato, and C. Schmid, editors, *Computer Vision - ECCV 2012*, volume 7576 of *Lecture Notes in Computer Science*, pages 846–859. Springer Berlin Heidelberg, 2012.

[JSK13] A. Jordt-Sedlazeck and R. Koch. Refractive structure-from-motion on underwater images. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 57–64, 2013.

[KC06] Y. Kwon and J. B. Casebolt. Effects of light refraction on the accuracy of camera calibration and reconstruction in underwater motion analysis. *Sports Biomech*, 5(1):95–120, 2006.

[KE13] A. Kim and R. M. Eustice. Real-time visual slam for autonomous underwater hull inspection using visual saliency. *IEEE Transactions on Robotics*, 29(3):719–733, June 2013.

[KHDK13] T. Kwasnitschka, T. H. Hansteen, C. W. Devey, and S. Kutterolf. Doing fieldwork on the seafloor: Photogrammetric techniques to yield 3d visual models from rov video. *Computers & Geosciences*, 52:218–226, 2013.

[Koc93] R. Koch. Dynamic 3-D scene analysis through synthesis feedback control. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(6):556–568, June 1993.

[KS08] C. Kunz and H. Singh. Hemispherical refraction and camera calibration in underwater vision. In *OCEANS 2008*, pages 1–7, 15-18 2008.

[KTS11] C. Kurz, T. Thormählen, and H.-P. Seidel. Bundle adjustment for stereoscopic 3D. In *MIRAGE*, pages 1–12, 2011.

[Kwo99] Y. Kwon. A camera calibration algorithm for the underwater motion analysis. In *ISBS - Conference Proceedings Archive, 17 International Symposium on Biomechanics in Sports (1999)*, 1999.

[KWY12a] L. Kang, L. Wu, and Y.-H. Yang. Experimental study of the influence of refraction on underwater three-dimensional reconstruction using the svp camera model. *Appl. Opt.*, 51(31):7591–7603, 11 2012.

[KWY12b] L. Kang, L. Wu, and Y.-H. Yang. Two-view underwater structure and motion for cameras under flat refractive interfaces. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision - ECCV 2012*, volume 7575 of *Lecture Notes in Computer Science*, pages 303–316. Springer Berlin / Heidelberg, 2012.

[KYK09] R. Kawai, A. Yamashita, and T. Kaneko. Three-dimensional measurement of objects in water by using space encoding method. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 2830–2835, 12-17 2009.

[LA09] M. I. A. Lourakis and A. A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30, 2009.

[LC87] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH Computer Graphics*, 21(4):163–169, 7 1987.

[LdLC03] I. Leifer, G. de Leeuw, and L. H. Cohen. Optical measurement of bubbles: System design and application. *Journal of Atmospheric and Oceanic Technology*, 20:1317–1332, 2003.

[LHK08] H. Li, R. Hartley, and J.-H. Kim. A linear approach to motion estimation using generalized camera models. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, 7 2008.

[LLZ+97] R. Li, H. Li, W. Zou, R. G. Smith, and T. A. Curran. Quantitative photogrammetric analysis of digital underwater video imagery. *Oceanic Engineering, IEEE Journal of*, 22(2):364 –375, 4 1997.

[Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[LRL00] J.-M. Lavest, G. Rives, and J.-T. Lapresté. Underwater camera calibration. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II*, pages 654–668, 2000.

[LSBS99] R. Lange, P. Seitz, A. Biber, and R. Schwarte. Time-of-flight range imaging with acustom solid state image sensor. *Proc. SPIE Vol. 3823, p. 180-191, Laser Metrology and Inspection*, 3823:180–191, 1999.

[LTZ96] R. Li, C. Tao, and W. Zou. An underwater digital photogrammetric system for fishery geomatics. In *Intl. Archives of PRS*, volume Vol.XXXI, pages pp.319–323, 1996.

[Maa92] H.-G. Maas. *Digitale Photogrammetrie in der dreidimensionalen Stroemungsmesstechnik*. PhD thesis, Eidgenoessische Technische Hochschule Zuerich, 1992.

[Maa95] H.-G. Maas. New developments in multimedia photogrammetry. In *Optical 3-D Measurement Techniques III*. Wichmann Verlag, Karlsruhe, 1995.

[MBJ09] S. W. Moore, H. Bohm, and V. Jensen. *Underwater Robotics - Science, Design & Fabrication*. MATE Center/Monterey Peninsula College, 2009.

[McG75] B. L. McGlamery. Computer analysis and simulation of underwater camera system performance. Technical report, Visibility Laboratory, Scripps Institution of Oceanography, University of California in San Diego, 1975.

[McG04] J. C. McGlone, editor. *Manual of Photogrammetry*. ASPRS, 5th edition, 2004.

[McL00] P. F. McLauchlan. Gauge independence in optimization algorithms for 3D vision. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, pages 183–199, London, UK, UK, 2000. Springer-Verlag.

[MK05] N. Morris and K. N. Kutulakos. Dynamic refraction stereo. In *Proc. 10th Int. Conf. Computer Vision*, pages 1573–1580, 2005.

[MLD+07] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Generic and real-time structure from motion. In *BMVC*, 2007.

[MLD+09] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Generic and real-time structure from motion using local bundle adjustment. *Image and Vision Computing*, 27:1178–1193, 2009.

[Mob94] C. D. Mobley. *Light and Water: Radiative Transfer in Natural Waters*. Academic Press, 1994.

[MP77] A. Morel and L. Prieur. Analysis and variations in ocean color. *Limnology and Oceanography*, 22:709–722, 1977.

[NCdB09] E. R. Nascimento, M. F. M. Campos, and W. F. de Barros. Stereo based structure recovery of underwater scenes from automatically restored images. In L. G. Nonato and

J. Scharcanski, editors, *Proceedings SIBGRAPI 09 (Brazilian Symposium on Computer Graphics and Image Processing)*, Los Alamitos, 10 2009. IEEE Computer Society.

[Nis04] D. Nistér. An efficient solution to the five-point relative pose problem. *TPAMI*, 26:756–777, 2004.

[NN05] S. G. Narasimhan and S.K. Nayar. Structured light methods for underwater imaging: light stripe scanning and photometric stereo. In *Proceedings of 2005 MTS/IEEE OCEANS*, volume 3, pages 2610–2617, September 2005.

[NNSK05] S. G. Narasimhan, S. K. Nayar, B. Sun, and S. J. Koppal. Structured light in scattering media. In *IEEE International Conference on Computer Vision (ICCV)*, volume I, pages 420–427, Oct 2005.

[NS07] D. Nistér and H. Stewénius. A minimal solution to the generalised 3-Point pose problem. *Journal of Mathematical Imaging and Vision*, 27:67–79, 2007.

[NSP07] S. Negahdaripour, H. Sekkati, and H. Pirsiavash. Opti-acoustic stereo imaging, system calibration and 3-D reconstruction. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[NSP09] S. Negahdaripour, H. Sekkati, and H. Pirsiavash. Opti-acoustic stereo imaging: On system calibration and 3-D target reconstruction. *Image Processing, IEEE Transactions on*, 18(6):1203 –1214, jun. 2009.

[NXKA98] S. Negahdaripour, X. Xu, A. Khamene, and Z. Awan. 3-D motion and depth estimation from sea-floor images for mosaic-based station-keeping and navigation of rovs/auvs and high-resolution sea-floor mapping. In *Autonomous Underwater Vehicles, 1998. AUV'98. Proceedings Of The1998 Workshop on*, pages 191–200, 1998.

[PA01] S. Premoze and M. Ashikhmin. Rendering natural waters. *Comput. Graph. Forum*, 20(4):189–199, 2001.

[PARN04] S. Premoze, M. Ashikhmin, R. Ramamoorthi, and S. K. Nayar. Practical rendering of multiple scattering effects in participating media. In *Rendering Techniques*, pages 363–373, 2004.

[PDH+97] K. Pulli, T. Duchamp, H. Hoppe, J. McDonald., L. Shapiro, and W. Stuetzle. Robust meshes from multiple range maps. In *3-D Digital Imaging and Modeling, 1997. Proceedings., International Conference on Recent Advances in*, pages 205–211, 1997.

[Pes03a] N. Pessel. *Auto-Calibrage d'une Caméra en Milieu Sous-Marin*. PhD thesis, Université Montpellier II, 2003.

[PES03b] O. Pizarro, R. Eustice, and H. Singh. Relative pose estimation for instrumented, calibrated imaging platforms. In *DICTA*, pages 601–612, 2003.

[PES04] O. Pizarro, R. Eustice, and H. Singh. Large area 3D reconstructions from underwater surveys. In *Proc. MTTS/IEEE TECHNO-OCEANS '04*, volume 2, pages 678–687 Vol.2, 2004.

[Ple03] R. Pless. Using many cameras as one. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages 587–93, june 2003.

[POA03a] N. Pessel, J. Opderbecke, and M.-J. Aldon. Camera self-calibration in underwater environment. In *WSCG*, 2003.

[POA03b] N. Pessel, J. Opderbecke, and M.-J. Aldon. An experimental study of a robust self-calibration method for a single camera. In *3rd International Symposium on Image and Signal Processing and Analysis, ISPA'2003, sponsored by IEEE and EURASIP*, Rome, Italie, september 2003.

[PP09] C. Papadopoulos and G. Papaioannou. Realistic real-time underwater caustics and godrays. In *Proc. GraphiCon '09*, 2009.

[Put08a] T. Putze. Erweiterte verfahren zur mehrmedienphotogrammetrie komplexer körper. In *Beiträge der Oldenburger 3D-Tage 2008*. Herbert Wichmann Verlag, Heidelberg, 2008.

[Put08b] T. Putze. *Geometrische und stochastische Modelle zur Optimierung der Leistungsfaehigkeit des Stroemungsmessverfahrens 3D-PTV*. PhD thesis, Technische Universität Dresden, 2008.

[PVT+02] W. H. Press, W. T. Vetterling, S. A. Teukolsky, A. Saul, and B. P. Flannery. *Numerical Recipes in C++: the art of scientific computing*. Cambridge University Press, New York, NY, USA, 2nd edition, 2002.

[QNCBC04] J. P. Queiroz-Neto, R. Carceroni, W. Barros, and M. Campos. Underwater stereo. In *Proc. 17th Brazilian Symposium on Computer Graphics and Image Processing*, pages 170–177, October 17–20, 2004.

[RLS06] S. Ramalingam, S. K. Lodha, and P. Sturm. A generic structure-from-motion framework. *Computer Vision and Image Understanding*, 103(3):218–228, September 2006.

[SBK08] I. Schiller, C. Beder, and R. Koch. Calibration of a PMD camera using a planar calibration object together with a multi-camera setup. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume Vol. XXXVII. Part B3a, pages 297–302, Beijing, China, 2008. XXI. ISPRS Congress.

[SC10] R. Schettini and S. Corchs. Underwater image processing: State of the art of restoration and image enhancement methods. *EURASIP Journal on Advances in Signal Processing*, 2010, 2010.

[Sch05] O. Schreer. *Stereoanalyse und Bildsynthese*. Springer Verlag, 2005.

[SFF12] R. Steffen, J.-M. Frahm, and W. Förstner. Relative bundle adjustment based on trifocal constraints. In K. N. Kutulakos,

editor, *Trends and Topics in Computer Vision*, volume 6554 of *Lecture Notes in Computer Science*, pages 282–295. Springer Berlin Heidelberg, 2012.

[SGdA+10] C. Stoll, J. Gall, E. de Aguiar, S. Thrun, and C. Theobalt. Video-based reconstruction of animatable human characters. In *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA 2010)*, volume 29(6), pages 139–149, 2010.

[SGN03] R. Swaninathan, M. D. Grossberg, and S. K. Nayar. A perspective on distortions. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages 594–601, june 2003.

[SK04] Y. Y. Schechner and N. Karpel. Clear underwater vision. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2004*, volume 1, pages I–536–I–543, June 27–July 2, 2004.

[SK05] Y. Y. Schechner and N. Karpel. Recovery of underwater visibility and structure by polarization analysis. *IEEE Journal of Oceanic Engineering*, 30(3):570–587, 2005.

[SK11a] A. Sedlazeck and R. Koch. Calibration of housing parameters for underwater stereo-camera rigs. In *Proceedings of the British Machine Vision Conference*, pages 118.1–118.11. BMVA Press, 2011. http://dx.doi.org/10.5244/C.25.118.

[SK11b] A. Sedlazeck and R. Koch. Simulating deep sea underwater images using physical models for light attenuation, scattering, and refraction. In P. Eisert, J. Hornegger, and K. Polthier, editors, *VMV 2011: Vision, Modeling & Visualization*, number 978-3-905673-85-2, pages 49–56, Berlin, Germany, 2011. Eurographics Association.

[SK12] A. Sedlazeck and R. Koch. Perspective and non-perspective camera models in underwater imaging - overview and error analysis. In F. Dellaert, J.-M. Frahm, M. Pollefeys, L. Leal-Taixé, and B. Rosenhahn, editors, *Outdoor and Large-Scale*

Real-World Scene Analysis, volume 7474 of *Lecture Notes in Computer Science*, pages 212–242. Springer Berlin Heidelberg, 2012.

[SKK09] A. Sedlazeck, K. Köser, and R. Koch. 3D reconstruction based on underwater video from ROV kiel 6000 considering underwater imaging conditions. In *Proc. OCEANS '09. OCEANS 2009-EUROPE*, pages 1–10, May 11–14, 2009.

[SR04] P. F. Sturm and S. Ramalingam. A generic concept for camera calibration. In *ECCV (2)*, pages 1–13, 2004.

[SRL06] P. Sturm, S. Ramalingam, and S. Lodha. On calibration, structure from motion and multi-view geometry for generic camera models. In K. Daniilidis and R. Klette, editors, *Imaging Beyond the Pinhole Camera*, volume 33 of *Computational Imaging and Vision*. Springer, aug 2006.

[SSC02] M. Slater, A. Steed, and Y. Chrysanthou. *Computer Graphics and Virtual Environments, from Realism to Real-Time*. Addison Wesley, 2002.

[Ste09] R. Steffen. *Visual SLAM from image sequences acquired by unmanned aerial vehicles*. PhD thesis, Institute of Photogrammetry, University of Bonn, 2009.

[Sze11] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag, 2011.

[TDO+00] E. Trucco, A. Doull, F. Odone, A. Fusiello, and D. Lane. Dynamic video mosaicing and augmented reality for subsea inspection and monitoring. In *In Oceanology International, United Kingdom*, 2000.

[TF06] G. Telem and S. Filin. Calibration of consumer cameras in a multimedia environment. In *ASPERS 2006 Annual Conference*, 2006.

[TF10] G. Telem and S. Filin. Photogrammetric modeling of underwater environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(5):433–444, 2010.

[TM08] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors - a survey. *Foundations and Trends in Computer Graphics and Vision*, 2008.

[TMHF00] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000.

[TOA06] E. Trucco and A. T. Olmos-Antillon. Self-tuning underwater image restoration. *IEEE Journal of Oceanic Engineering*, 31(2):511–519, APR 2006.

[TS06] T. Treibitz and Y. Y. Schechner. Instant 3Descatter. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1861–1868, 2006.

[TS08] T. Treibitz and Y. Y. Schechner. Active polarization descattering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:385–399, 2008.

[Tsa87] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, 1987.

[TSS08] T. Treibitz, Y. Y. Schechner, and H. Singh. Flat refractive geometry. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR 2008*, pages 1–8, 2008.

[TZSB10] K. Thomanek, O. Zielinski, H. Sahling, and G. Bohrmann. Automated gas bubble imaging at sea floor; a new method of in situ gas flux quantification. *Ocean Science*, 6(2):549–562, 2010.

[vDP12] J. Schneider von Deimling and C. Papenberg. Technical note: Detection of gas bubble leakage via correlation of water column multibeam images. *Ocean Science*, 8(2):175–181, 2012.

[Vos91] K. J. Voss. Simple empirical model of the oceanic point spread function. *Appl. Opt.*, 30(18):2647–2651, Jun 1991.

[WACS11] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. In *CVPR*, pages 3057–3064, 2011.

[Wol07] K. Wolff. *Zur Approximation allgemeiner optischer Abbildungsmodelle und deren Anwendung auf eine geometrisch basierte Mehrbildzuordnung am Beispiel einer Mehrmedienabbildung*. PhD thesis, Rheinische Friedrich-Wilhelms-Universitaet Bonn, 2007.

[WSK10] R. Wulff, A. Sedlazeck, and R. Koch. Measuring in automatically reconstructed 3D models. In *Geoinformatik 2010*, 2010.

[WSK13] R. Wulff, A. Sedlazeck, and R. Koch. 3D reconstruction of archaeological trenches from photographs. In H. G. Bock, W. Jäger, and M. J. Winckler, editors, *Scientific Computing and Cultural Heritage*, volume 3 of *Contributions in Mathematical and Computational Sciences*, pages 273–281. Springer Berlin Heidelberg, 2013.

[Wu07] C. Wu. SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). http://cs.unc.edu/ ccwu/siftgpu, 2007.

[XN01] X. Xu and S. Negahdaripour. Application of extended covariance intersection principle for mosaic-based optical positioning and navigation of underwatervehicle. In *ICRA'01*, pages 2759–2766, 2001.

[YFK07] A. Yamashita, M. Fujii, and T. Kaneko. Color registration of underwater images for underwater sensing with considera-

tion of light attenuation. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4570–4575, 2007.

[YFK08] A. Yamashita, A. Fujii, and T. Kaneko. Three dimensional measurement of objects in liquid and estimation of refractive index of liquid by using images of water surface with a stereo vision system. In *ICRA*, pages 974–979, 2008.

[YHKK03] A. Yamashita, E. Hayashimoto, T. Kaneko, and Y. Kawata. 3-D measurement of objects in a cylindrical glass water tank with a laser range finder. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1578–1583, 2003.

[Zha99] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the International Conference on Computer Vision*, pages 666–673, Corfu, Greece, 1999.