ANALYSIS OF UNCERTAINTY IN UNDERWATER MULTIVIEW RECONSTRUCTION

BY

Igor Kozlov

BS in Information Systems, Peter the Great St.Petersburg Polytechnic University, 2015

THESIS

Submitted to the University of New Hampshire

in Partial Fulfillment of

the Requirements for the Degree of

Master of Science In Computer Science

September, 2018

This thesis has been examined and approved in partial fulfillment of the requirements for the degree of M.S. in Computer Science by:

Thesis Director, Philip J. Hatcher, Professor, Computer Science

R. Daniel Bergeron, Professor Emeritus, Computer Science

Yuri Rzhanov, Research Professor, Ocean Engineering

On August 27, 2018

Original approval signatures are on file with the University of New Hampshire Graduate School.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to the members of my thesis committee Prof. Philip J. Hatcher, Prof. R. Daniel Bergeron and Prof. Yuri Rzhanov for all the help and guidance they have provided. Their support, encouragement and insight made this thesis possible.

I would also also like to thank Paul Lavoie for his assistance with constructing a multi-camera rig for our experiments.

I also would like to express my thanks to the Department of Computer Science and the Center for Coastal and Ocean Mapping at the University of New Hampshire for creating an environment that allowed me to learn many new skills and work on my research.

Last but not the least, I would like to thank my friends and family. They have always been there for me, providing their encouragement and support when I needed it the most.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

ANALYSIS OF UNCERTAINTY IN UNDERWATER MULTIVIEW RECONSTRUCTION

by

Igor Kozlov

University of New Hampshire, September, 2018

Multiview reconstruction, a method for creating 3D models from multiple images from different views, has been a popular topic of research in the field of computer vision in the last two decades. Increased availability of high-quality cameras led to the development of advanced techniques and algorithms. However, little attention has been paid to multiview reconstruction in underwater conditions. Researchers in a wide variety of fields (e.g. marine biology, archaeology, and geology) could benefit from having 3D models of seafloor and underwater objects. Cameras, designed to operate in air, must be put in protective housings to work underwater. This affects the image formation process. The largest source of underwater image distortion results from refraction of light, which occurs when light rays travel through boundaries between media with different refractive indices. This study addresses methods for accounting for light refraction when using a static rig with multiple cameras. We define a set of procedures to achieve optimal underwater reconstruction results, and we analyze the expected quality of the 3D models' measurements.

CHAPTER 1
INTRODUCTION

1.1 Motivation

It is estimated that 71% of Earth is covered by water. However, despite all the efforts of mankind, a large portion of it remains almost unexplored using modern high resolution methods. In most regions of the ocean, the water pressure at the seafloor makes it extremely difficult to operate for humans, further slowing down the process of ocean exploration.

Inexpensive 3D reconstruction of the undersea objects would be extremely valuable for groundtruthing of underwater video and acoustic data. 3D information allows for the extraction of spatial spectra (i.e., seafloor rugosity, which is considered to be one of the most important characteristics of marine habitats), and shapes of larger species, such as scallops, starfish, etc. Quantitatively accurate reconstruction requires careful calibration of individual cameras and a multi-camera rig. Underwater imaging adds another level of complexity due to refractive effects on the interfaces between media with different refraction indices. Typical cameras used for underwater imaging are designed for air, and are encased in waterproof housings, which causes the light rays that carry information about scenery to be bent by refraction. This is a nonlinear process governed by Snell's Law that renders the standard pinhole camera model invalid. Treibitz *et al.* [1] show that optical systems with a flat refractive interface do not have a single viewpoint. Instead, different rays behave as if the camera has different focal lengths; i.e., is varifocal. Thus, ignoring refraction [3] or attempting to compensate for refractive effects by standard calibration conducted underwater [4] leads to erroneous 3D reconstruction. Quantitatively accurate reconstruction is possible only when

parameters related to refractive effects are estimated and taken into account during the reconstruction process.

1.2 Purpose of this Study

A multi-camera system can be used to create a 3D reconstruction from a set of images captured simultaneously by these cameras. However, it requires intrinsic, extrinsic and refractive calibrations to be performed in advance for each of the cameras. In each calibration there is an uncertainty in estimating each parameter. This study investigates the uncertainty for measurements of these parameters as well as the ways to improve the accuracy of calibration.

CHAPTER 2

ALGEBRAIC GEOMETRY MODELS IN COMPUTER VISION

2.1 Pinhole Camera Model

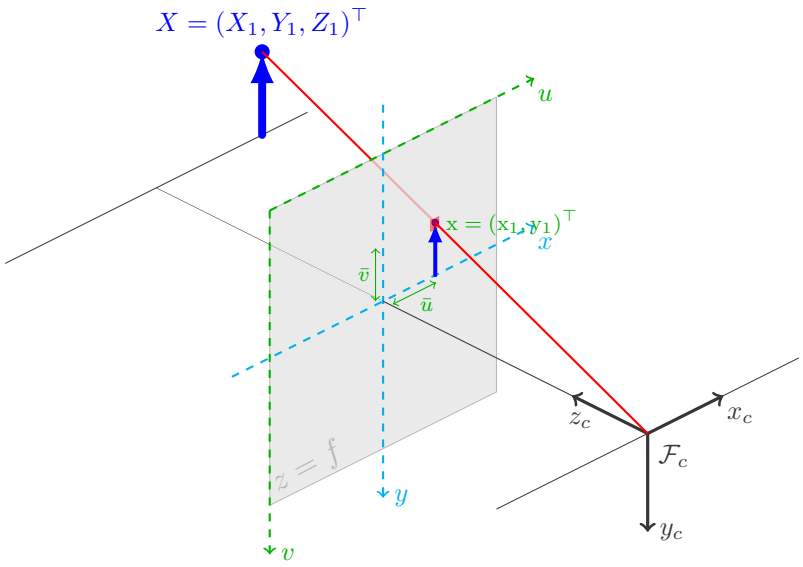

Figure 2.1: Pinhole camera model

We use Figure 2.1 to describe the pinhole camera model. Let the center of projection F_c be the center of an orthogonal coordinate system. The image plane is located at $Z = f$. The line to any point $X = (X_1, Y_1, Z_1)^\top$ to the camera center F_c intersects the camera's image plane at the projection point $\mathrm{x} = (\mathrm{x}_1, \mathrm{y}_1)^\top$. The line from the camera center perpendicular to the image plane is called principal axis of the camera or the camera axis. Equation 2.1 describes this transformation.

$$\mathrm{x} = KX \tag{2.1}$$

where K is called a camera calibration matrix (2.2).

$$K = \begin{bmatrix} f & s & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

f represents focal length, s is the skew parameter, and x_0, y_0 is the principal point. The skew parameter for most digital cameras is zero, and in this thesis is assumed to be zero as well. The principal point x_0, y_0 describes where the principal axis of the camera intersects the image plane as shown in Figure 2.2.

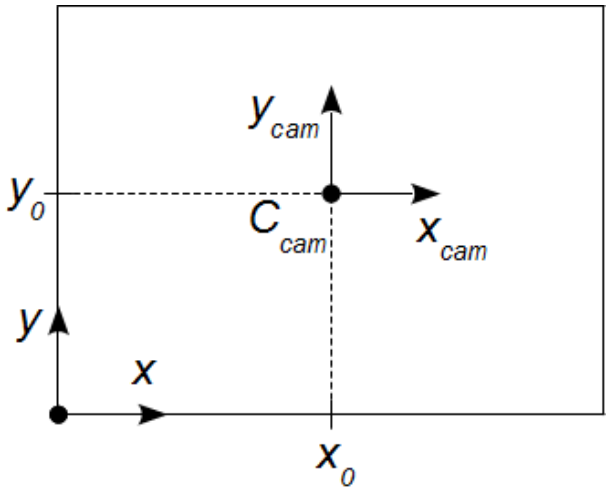

Figure 2.2: Principal point

Transformation between two coordinate system can be described by Equation 2.3

$$X_{cs1} = \begin{bmatrix} R & -RC \\ 0 & 1 \end{bmatrix} X_{cs2} \tag{2.3}$$

where X_{cs1} and X_{cs2} are coordinates of point X in two different coordinate systems, R is a $[3 \times 3]$ rotation matrix and C is the column vector coordinates of the origin of the first coordinate system in the second coordinate system. Equation 2.1 assumes that the camera center is located at the origin of the coordinate system. To extend it to a different coordinate

system Equation 2.4 is used.

$$\mathrm{x} = \begin{bmatrix} f & s & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & | -RC \end{bmatrix} X = PX \tag{2.4}$$

where P is a projection matrix.

2.2 Lens Distortion

Most modern cameras have a single lens or a system of lenses. The nature of how light rays refract passing through the lens causes the image to be distorted. It is most noticeable, if the scene contains objects that have straight line patterns. Due to lens distortion these lines appear curved on an image. In order to apply the pinhole camera model to an image acquired by such a camera, images need to be corrected for the distortion caused by lenses.

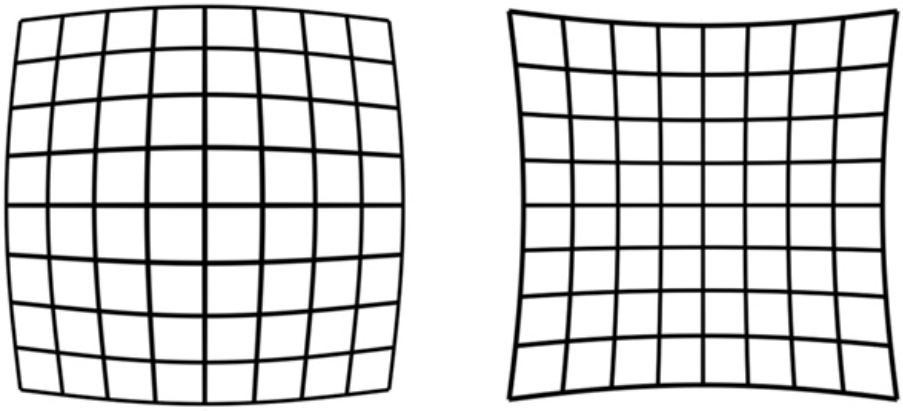

Figure 2.3: Radial lens distortion: barrel(left), pincushion(right)

The Brown-Conrady model is the most commonly used distortion model. Using Equations 2.5 and 2.6 we can transform a camera image to match one taken by a pinhole camera. Brown classifies lens distortion into two types: radial and tangential. There are two simple types of radial distortion: barrel and pincushion as shown on figure 2.3. Tangential distortion

is usually cause by minor misalignments of the optical elements in the camera.

$$\mathrm{x_u} = (1 + \mathrm{R_1 r^2} + \mathrm{R_2 r^4} + \mathrm{R_3 r^6})\mathrm{x_d} + 2\mathrm{T_1 x_d y_d} + \mathrm{T_2}(\mathrm{r^2} + 2\mathrm{x_d^2}) \tag{2.5}$$

$$\mathrm{y_u} = (1 + \mathrm{R_1 r^2} + \mathrm{R_2 r^4} + \mathrm{R_3 r^6})\mathrm{x_d} + 2\mathrm{T_2 x_d y_d} + \mathrm{T_1}(\mathrm{r^2} + 2\mathrm{y_d^2}) \tag{2.6}$$

where R_1, R_2, R_3 are radial distortion coefficients, T_1, T_2 are tangential distortion coefficients, $\mathrm{x_u}$, $\mathrm{y_u}$ and $\mathrm{x_d}$, $\mathrm{y_d}$ are coordinates of the undistorted image point and distorted image point on an image plane accordingly, and r is the distance from the distorted image point to the principal point (Equation 2.7).

$$\mathrm{r} = \sqrt{(x_d - x_0)^2 + (y_d - y_0)^2} \tag{2.7}$$

2.3 Camera Calibration

Camera calibration is the process of estimating parameters discussed in Sections 2.1 and 2.2. There have been multiple efforts to develop a robust algorithm for calibrating cameras [5–8]. However, in modern computer vision, Zhang's method [5] is the most widely used. This method requires a planar calibration object with easily detectable features, and a known geometry of those features, in contrast to many of the previous methods that require more complicated calibration object or previous knowledge about the geometry of the scene. The chessboard pattern is often used for these purposes.

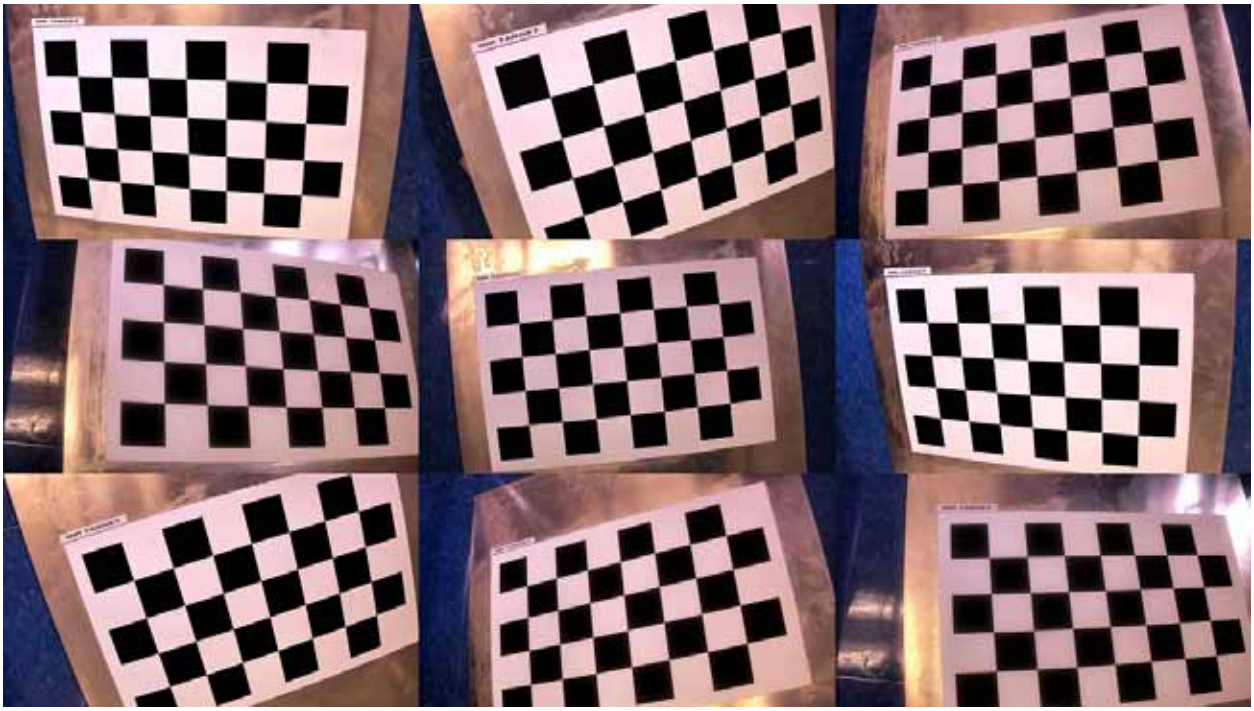

Figure 2.4: Calibration images

Zhang's method requires multiple images of the calibration object to be taken at different poses, by either moving the calibration object or the camera itself as shown on Figure 2.4. The projections of feature points are then detected in each image. In the final step calibration software is used, wgich consists of a closed-form solution step and a nonlinear refinement step to obtain the estimates of the cameras' parameters.

CHAPTER 3
REFRACTION

Whenever we look at an object submerged in water from above the water level, that object appears to be closer to us than it really is. This effect happens due to a phenomenon called refraction. Refraction occurs when the light ray passes through an interface between two materials in which light travels at different speeds. At that point the light wave velocity changes, but the frequency stays the same. Snell's law (Equation 3.1), also known as law of refraction, describes the relationship between the angles of incidence of the light ray before and after passing the the interface between two media.

$$\mu_1 \sin\theta_1 = \mu_2 \sin\theta_2 \tag{3.1}$$

where μ_1, μ_2 are the refractive indices and θ_1 and θ_2 are incidence angles. Figure 3.1 shows how the light ray changes its direction when passing through the interface between two media with different refractive indices ($\mu_1 < \mu_2$).

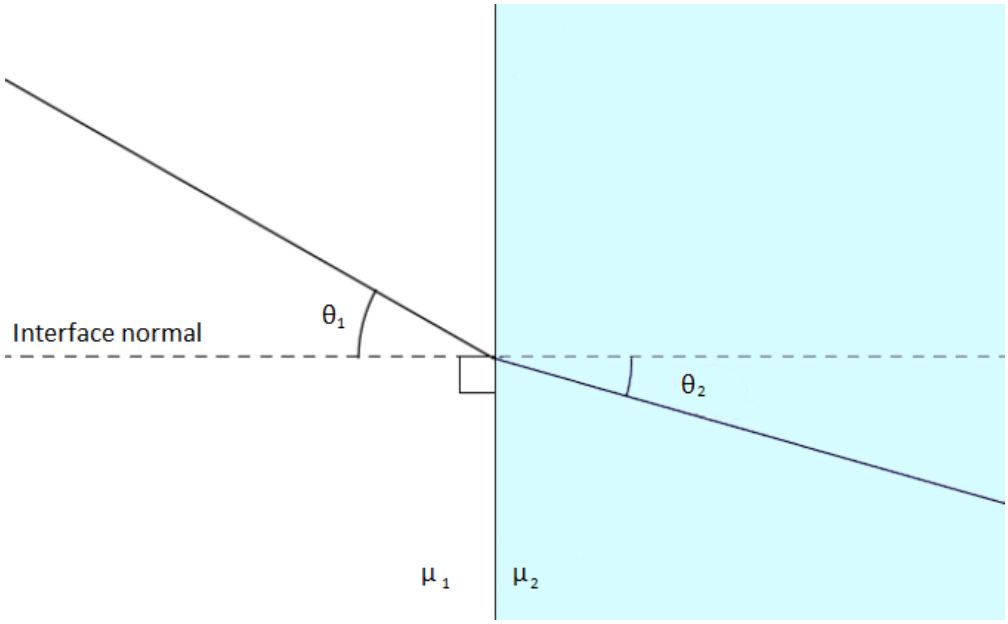

Figure 3.1: Snell's law

From Snell's law we can see that only light rays that are perpendicular to the interface between two media do not change their direction.

3.1 Flat Port Housing Geometry

A typical flat port housing consists of a waterproof case with a flat transparent material (e.g. glass) for the camera to look through. The camera is installed inside the housing, so that it faces the glass and the camera axis is nearly perpendicular to the glass surface. Inside the housing there is either air or vacuum. In this case, when the camera is submerged in water, light rays travel through two parallel interfaces — one between water on the outside of the housing and glass, and one between glass and air inside the housing. Figure 3.2 shows the refractive geometry of a flat port housing. The distance to the interface d_0 is the shortest distance along the interface normal between the camera center and the glass surface, and d_1 is the thickness of the glass medium. Snell's law is symmetric, meaning that the light ray traveling from the camera center to the object point has the same path as the light ray traveling in the opposite direction. In reality, the camera sensor captures the light ray

reflected from the object point, but it is more convenient to observe the ray in the direction from the camera center to the object point. Rays v_0, v_1, v_2 show the path of the light ray that enters the camera center after being reflected from the object point. We know that refraction occurs when the light ray passes through both air-glass and glass-water interfaces. Hence, Snell's law can be applied to both refractions:

$$\mu_0 \sin\theta_0 = \mu_1 \sin\theta_1 = \mu_2 \sin\theta_2 \tag{3.2}$$

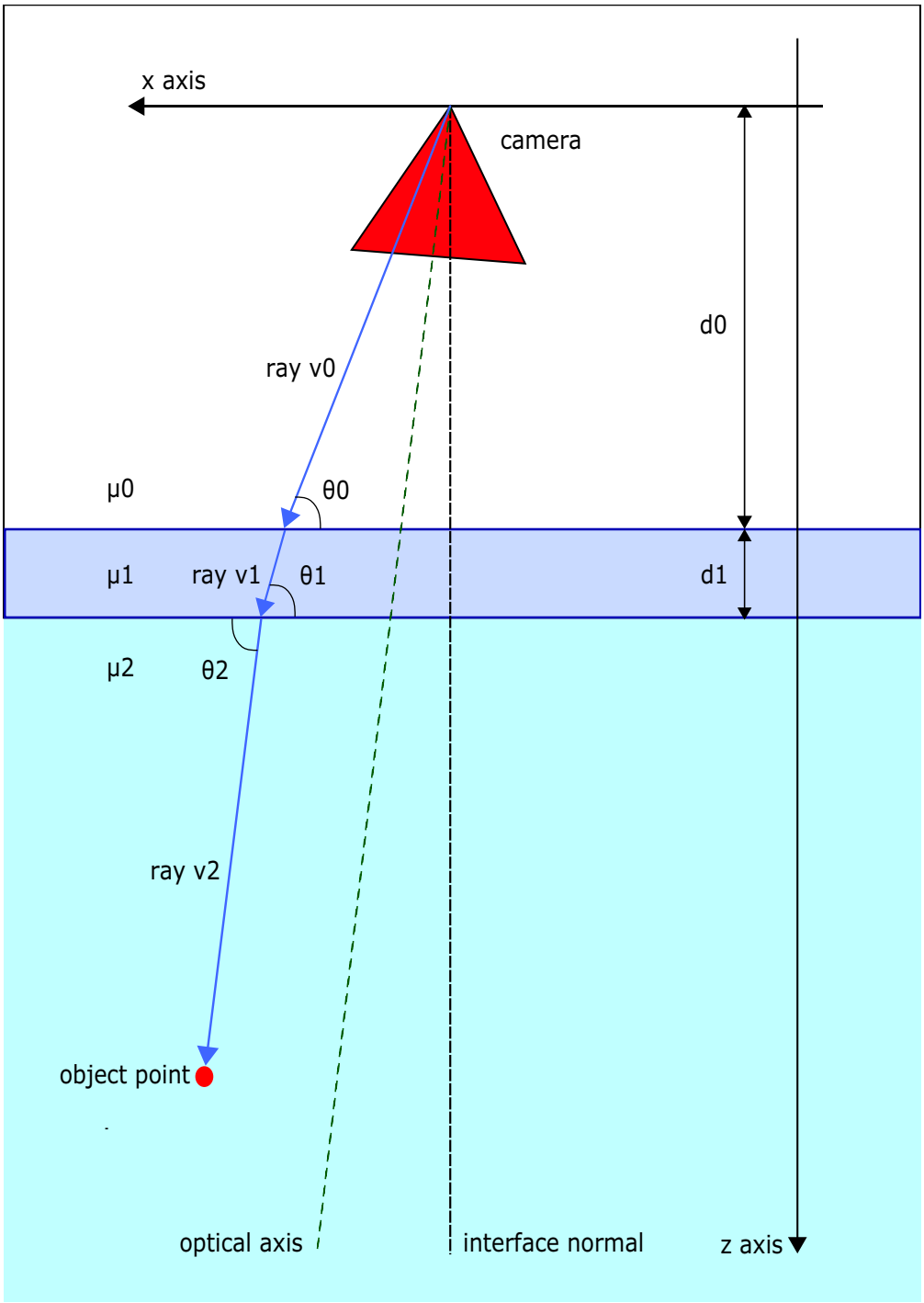

Figure 3.2: Flat port housing geometry

Agrawal *et al.* [9] shows that the flat port housing refraction system is an axial camera system, where the axis is defined as the line parallel to the interface normal and passing through the camera center. We refer to this axis as the interface normal axis. This means

that every segment of the light ray path and the interface normal axis lie in a single plane π called the plane of refraction (POR).

3.1.1 Forward Projection

The process of projecting a point in the world onto the image plane of the camera is called forward projection. Agrawal *et al.* [9] observe three cases of refractive environments. Case 1 is a single refraction case ($\mu_1 \neq \mu_2$). In case 2 there are two refractions, but the refractive index of the first layer is the same as the third layer($\mu_1 \neq \mu_2$, $\mu_1 = \mu_3$), which corresponds to a flat port housing system in air. And in case 3 there are three layers, each with a different refractive index, which corresponds to a flat port housing system in water. Agrawal *et al.* [9] derive a polynomial equation for each case. For cases 1 and 2, the equations are 4^{th} degree, and for case 3 it is a 12^{th} degree equation. Yau *et al.* also propose an optimization solution for the forward projection [10].

3.1.2 Backward Projection

The process of computing the directions of each of the segments of the refracted ray is called backward projection. Finding the geometry of a ray is a typical problem for ray tracing. The direction of the ray from the camera to the first refractive interface can be acquired using the camera calibration matrix using Equation 2.1. A recursive algorithm is then applied to find the direction of each segment, using Equation 3.3 [11].

$$v_{i+1} = a_{i+1} v_i + b_{i+1} \mathrm{n} \tag{3.3}$$

where

$$a_{i+1} = \mu_i / \mu_{i+1}, \tag{3.4}$$

$$b_{i+1} = \frac{-\mu_i v_i \mathrm{n} - \sqrt{\mu_i^2 (v_i n)^2 - (\mu_i^2 - \mu_{i+1}^2) v_i \cdot v_i}}{\mu_{i+1}}. \tag{3.5}$$

CHAPTER 4
REFRACTIVE CALIBRATION

In the case of a flat port interface, the refractive parameters consist of:

- Refractive indices of air, glass and water
- Distance from the camera viewpoint to the glass interface
- Glass layer thickness
- Normal to the plane of the glass interface

Refractive calibration is aimed at estimating these parameters. Various research focused on calibration for one or more of the refractive parameters. The refractive indices are usually assumed to be known. Treibitz *et al.* [1] suggest a calibration method, in which the camera is oriented normally and the glass thickness is negligible. Gedge *et al.* [12] and Chen *et al.* [13] use stereo matching for calibration. Yau *et al.* [10] use a calibration target that emits disparate wavelengths of light. Agrawal *et al.* [9] propose an analytical solution using an axial camera model. Most algorithms have a nonlinear refinement procedure to minimize either reprojection error or 3D error of the calibration object features.

4.1 Pinhole Camera Model Approximation

When a camera is submerged in water the field of view decreases. Straight lines become curved, which is similar to the effect of pincushion radial distortion. Many researchers consider that refraction can be taken into account by using the pinhole camera model with modified field of view and distortion coefficients [14–18].

The pinhole camera model assumes that the camera has a single view point (SVP). Treibitz *et al.* [1] show, however, that a camera placed in a flat port housing corresponds to a non-SVP camera model as shown on Figure 4.1.

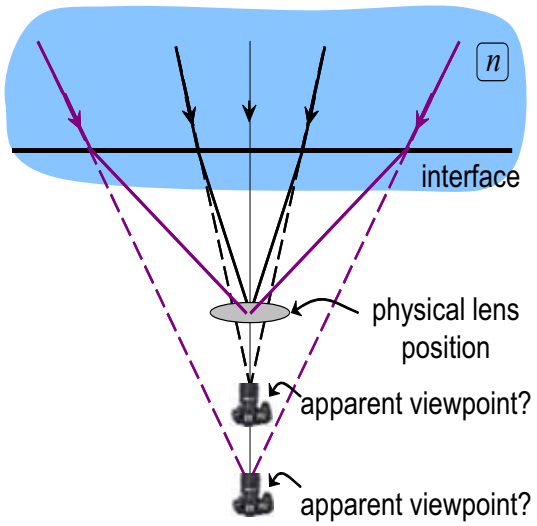

Figure 4.1: Non-SVP nature of flat refraction model [1]

Dolereit *et al.* [19] describe a method of converting an image taken underwater to an image as if it was taken in air. This approach works if the object captured by the camera is flat. In most situations, however, 3D reconstruction is performed for objects with a complex structure. In this case, the "look behind the corner" phenomenon can be observed as shown on Figure 6.2 in which point X_1 is visible if the image is taken underwater, but is not visible when taken in air..

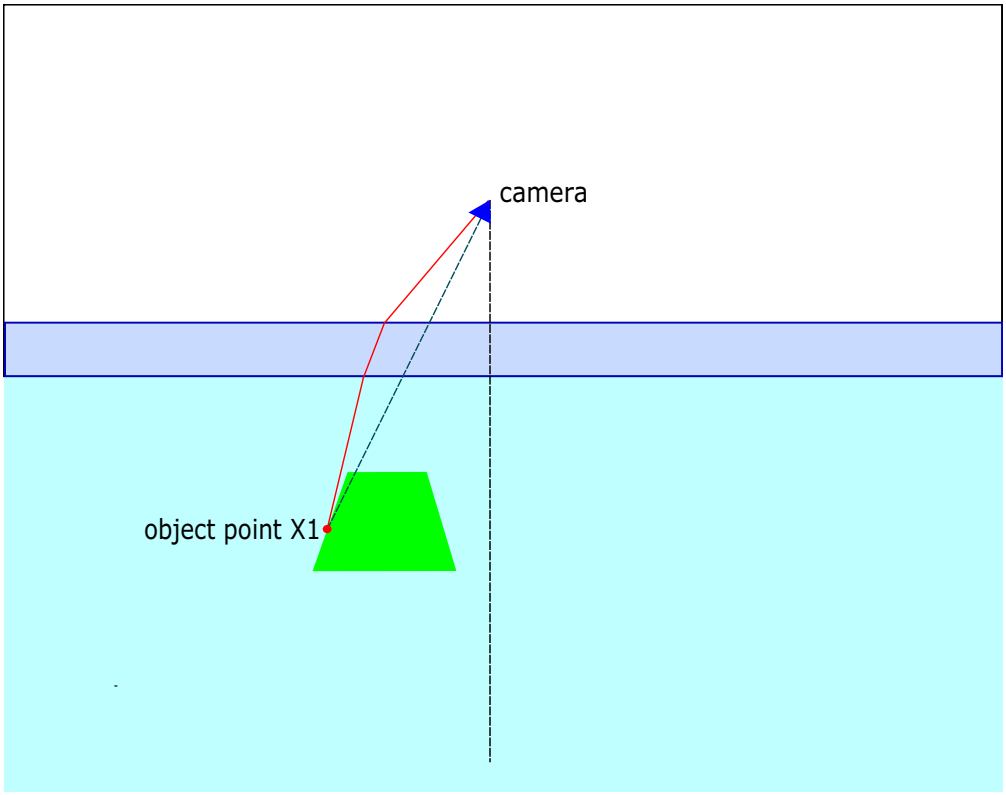

Figure 4.2: "Look behind the corner" phenomenon

Point X_1 will be visible if the image is taken underwater, however if the image was captured in air it would be occluded.

4.2 Calibration Method by Agrawal *et al.*

One of the most comprehensive existing calibration methods was developed by Agrawal *et al.* [9], because it allows the estimation of almost all of the refractive parameters.

$$vp_2 \times \begin{bmatrix} vp_0/c_0 & vp_1/c_1 & z_p \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ \alpha \end{bmatrix} = -vp_0 \times u \tag{4.1}$$

where vp_0, vp_1, vp_2 are segments of the refracted ray in air, glass and water respectively, $z_p = [0; 1]$ is a unit vector, $c_n = vp_n^\top z1$, d_0 is the distance to the glass, d_1 is the thickness of the glass, α is the translation magnitude along the refractive axis and $u = [z_2^\top P_c, z_1^\top P_c]$ is

the projection of the feature point P_c on POR. In the end a nonlinear optimization is used to improve the accuracy of estimated parameters. The optimization algorithm minimizes reprojection error for each detected feature as shown in Equation 4.2.

$$J(R, T, d_0, d_1, n) = \sqrt{\frac{1}{K}\sum_{i=1}^{K}(p(i) - \hat{p}(i))^2} \tag{4.2}$$

4.3 Calibration Method by Traffelet *et al.*

Traffelet *et al.* [20] proposes a calibration method that is based on acquiring multiple images of a calibration object with different camera poses for each image. This approach is similar to intrinsic camera calibration method by Zhang [5]. Nonlinear optimization is performed for a sequence of images. Camera pose is estimated for each image, while housing parameters optimized for all images simultaneously.

4.4 Multi-Camera Rig Calibration Method

The method proposed by Agrwal *et al.* [9] gives an accurate estimate of the normal to the interface. However, the error in estimating the distance to the interface can be quite significant. We propose a new calibration technique for a multi-camera system with fixed poses of the cameras with respect to each other. Our calibration method extends the work presented in [9].

4.4.1 Glass thickness

Most flat port underwater housings have a nonzero distance from camera center to glass, as well as a nonzero glass thickness. However, in most situations the thickness of glass is known from the housing specifications, therefore it does not need to be estimated. The equation 4.1 can be changed to equation 4.3, where d_0 and α need to be determined.

$$vp_2 \times \begin{bmatrix} vp_0/c0 & z_p \end{bmatrix} \begin{bmatrix} d_0 \\ \alpha \end{bmatrix} = -vp_0 \times u - vp_2 \times (vp_1/c1)d_1 \tag{4.3}$$

4.4.2 Distance to the interface

The method described in [9] gives an exact solution in the absence of noise in the detected calibration object features. However, even as little as 0.1 pixel Gaussian noise leads to significant errors in estimating the distance to the interface. We conducted a simulation experiment to explore how the error in distance to the interface affects the reprojection error. In the optimization step of the refractive calibration the distance to the interface parameter is fixed for a given realization, and incremented by 1 mm for the next realization. All other parameters are optimized. In the experiment, the ground truth distance to the interface is $d_0 = 100$ mm, and the calibration object is located 800 mm away from the camera normally oriented to the optical axis. The location of detected feature projections is known.

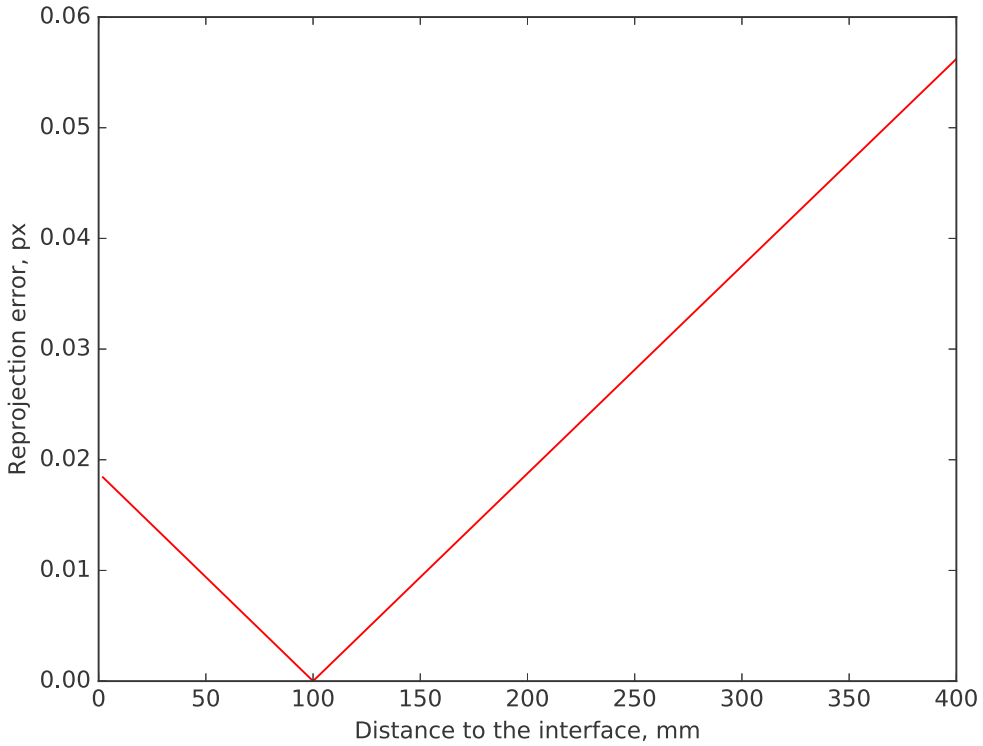

Figure 4.3: Reprojection error due to incorrectly estimated distance to the glass

Figure 4.3 shows the results of the simulation. It is obvious that even when the distance to the interface is estimated with error of 300 mm, the reprojection error is less than 0.06 pixel, which is significantly smaller than the noise usually present in the detected feature projections from real images. It can be observed that when the distance to the interface increases, the estimated distance to the calibration object decreases, as seen in Figure 4.4

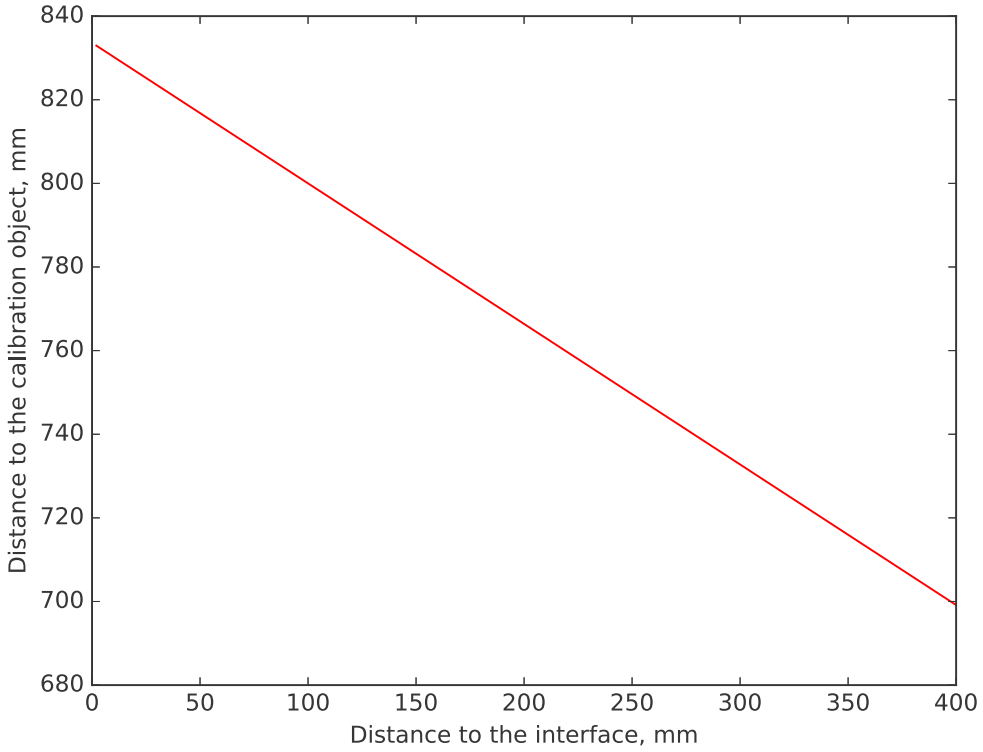

Figure 4.4: Optimal distance to the calibration object for incorrectly estimated distance to the glass

It has been shown in [9] that in the air/layer/air scenario the distance from the camera center to first refractive interface cannot be determined because in this case feature projections are not affected by this parameter. We have found that in water the refractive-related distortions due to change of this distance can be compensated by the change of the distance to the calibration object. Thus, even moderate noise in measurement of features' projections is stronger than the difference between refractive distortions related to these two factors and may lead to significant errors in determination of refractive parameters. This, however, is

true for a single camera. For a multi-camera rig with verged cameras, extrinsic parameters can be accurately estimated in air, where the viewport layer leads only to relatively small parallel shifts of all light rays. These estimates act as additional constraints in an overall optimization procedure by restricting distance to the calibration object, and hence the distance to the refractive interface as well. It is worth noting that cameras without vergence do not have the additional constraints and noise in images reduces the accuracy of determined refractive parameters in the same way as for a single camera.

4.4.3 Calibration procedure

The following steps must be taken to calibrate a camera rig:

1. Acquire a set of images of a chessboard pattern calibration target in air in different poses. The images have to be captured simultaneously by all cameras.

2. Repeat the previous step with underwater images.

3. Obtain the estimates of the unknown parameters, using the method described in [9].

4. Perform an overall nonlinear optimization

For each set of images from all cameras, the relative poses of the cameras are the same, and thus for each pose of the calibration object we only need to estimate the pose of one camera with respect to the target. Our method can be used in combination with the method described by Traffelet [20] by taking multiple sets of images both in air and underwater to further improve the accuracy. The function that minimizes by nonlinear optimization is shown in Equation 4.4.

$$J(W_{cam(1..m-1)}, W_{pose(1..k)}, d_{cam(1..m)}, n_{cam(1..m)}) = \frac{1}{K}\sum_{i=1}^{K}(p(i) - \hat{p}(i))^2 \tag{4.4}$$

where W_{cam} is the position of the cameras (R, T) in relation to the camera that is assigned to be the main camera, W_{pose} is the position of the main camera in relation to the calibration

object, d_{cam} is the distance to the interface for each camera, n is a normal unit vector, m is the number of cameras, k is the number of poses, p is the detected feature point and $\hat{p}$ is the point acquired from the forward projection, K is the number of feature points.

$$K = s_x s_y m n \tag{4.5}$$

where s_x, s_y denote the number of squares on a chessboard pattern calibration object, horizontally and vertically respectively. To initialize the optimization, the estimates acquired using [9] are used. In our implementation of the calibration method we use the Levenberg–Marquardt nonlinear optimization algorithm, implemented by Lourakis *et al.* [21].

4.5 Single Camera Calibration Method

In the multi camera rig calibration method the relative poses of the cameras with respect to each other act as an additional constraint. It is also possible to add an additional constraint for the case of a single camera. In this case, the calibration object must be fixed with respect to the camera. Two images are required: one taken in air and one taken underwater. Thus, it is only needed to determine one camera pose in respect to the calibration object.

$$J(W_{pose}, d_{cam}, n_{cam}) = \frac{1}{K} \sum_{i=1}^{K} (p(i) - \hat{p}(i))^2 \tag{4.6}$$

$$K = 2 s_x s_y \tag{4.7}$$

4.5.1 Refractive Principal Point

Using the fixed target calibration method it is possible to obtain the normal to the interface in a much simpler way than proposed by Agrawal *et al.* [9]. Figure 4.5 shows an example of images acquired by using the fixed target calibration method. First, all the points on each image are bijectively matched and a line is drawn through each pair of points as shown on Figure 4.6. Note that projections of any feature on two images taken by the same camera are

in a plane of refraction and thus lie on a line that also passes through the point where the ray from the focal point and the normal to the interface intersects the retinal plane. Second, we compute the intersections of all resulting lines.

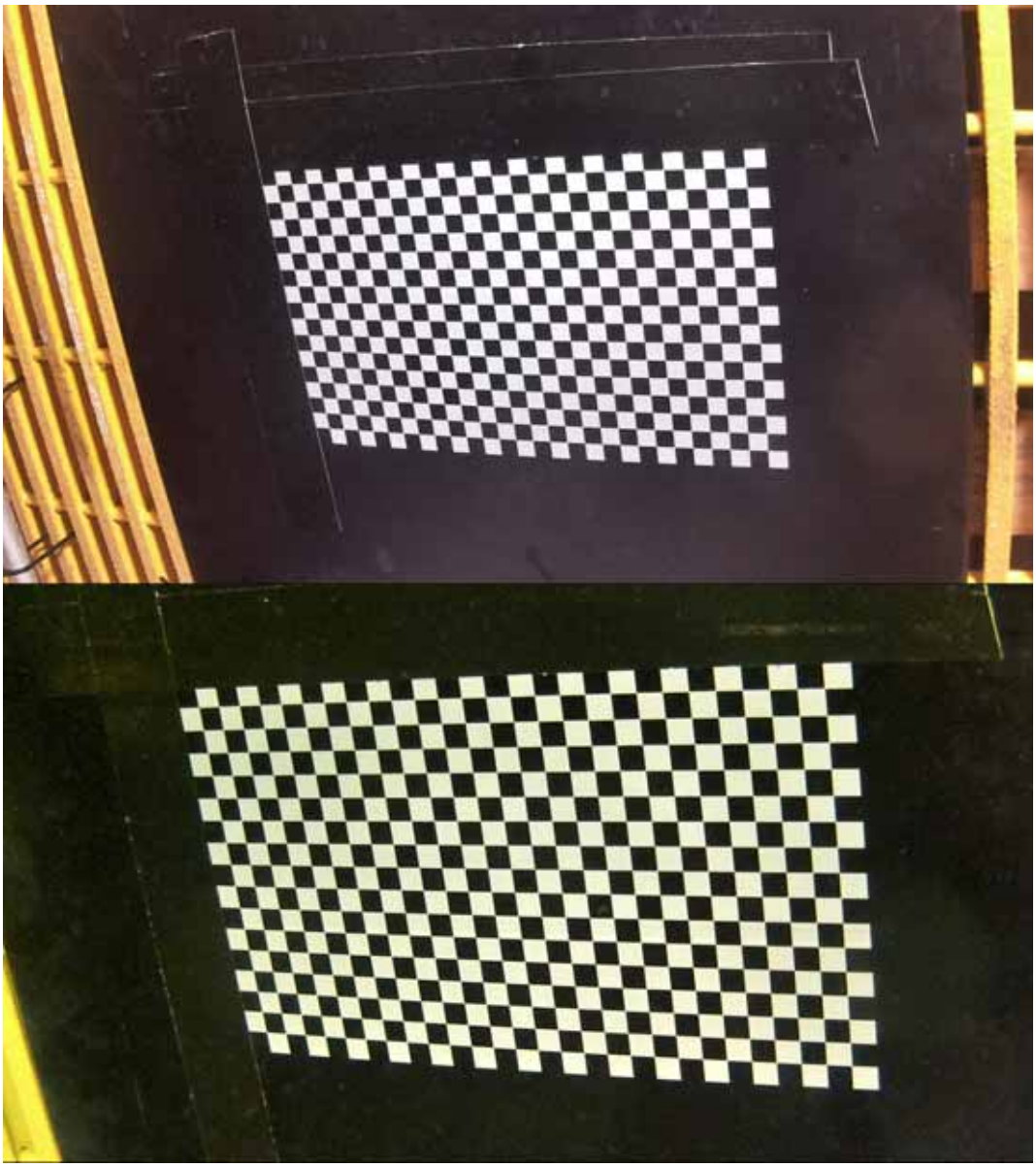

Figure 4.5: Refractive principal point

The intersections of the lines happen at a single point if no noise is present in the detected features. However, when noise is present, the line intersections result in a dense point cloud, from which a single point can be obtained using least squares method. This point is called

a refractive principal point (RPP) for convenience and it is, in fact, the point where the refractive optical axis passes through the image plane. Using the calibration matrix we can obtain the direction of the refractive optical axis, and hence, the normal to the refractive interface.

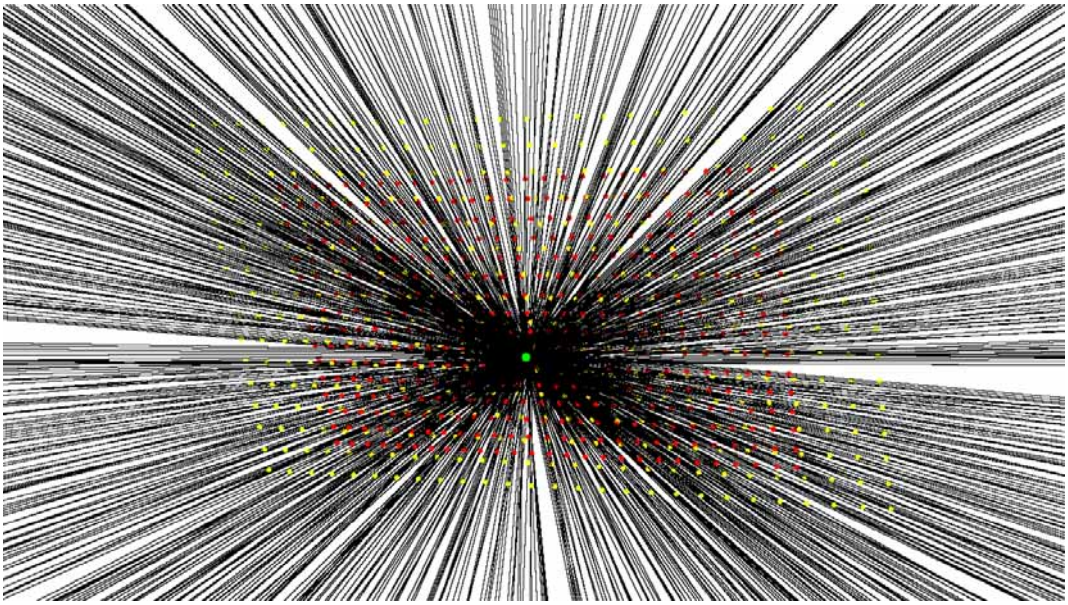

Figure 4.6: Refractive principal point

4.6 Simulation Results

In this section we show the comparison of synthetic calibration results. We compare our two methods of refractive calibration to the one proposed in [9]. For the multi-camera rig calibration method, three setups were simulated:

1. 2 cameras, 1 image in air, 1 image underwater for each camera
2. 5 cameras, 1 image in air, 1 image underwater for each camera
3. 5 cameras, 4 images in air, 4 images underwater for each camera

A particular scene for the calibration setup was chosen: the cameras have a resolution of 1280 by 720 pixels, focal length varies uniformly between 1540 and 1580, which gives a field of view of approximately 22 degrees; cameras are located such that one camera is placed

directly above the calibration object and the other cameras are placed in corners of a square with a side of 800 mm; the plane of the square is parallel to the plane of "the floor" where the calibration object is located; all cameras are pointed at a single orientation point on the floor that is directly below the central camera; the calibration object is a chessboard pattern with dimensions of 26 by 17 squares and the square size of 11.94 mm. The chessboard pattern is placed in close proximity with the orientation point with different orientation for each set of images, orientations vary with uniformly distributed rotations between 10 deg and −10 deg around X, Y, Z axes; and the translation also changes uniformly in a bounding box with a side of 150 mm. We ran 1000 simulations for different noise levels for feature point detection ranging from 0.0 pixels to 0.5 pixels with 0.1 pixel increment.

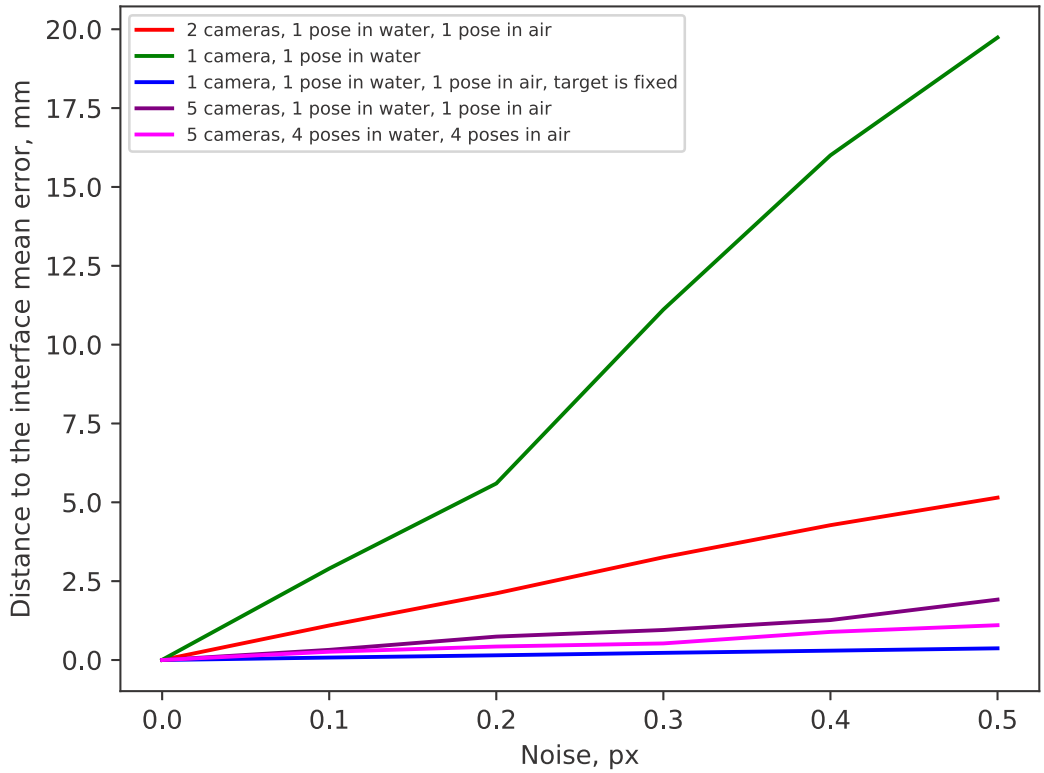

Figure 4.7: Average error in estimation of distance to the interface

Figures 4.7 and 4.8 show the results of the simulation for the estimation of the housing

parameters. It is noticeable, that even with only two cameras, the rig calibration method gives superior results in comparison to the method proposed in [9] (corresponds to 1 camera, 1 pose in water). The average error decreases when more cameras are used, as well as when more images for each camera are used. In case of a single camera calibration method with a fixed target the error is even smaller, and furthermore rises very slowly when more noise is present in the detected calibration features. Table 4.1 shows the comparison of error in estimation of the distance to the interface for 0.5 pixel noise level.

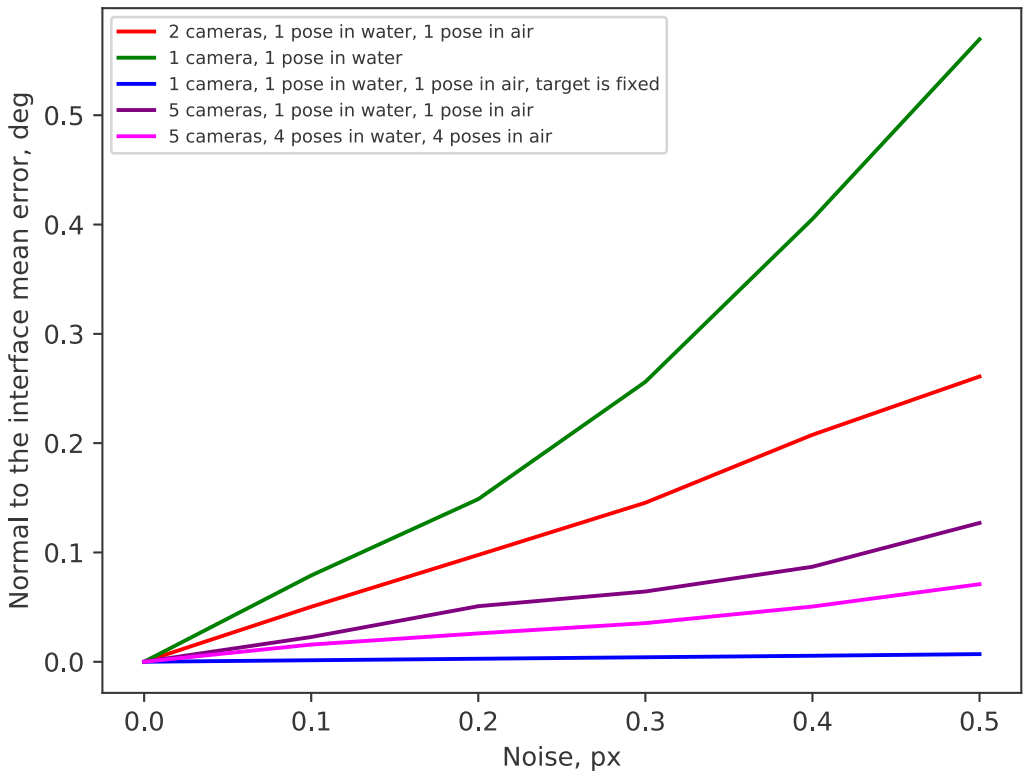

Figure 4.8: Average error in estimation of normal to the interface

Calibration method	d_{int} average error, mm	Accuracy improvement
$N_{cam} = 1, N_{water} = 1, N_{air} = 0$	13.7	
$N_{cam} = 2, N_{water} = 1, N_{air} = 1$	5.1	166%
$N_{cam} = 5, N_{water} = 1, N_{air} = 1$	1.5	792%
$N_{cam} = 5, N_{water} = 4, N_{air} = 4$	0.7	1783%
$N_{cam} = 1, N_{water} = 1, N_{air} = 1$, fixed	0.3	4085%

Table 4.1: Comparison of error in estimation of the distance to the interface for 0.5 pixel noise level

CHAPTER 5
3D RECONSTRUCTION

3D reconstruction is a process of extracting information about shape and appearance of an object. The methods of 3D reconstruction can be divided into two categories: active and passive. Active methods are based on using either mechanical or radiometric range finders. Such methods actively interfere with an object. In contrast, passive methods are non-invasive. Usually a camera sensor is used to measure the characteristics of light reflected from the object. There are multiple cues that can be used to extract 3D information from a 2D image. Monocular techniques are based on using a single image. They can use shade, silhouette, or texture as the cues. Multiview methods of 3D reconstruction use multiple images. They can be images produced either subsequently by the same camera, or by different cameras. This study focuses on multiview reconstruction from images capture by cameras that are statically fixed on a rig.

5.1 Multiview Reconstruction

Multiview reconstruction has been one of the most popular topics of research in computer vision for years. There are multiple applications of this technology: maps with 3D models of buildings and other objects, special effects in the movie industry, archaeology site virtual reality tours and many others. Figure 5.1 shows an example of 3D reconstruction. We will discuss some of the fundamental principles of multiview reconstruction.

Figure 5.1: Point cloud of a house model acquired from 3D reconstruction [2]

5.1.1 Triangulation

In a system where point X can be observed by two cameras the projections x_1, x_2 of this point appears on the image planes of both of the cameras as seen in Figure 5.2.

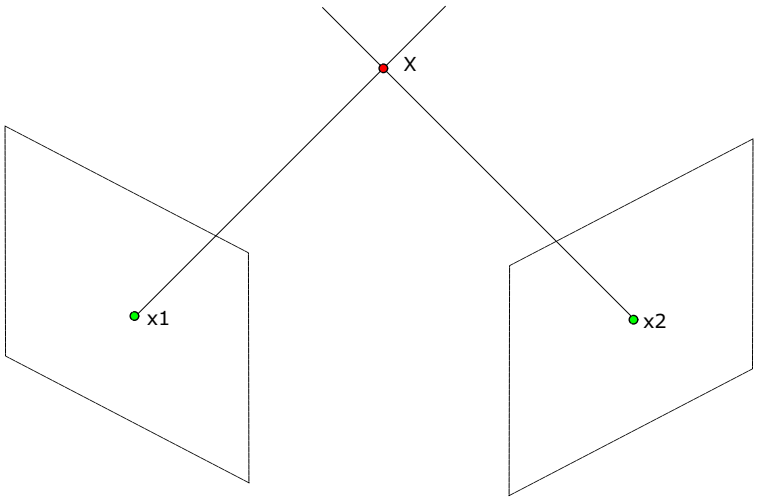

Figure 5.2: Triangulation

Hence, we can use Equation 2.4 to derive a constraint:

$$\begin{cases} \mathrm{x}_1 = \mathrm{P}_1\mathrm{X} \\ \mathrm{x}_2 = \mathrm{P}_2\mathrm{X} \end{cases} \tag{5.1}$$

where P_1 and P_2 are the projection matrices of the cameras. Given the exact $\mathrm{x}_1, \mathrm{x}_2, \mathrm{P}_1, \mathrm{P}_2$ it is possible to find the exact coordinates of point X. However, in real life all parameter measurements have an error, which causes the lines (L_1, L_2) connecting detected feature points $\mathrm{x}'_1, \mathrm{x}'_2$ and camera centers to not intersect. In this case optimal triangulation methods are used to estimate the location of point X in space. The linear transformation method suggests solving the resulting linear equation by obtaining the least squares solution. The mid-point point is based on minimizing the sum of squares of the Euclidean distances between X_{est} and the lines L_1, L_2 as shown in Equation 5.2.

$$J = d(L_1, X)^2 + d(L_2, X)^2 \tag{5.2}$$

Other more complex techniques were proposed, such as a polynomial method, minimizing the sum of the magnitudes of distances and others. Hartley and Strum [22] give an overview of comparative performances of the most used algorithms.

5.1.2 Epipolar Geometry

Consider a scene, where two cameras observe a point in space X as shown in Figure 5.3. The points e_L, e_R are the intersections between the lines connecting the camera optical centers and respective image planes of left and right cameras. These points are called the epipoles.

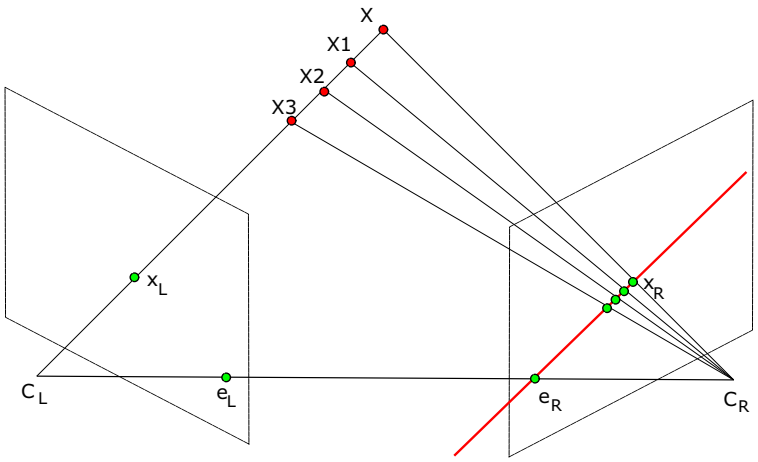

Figure 5.3: Epipolar geometry

A line $C_L X$ appears as a point on the image plane of the left camera. For the right camera, however, it appears as a line $e_R x_R$, which is called the epipolar line. In fact, all epipolar lines pass through the epipole e_R. This relation can be written in Equation 5.4.

$$x_R^\top F x_L = 0 \tag{5.3}$$

where F is called the fundamental matrix. The epipolar line corresponding to point x_L can be expressed as:

$$l_r = F x_L \tag{5.4}$$

5.1.3 Multiview Reconstruction Methods

Over the last twenty years, a variety of different methods and techniques for multiview reconstruction were developed. These methods differ in performance speed, accuracy and completeness of reconstruction and other parameters. Seitz *et al.* [23] provide a comparison for several algorithms. They categorized the existing methods by the scene representation, photoconsistency measure, visibility model, shape prior, reconstruction algorithm, and initialization requirements. They also provide a publicly available benchmark along with calibrated images and ground truth models captured by the Cyberware Model 15 laser stripe

scanner.

5.2 Underwater Multiview Reconstruction

Most algorithms for multiview reconstruction are designed to work for images taken in air. However, as discussed in Chapter 3, refraction has a big impact on image formation for underwater imagery. Hence, to perform an accurate multiview reconstruction of underwater objects, the refraction model has to be taken into account explicitly. One of the major differences with in air images is that straight lines appear curved on images taken by cameras submerged in water in flat port housings, and as the epipolar lines become epipolar curves [12, 24, 25].

In our work, we chose to use PMVS2 software [26] along with the modifications added by Yau *et al.* [10] to take refraction into account. Although the original PMVS2 was developed in 2010, it still compares well to the newer reconstruction methods. In PMVS2 salient feature patches are extracted from the images. Correlating patches from pairs of images are checked to satisfy epipolar geometry and triangulated to find their position in space, and then optimized to improve their orientation in space. The patches are then filtered to satisfy photometric discrepancy and neighborhood requirements. Yau *et al.* [10] expanded PMVS2 to explicitly incorporate a physically correct flat port housing refraction model. Housing parameters, estimated from refractive calibration were added as additional input parameters. The main changes were made in initial feature point matching, point triangulation, patch projection and sampling, and patch neighbor radius determination. Rays are triangulated using refractive forward projection. These modifications allow the acquisition of multiview reconstruction results for underwater imagery, similar to the ones from images captured in air.

CHAPTER 6

UNCERTAINTY IN UNDERWATER 3D RECONSTRUCTION

Before performing a 3D reconstruction from images taken underwater by a multi-camera rig the system has to be calibrated. A complete calibration consists of a series of steps, during which the camera intrinsics and extrinsic parameters, as well as the housing's refractive parameters have to be estimated. In each calibration step an error is introduced, and the error from each of the previous steps propagates to the subsequent ones. Because of those factors, finding an analytical solution to estimate the uncertainty in the 3D reconstruction can be very difficult.

Accuracy of 3D reconstruction depends not only on the hardware used for image acquisition, but also on the scene being imaged. Four major scene properties can be distinguished: spatial frequencies and contrast of a texture and 3D structure (or Digital Elevation Model) respectively. All of these are closely intertwined with the hardware capabilities. For example, high texture contrast is clearly important for reconstruction, but algorithms failing with 8-bit imagery may be successful with images having 16-bit per pixel per channel. Generally, substantial distance between cameras (baseline) makes reconstruction more accurate, but to resolve a deep narrow crevasse cameras should have a small baseline, and the imagery with low resolution may not resolve the crevasse at all. The great variety of possible scenarios and the impossibility of formulation an analytical model due to a complex image forming process does not allow for devising universal reconstruction optimization rules. This research resorts to numerical simulation and thus is applicable only to a limited set of imaged scenes and hardware configurations. However, the spatial frequencies and contrast levels are chosen

to be similar to those that are encountered in reality, and the tendencies that are reported are likely to hold for a wider set of conditions than the experiments were conducted for. More accurate estimates of uncertainties can be obtained by modeling specific hardware and typical characteristics of the 3D scene that needs to be reconstructed.

6.1 Monte Carlo Method

The Monte Carlo method is a computational technique that requires multiple random sampling in order to estimate the probability distribution of the possible results. This method is useful when an analytical solution of a problem is too complicated. Areas of application include calculating risks in business and financing, exploring the properties of physical and chemical processes, mathematical optimization, and others. In general, a Monte Carlo algorithm follows these steps:

1. Define parameters of a process that are random in nature
2. Define probability distributions of these parameters
3. Generate random parameter values from the distributions
4. Numerically process the values in accordance with the explored process or model
5. Process the output

6.2 Monte Carlo Simulation Framework

We developed a software framework that allows us to simulate each of the required calibration steps, along with a final step of creating a 3D model from synthetic images. Each realization of the simulation framework goes through the following steps:

1. Generate a set of ground truth parameters for each camera with uniform distribution for each camera in a given range.

- Intrinsic parameters
- Extrinsic parameters
- Refractive parameters

2. Generate a set of feature projections of a calibration object. Poses of the cameras in relation to the calibration object are uniformly distributed in a given range. Gaussian noise is added to simulate the error in feature detection.

3. Generate a set of feature projections of a calibration object, as if the cameras were enclosed in a protective housing and the images were taken in air and underwater. R, T of the cameras in relation to the calibration object are uniformly distributed in a given range. Gaussian noise is added to simulate the error in feature detection.

4. Obtain a set of estimated intrinsic parameters through intrinsic calibration

5. Perform refractive calibration, acquire a set of estimated refractive parameters and extrinsic parameters through refractive calibration.

6. Generate a set of images of a 3D model

7. Perform a 3D reconstruction using the estimated parameters from previous steps.

6.2.1 Embree Ray Tracer

For generating images that are affected by refraction, the Embree ray tracing technology developed by Intel [27] was used. The Intel Embree kernel supports assigning a material to any surface, with a variety of properties. To simulate a camera placed in a flat port housing, two square polygons were placed in front of the virtual cameras. The polygons were assigned a material that is transparent, but causes the light rays to be refracted which corresponds to a physical process, happening in a real life situation, when a camera captures an image underwater.

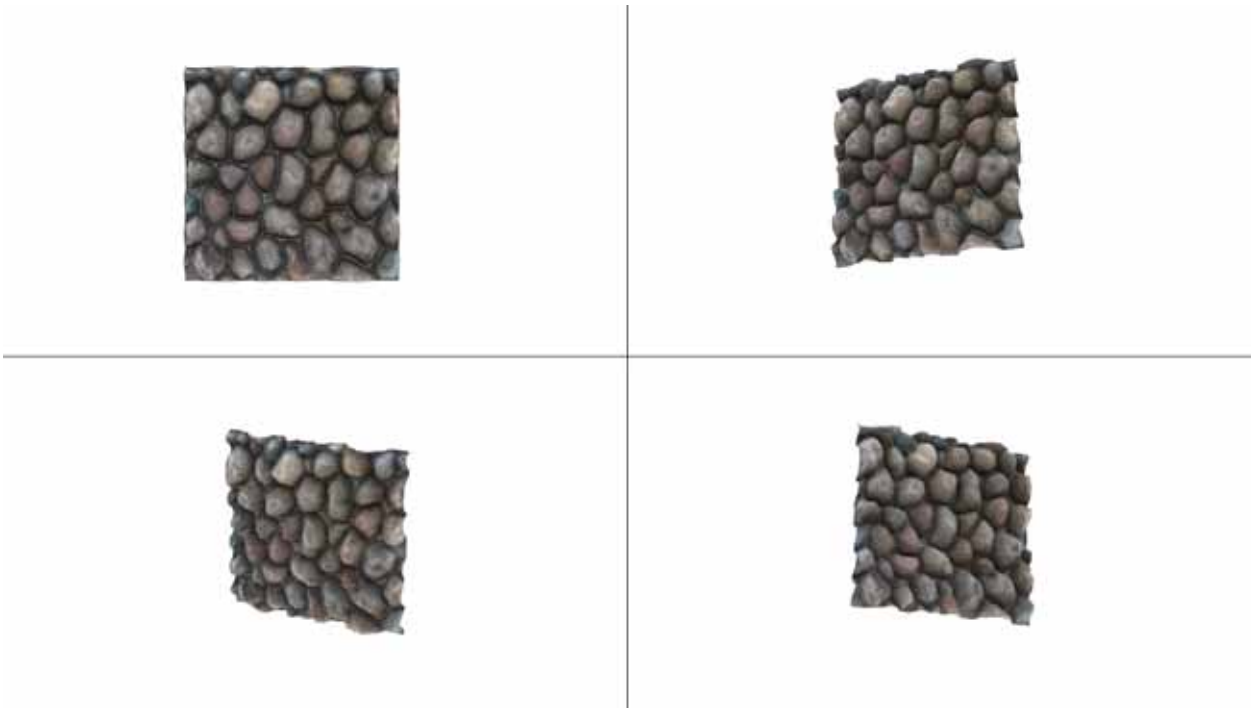

Figure 6.1: Intel Embree generated images

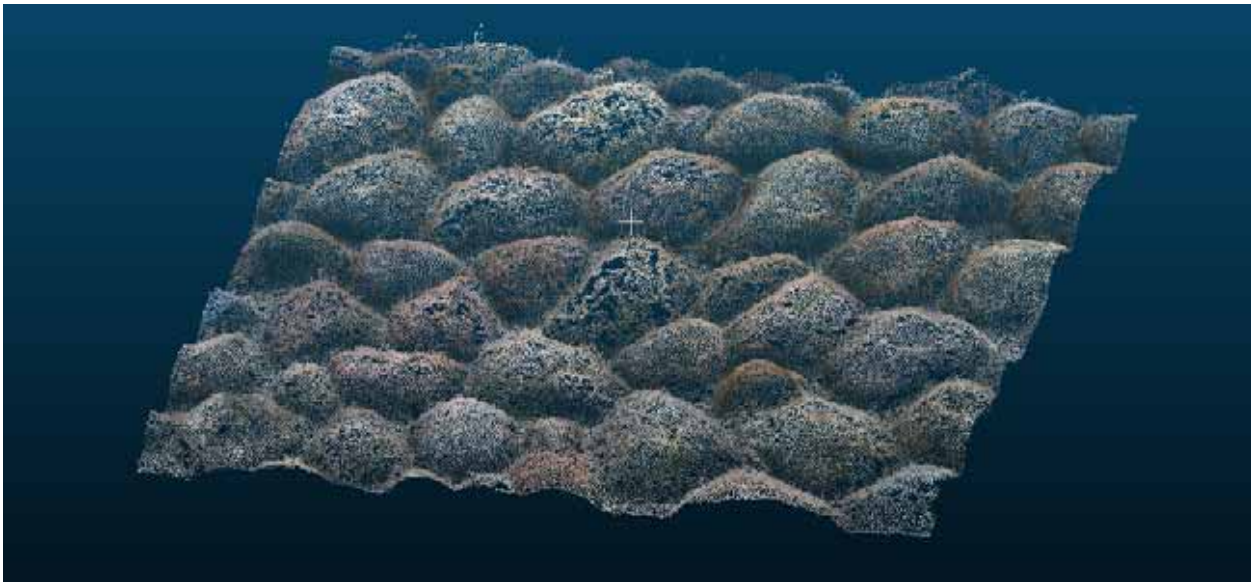

Figure 6.2: Point cloud from 3D reconstruction

6.3 Simulation Results

A camera setup, described in Section 4.6, was used for Monte Carlo simulations. The object was rendered at a distance of 900 mm directly under the central camera. For refractive calibration the multi-camera rig method was used with one image in-air and one in-water for each camera. Figures 6.3-6.11 show the histograms of error distributions for various parameters. The iterative closest point algorithm was used to adjust for the error in the location of the 3D model.

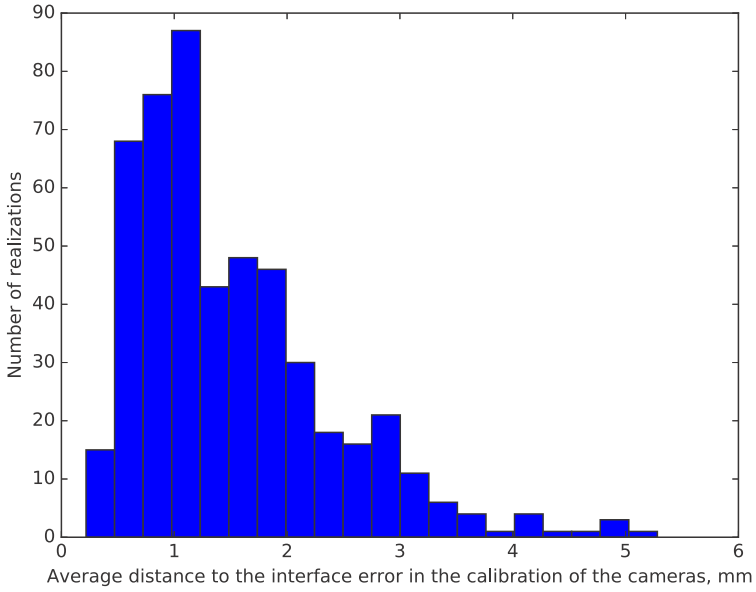

Figure 6.3: Distance to the interface error

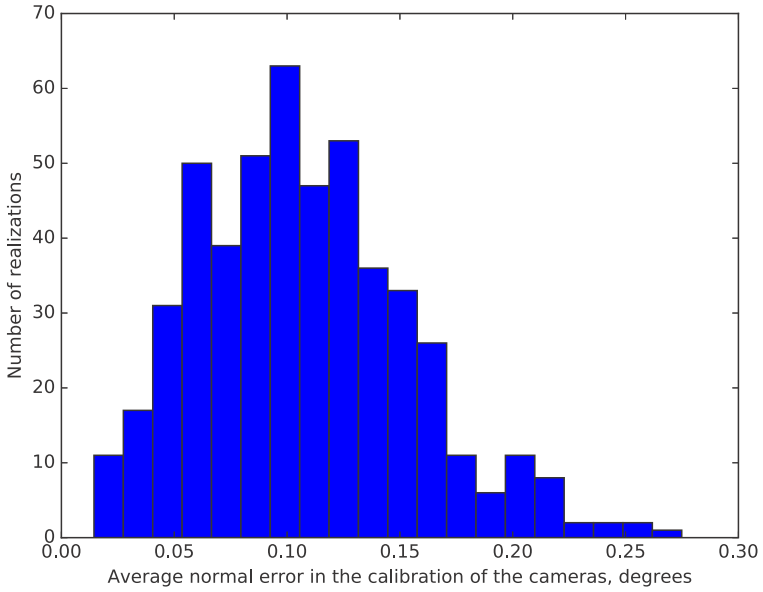

Figure 6.4: Normal to the interface error

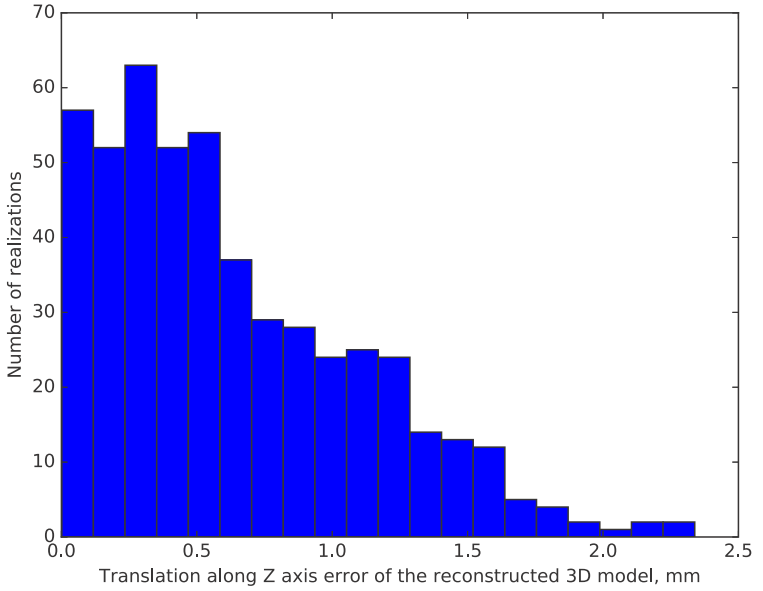

Figure 6.5: Translation along Z axis error

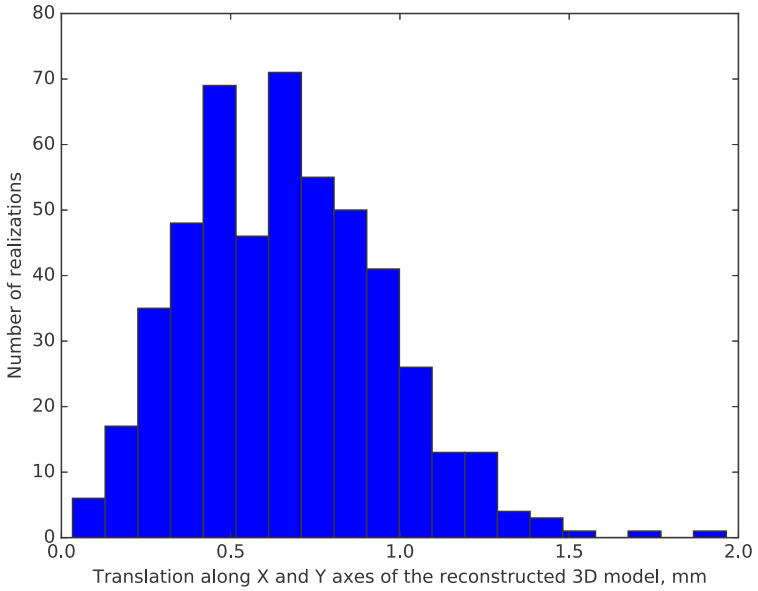

Figure 6.6: Translation along X and Y axes error

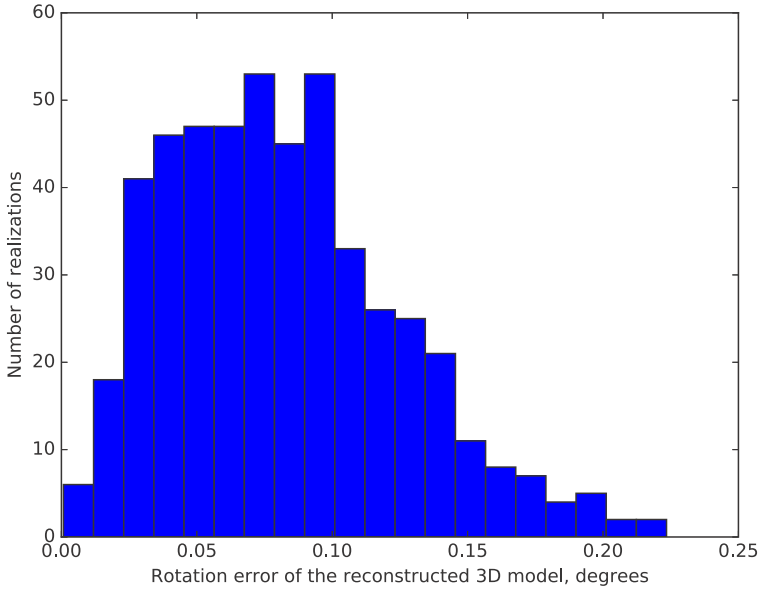

Figure 6.7: Rotation error

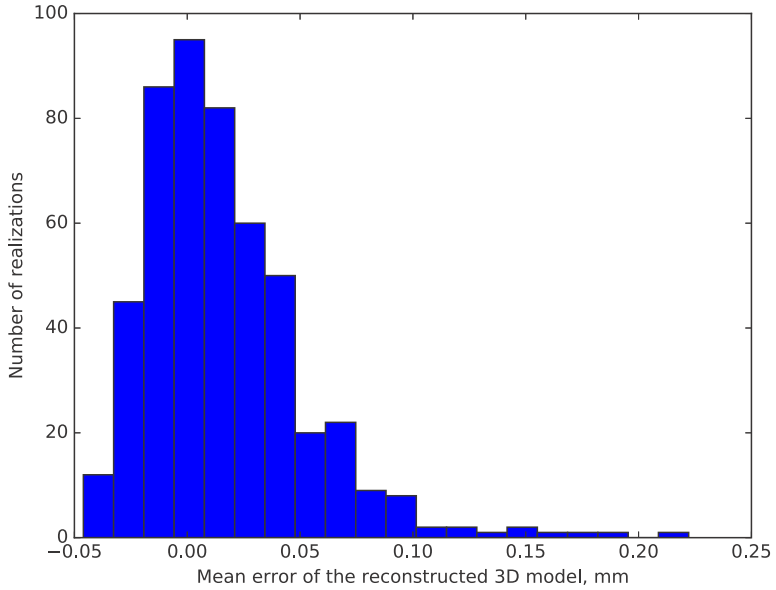

Figure 6.8: Mean error of 3D reconstruction

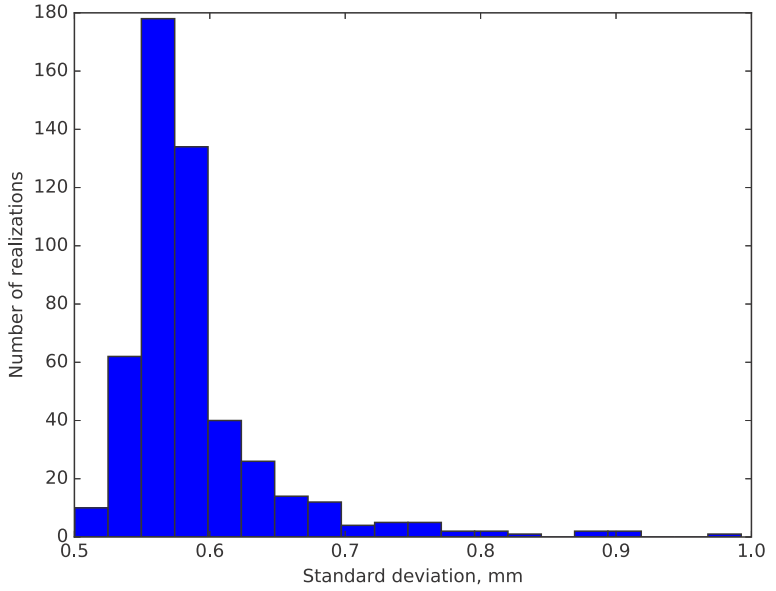

Figure 6.9: Standard deviation of 3D reconstruction

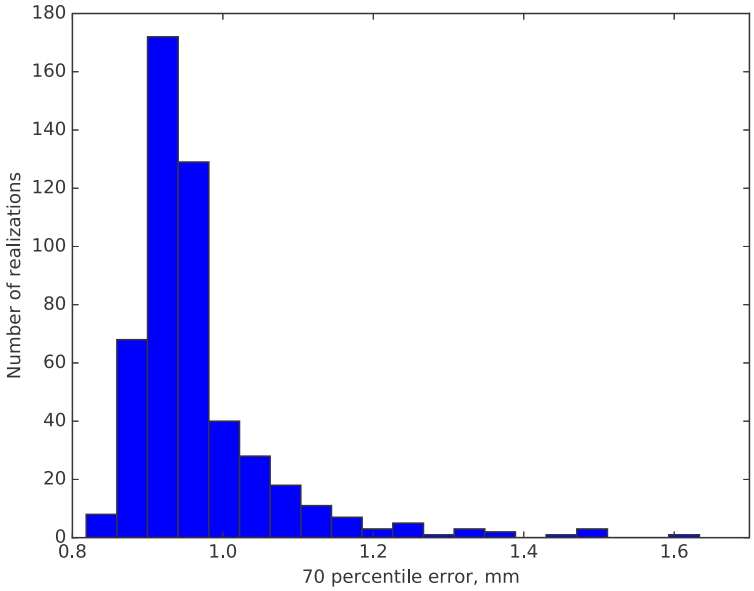

Figure 6.10: 70 percentile absolute maximum error, mm

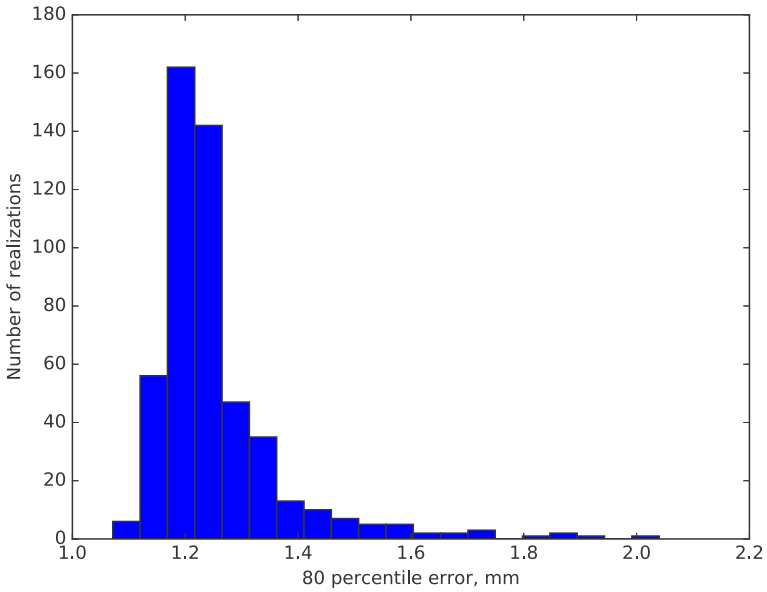

Figure 6.11: 80 percentile absolute maximum error

Strong correlation was found between the average error of the estimated distance to the glass interface of the cameras and the error in the location of the reconstructed 3D model as shown on Figure 6.12.

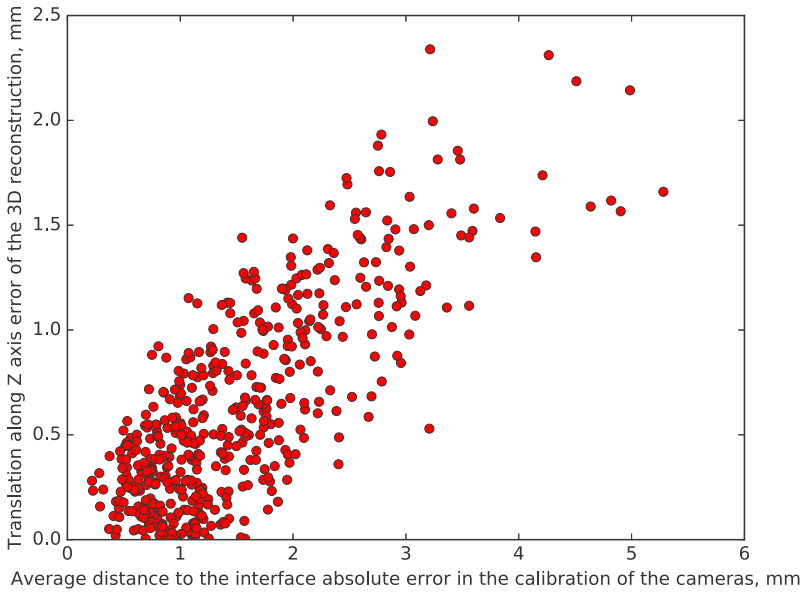

Figure 6.12: Intel Embree generated images

As expected, the reconstructed 3D models had a relatively small error, with the majority of the reconstructed points being within 1 mm of the groundtruth. Small errors in the calibration did not severely affect the accuracy of the 3D reconstruction. However, this may not be the case if the error in the calibration is more significant or a different camera setup is used. For any given camera setup, the simulation framework allows to estimate the quality of 3D reconstruction from real underwater imagery, which can be used during the preliminary stages of the multi-camera rig design.

CHAPTER 7
REAL EXPERIMENTS

A rig was constructed for conducting underwater experiments. Underwater flat port housings were built to protect the cameras from water. The housings were attached to the rig using flexible clasps. e-con Systems See3CAM_CU30 cameras were put into housings so that each camera has a slightly different distance to the glass interface and normal. Cameras have a USB 3.0 interface. All cameras connect to a USB hub with a 6ft cord. The hub is connected to a PC. We developed software that allows images to be taken from the cameras almost simultaneously.

A calibration target with a chessboard pattern was etched on a flat anodized aluminum panel. The size of each square $S = 11.994$ mm. The dimensions are 27×18 squares, which effectively means 432 detectable feature points.

All the underwater experiments were conducted in the UNH School of Marine Science and Ocean Engineering water tank. The rig was submerged in water using a crane.

7.1 Calibration Results

Two methods proposed in Chapter 4 to estimate the refractive parameters of the camera were used as well, as the method proposed in [9] for comparison. For the multi camera method we used four in-water and four in-air images. Table 7.1 shows the results of the calibration for 30 realizations. As expected, the fixed target calibration method gives very robust results. However, the multi-camera method also gives better results compared to the method proposed by Agrawal *et al.* [9].

# camera	Multi-camera method		Fixed target method		Method by Agrawal *et al.*	
	mean	σ	mean	σ	mean	σ
1	27.2	3.2	26.8	0.1	33.2	32.1
2	20.5	2.3	20.1	0.2	28.5	45.2
3	26.1	2.8	26.2	0.1	44.8	20.9
4	33.2	2.9	31.3	0.3	22.8	40.1
5	24.8	2.2	24.7	0.2	24.1	29.4

Table 7.1: Refractive calibration results

7.2 Multiview Reconstruction

Five images of a fish model captured underwater were used to perform a 3D reconstruction. To obtain groundtruth we performed a 3D reconstruction of the model in air, as well as captured the model using Kinect2 hardware. Unfortunately, the resulting Kinect2 model was of noticeably poor quality. Hence, the comparison with the Kinect model is omitted.

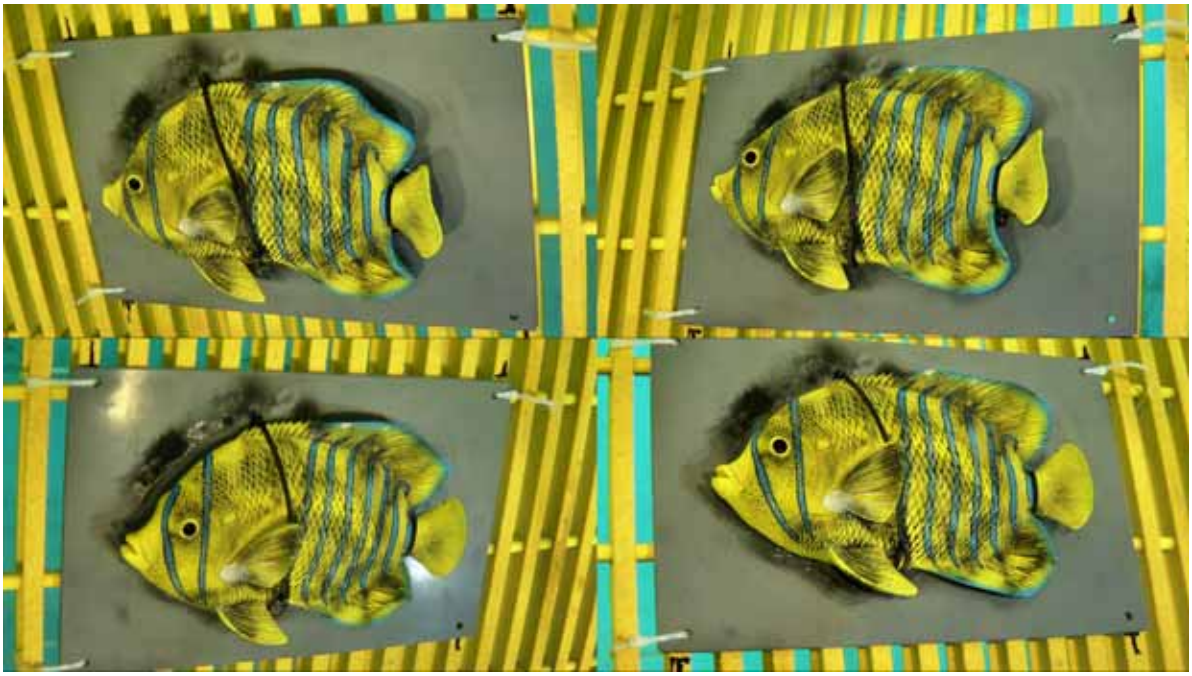

Figure 7.1: Images for multiview reconstruction

500 realizations of Monte Carlo simulations were ran for a system closely resembling the one we used for real experiments and compared the quality of thereconstructed 3D models to the one obtained from a real experiment. A methodology used by the Middlebury archive [23]

was applied to evaluate the quality of reconstruction. Table 7.2 shows the maximum absolute error for 70, 80 and 90 percentile.

	Monte Carlo simulations	Real experminet
70%, mm	1.16	1.38
80%, mm	1.55	1.95
90%, mm	2.37	3.43

Table 7.2: Accuracy of underwater 3D reconstruction

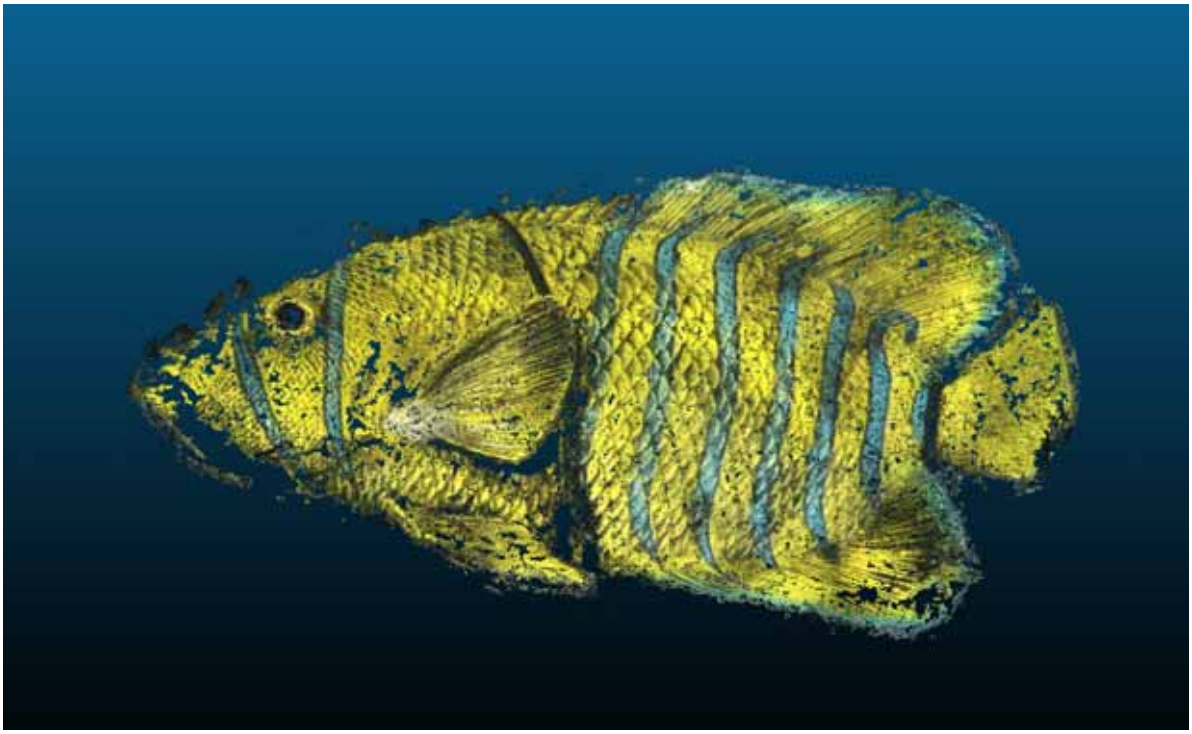

Figure 7.2: 3D model acquired using refractive PMVS2

CHAPTER 8
CONCLUSION AND FUTURE WORK

In this thesis, we proposed two novel methods for refractive calibration of a flat port housing. Both methods use a simple chessboard pattern and give more accurate estimates of the refractive parameters in comparison to the existing methods. A Monte Carlo simulation framework was developed to investigate the effects of error in the calibration and the quality of 3D reconstruction for a chosen multi-camera setup. We found that error in the distance to the refractive interface affects the location of a reconstructed model.

We focused our research on flat port housing modeling. However, often underwater imagery is acquired using hemispherical port housings. Although, such housings are supposed to compensate for the refraction effects due to the fact that all light rays passing through the refractive interface perpendicularly, it can only work if the camera's optical center is located precisely at the center of the hemisphere. In real life, it is impossible to achieve that, and therefore it could be useful to use a refractive model, which explicitly incorporates the shift of the optical center of the camera with respect to the hemisphere. We plan to investigate how such a shift affects the quality of 3D reconstruction and a possibility of using the proposed flat port housing calibration methods to estimate the refractive parameters of a hemispherical housing.

LIST OF REFERENCES

[1] Tali Treibitz, Yoav Schechner, Clayton Kunz, and Hanumant Singh. Flat refractive geometry. *IEEE transactions on pattern analysis and machine intelligence*, 34(1):51–65, 2012.

[2] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014.

[3] Amanda Piaia Silvatti, Fabio Augusto Salve Dias, Pietro Cerveri, and Ricardo M.L. Barros. Comparison of different camera calibration approaches for underwater applications. *Journal of Biomechanics*, 45(6):1112 – 1116, 2012.

[4] Ryan M Eustice, Oscar Pizarro, and Hanumant Singh. Visually augmented navigation for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 33(2):103–122, 2008.

[5] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 2000.

[6] Bruno Caprile and Vincent Torre. Using vanishing points for camera calibration. *International journal of computer vision*, 4(2):127–139, 1990.

[7] Guillaume Champleboux, Stéphane Lavallée, Pascal Sautot, and Philippe Cinquin. Accurate calibration of cameras and range imaging sensor: the npbs method. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pages 1552–1557. IEEE, 1992.

[8] Lourdes Agapito, Eric Hayman, and Ian Reid. Self-calibration of rotating and zooming cameras. *International Journal of Computer Vision*, 45(2):107–127, 2001.

[9] Amit Agrawal, Srikumar Ramalingam, Yuichi Taguchi, and Visesh Chari. A theory of multi-layer flat refractive geometry. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3346–3353. IEEE, 2012.

[10] Timothy Yau, Minglun Gong, and Yee-Hong Yang. Underwater camera calibration using wavelength triangulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2499–2506, 2013.

[11] S Glassner Andrew. An introduction to ray tracing, 1989.

[12] Jason Gedge, Minglun Gong, and Yee-Hong Yang. Refractive epipolar geometry for underwater stereo matching. In *Computer and Robot Vision (CRV), 2011 Canadian Conference on*, pages 146–152. IEEE, 2011.

[13] Xida Chen and Yee-Hong Yang. Two-view camera housing parameters calibration for multi-layer flat refractive interface. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 524–531, 2014.

[14] JG Fryer and CS Fraser. On the calibration of underwater cameras. *The Photogrammetric Record*, 12(67):73–85, 1986.

[15] Jean-Marc Lavest, Gérard Rives, and Jean-Thierry Lapresté. Underwater camera calibration. In *European Conference on Computer Vision*, pages 654–668. Springer, 2000.

[16] Oscar Pizarro, Ryan M Eustice, and Hanumant Singh. Relative pose estimation for instrumented, calibrated imaging platforms. In *DICTA*, pages 601–612. Sydney, Australia, 2003.

[17] Nathalie Pessel, Jan Opderbecke, and Marie-Jose Aldon. Camera self-calibration in underwater environment. 2003.

[18] Shahriar Negahdaripour, Caroline Barufaldi, and Ali Khamene. Integrated system for robust 6-dof positioning utilizing new closed-form visual motion estimation methods in planar terrains. *IEEE Journal of Oceanic Engineering*, 31(3):533, 2006.

[19] Tim Dolereit and Uwe Freiherr von Lukas. Calibration of shared flat refractive stereo systems. In *International Conference Image Analysis and Recognition*, pages 433–442. Springer, 2016.

[20] Leonie Traffelet, Thomas Eppenberger, Alexander Millane, Thomas Schneider, and Roland Siegwart. Target-based calibration of underwater camera housing parameters. In *Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium on*, pages 201–206. IEEE, 2016.

[21] Manolis IA Lourakis. A brief description of the levenberg-marquardt algorithm implemented by levmar. *Foundation of Research and Technology*, 4(1):1–6, 2005.

[22] Richard I Hartley and Peter Sturm. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997.

[23] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 519–528. IEEE, 2006.

[24] Ricardo Ferreira, João P Costeira, and João A Santos. Stereo reconstruction of a submerged scene. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 102–109. Springer, 2005.

[25] Visesh Chari and Peter Sturm. Multiple-view geometry of the refractive plane. In *BMVC 2009-20th British Machine Vision Conference*, pages 1–11. The British Machine Vision Association (BMVA), 2009.

[26] Y Furukawa. Pmvs2, department of computer science and engineering, university of washington, seattle, washington, 2010.

[27] Ingo Wald, Sven Woop, Carsten Benthin, Gregory S Johnson, and Manfred Ernst. Embree: a kernel framework for efficient cpu ray tracing. *ACM Transactions on Graphics (TOG)*, 33(4):143, 2014.