Diving into 3D – Underwater 3D Reconstruction

Master Thesis

Jonathan Cotugno
Master Curriculum Geomatics Engineering
Spring Semester 2017

Lead:
Prof. Dr. Konrad Schindler
Dr. Paul Beardsley

Supervisors:
Dr. Jan D. Wegner
Dr. Mathias Rothermel

Institute of Geodesy and Photogrammetry
Chair of Photogrammetry and Remote Sensing
ETH Zurich

Date of Submission: July 3rd 2017

Acknowledgements

I would like to thank the following people for their help and support throughout this master's thesis.

Prof. Dr. Konrad Schindler for allowing me to work within the Photogrammetry and Remote Sensing group during my master's thesis, for helping organizing this project and for his valuable instruction over the past two years.

Dr. Paul Beardsley from Disney Research Zürich for helping organize this collaborative effort and for his wisdom, support, and kindness throughout this project.

Dr. Jan Wegner for organizing the collaborative effort, for his commitment and support, and for allowing me to continue working on underwater reconstruction for my master thesis.

Dr. Mathias Rothermel for his supervision, endless patience, thorough expertise, and dealing with my ceaseless door-knocking and questions. I certainly would not have come this far without his support.

My family who has graced me with infinite support and comfort throughout my studies.

Abstract

Research in automatic 3D reconstruction systems in underwater settings has been scarce until recently. Demands from marine environmental studies, in particular, have driven the motivation for research into new, robust methods for reconstructing underwater scenes. However, applications extend into many other fields including archaeology, industrial inspection, and underwater augmented reality. When imaging underwater, there exists geometric and radiometric effects due to the propagation of light through water and through changing media. A camera inside of a protective housing with a flat viewing port window images light rays that have been refracted twice – once at the interface between water and the viewing port, and again at the interface between the viewing port and air inside of the housing. Therefore, due to these effects, typical perspective methods for 3D reconstruction either fail or exhibit systematic errors in an underwater setting.

This project takes an explicit physical model for underwater refraction and attempts to adapt a structure from motion (SfM) approach to underwater 3D reconstruction. The general pipeline for SfM is nearly the same as above water, however, the individual steps need to be adapted accordingly. The proposed system is able to robustly match features, triangulate observations, and estimate absolute pose. In the current state, the bundle adjustment is failing when noise is present in the image observations.

Contents

List of Figures

List of Tables

1 Introduction

Automatic 3D reconstruction systems are an actively researched topic in computer vision. However, research in reconstruction systems in underwater settings has been neglected until recent years. In the past decade, a wide variety of applications for underwater imaging and 3D reconstruction has arisen. Demands from environmental studies, particularly monitoring marine ecosystems, has driven research into robust and effective methods for underwater computer vision. However, underwater computer vision has other applications in archaeology, industrial inspection, and underwater augmented reality, for instance.

A particular area of research seeking automatic underwater reconstruction is the coral reef research conducted at Disney's Castaway Cay and Great Abaco in the Bahamas. Here, Disney's environmental researchers nurture damaged corals and attempt to regrow them a nursery. Damaged corals are transplanted back to their original reefs once growth and health have stabilized. Currently, coral growth is being measured manually by divers. The demand for a more efficient and automatic approach has led to research into the potential of imaging sensors for underwater reconstruction systems to more accurately and efficiently monitor growth of corals.

As of recent, there have been many uses for underwater imaging. These uses include industrial inspection, offshore monitoring, and archaeological documentation. However, utilization of passive sensors for 3D reconstruction underwater has been scarce. By far, the most popular method for depth recovery underwater has been the use of active sonar systems. A common use of sonar imaging is bathymetric mapping. Often, multibeam sonars are used in conjunction with side-scan sonars to produce detailed views of an underwater scene. However, these systems are typically very expensive, heavy, and require training to use. While these complex systems can supply a detailed view of an underwater scene, it is possible to meet or exceed the level of detail with imaging sensors underwater. This has already been seen and employed by using cameras to create image mosaics to overlay 3D models derived from sonar bathymetry. However, using cameras instead to create the 3D models could greatly decrease costs for underwater scene reconstruction. The methods for 3D reconstruction in air have been rigorously developed for years in the fields of computer vision and photogrammetry, and they now need to be adapted to underwater scenarios.

Diving into 3D is an underwater 3D reconstruction project in collaboration with Disney Research Zurich (DRZ) and the Photogrammetry and Remote Sensing (PRS) group at ETH Zurich. The project was realized as an interdisciplinary project and continued as a master's thesis within the scope of the Geomatics Engineering master's degree program at ETHZ. The project uses a small fisheye stereo camera unit and investigates solutions to incorporate fisheye and underwater refraction models to attempt to reconstruct 3D scenes in a controlled underwater setting. The work completed in the interdisciplinary project includes a familiarization with theory and applications of underwater imaging, experimental setups for underwater acquisition, and most importantly, a calibration technique for both the fisheye stereo camera unit and the underwater camera housing.

This project takes the work completed during the interdisciplinary project and attempts to adapt 3D reconstruction techniques to underwater scenarios. The main purpose of this project is to adapt the well-known structure-from-motion (SfM) technique that recovers 3D structure of a scene from 2D images to an underwater scenario. Adaptation of a typical SfM algorithm is done by modifying the technique to include a model for refraction.

This report begins with a theoretical background of camera models, underwater imaging, and structure from motion in Chapter 2. Then, the steps and procedures undergone during the project are described in the Chapter 3. Next, results of the proposed system are presented and discussed in Chapter 4. Lastly, a conclusion of the the system and suggestions for further improvement and future work are provided in Chapter 5.

2 Theoretical Background

As stated previously, this project attempts to adapt conventional camera models and 3D reconstruction techniques to an underwater setting. This chapter underlines the theory needed to understand the techniques used from both above water and below water settings. The chapter begins with a section on the camera models related and used in this project, followed by theory on refraction, and lastly, a section on structure-from-motion, the 3D reconstruction technique used in this project.

2.1 Camera Models

2.1.1 Pinhole Camera Model

Most consumer cameras used in photogrammetry and computer vision are modeled with a central projection model – one such that incoming rays converge at a single projection center, which is the camera center. The most basic model of this form is called the *pinhole camera* model. In the pinhole model, there exists a plane at $Z = f$, where f is the focal length of the camera, such that a 3D point in the *world coordinate frame* $\mathbf{X} = (X, Y, Z)^\top$ can be mapped to this plane by (Hartley and Zisserman, 2004)

$$(X, Y, Z)^\top \mapsto (f\frac{X}{Z}, f\frac{Y}{Z})^\top \tag{2.1}$$

where the map is from 3D Euclidian world space to 2D Euclidian image space. This mapping is visualized in Figure 2.1.

This model assumes that the camera sensor is perfect. This is almost never the case and as such, there are offsets to the *principal point* (center of image coordinate system) (x_0, y_0), deviations in scale due to non-square pixels (α_x, α_y), and potentially a skew parameter s. If these parameters are known or can be found, we can define a calibration matrix K

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

Further, a camera is typically not at the origin of the world coordinate frame. Therefore, a transformation from the world coordinate frame to the camera coordinate frame needs to be applied. This transformation can be represented by a 3×3 rotation matrix R and the location of the camera center in the world coordinate frame $\mathbf{C}$. The transformation from world coordinate frame can be applied by

$$\mathbf{X}_{cam} = \begin{bmatrix} \mathrm{R} & -\mathrm{R}\mathbf{C} \\ 0 & 1 \end{bmatrix} \mathbf{X} \tag{2.3}$$

where $\mathbf{X} = (X, Y, Z)^\top$ is a point in the world coordinate frame represented in homogeneous coordinates and $\mathbf{X}_{cam}$ is the point in the camera coordinate frame represented in homogeneous

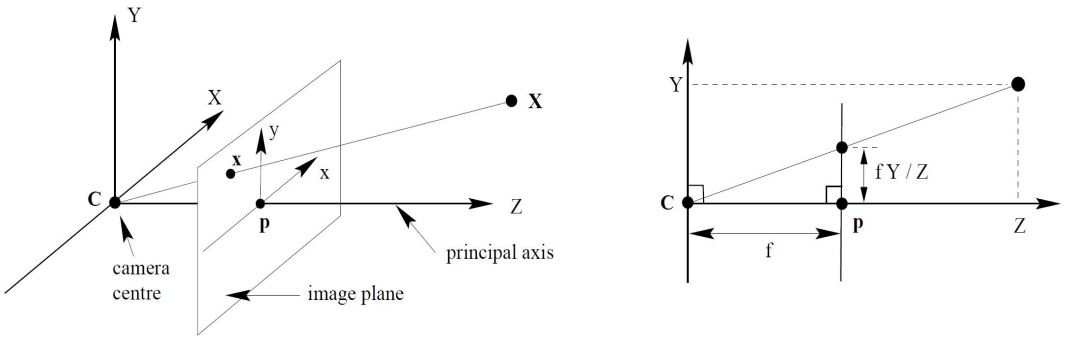


Figure 2.1: The pinhole camera model. The left diagram shows the model from a 3D perspective. The right diagram shows the cross section of the model and highlights the relationship between the focal length and the projection of a 3D point into the 2D image plane. (Image source: Hartley and Zisserman (2004))

coordinates.

Finally, joining these together, a camera P can be defined as

$$\lambda\ \mathbf{x} = \mathrm{KR}\left[\mathrm{I} \quad | -\mathbf{C}\right] = \mathrm{P}\mathbf{X} \tag{2.4}$$

where P is the camera projection matrix that projects a 3D point in the world coordinate frame into the camera's 2D image coordinate frame up to scale a λ.

Typically there are additional distortions due to lens defects, e.g. barrel distortion. A fisheye lens utilized significant barrel distortion to map wide-angle views of a scene to a 2D image. In this project, a fisheye lens is used and modeled using the method by Kannala and Brandt (2006). Further details on fisheye lenses and the model used to acccount for fisheye distortion can be found in Cotugno et al. (2016).

Together, the pinhole camera model with the calibration matrix, Euclidian transformation, and distortion corrections will be referred to as a *perspective camera*.

2.1.2 General Camera Model

Recent research has led to the realization that there are limitations to the perspective camera model. Variations of camera types and camera systems have become more popular. These include, for instance, catadiotropic cameras, camera clusters of a single or multiple camera types, and compound camera systems. A particularly popular case is the autonomous vehicle that has a multi-camera system that needs to be related to each other in some non-central center of projection fashion.

Thus, to deal with this, Grossberg and Nayar (2001) developed the *general camera* model to deal with arbitrary imaging systems. The main idea is that all imaging systems need to have a mapping from incoming scene rays to photo-sensitive elements. Essentially the smallest element

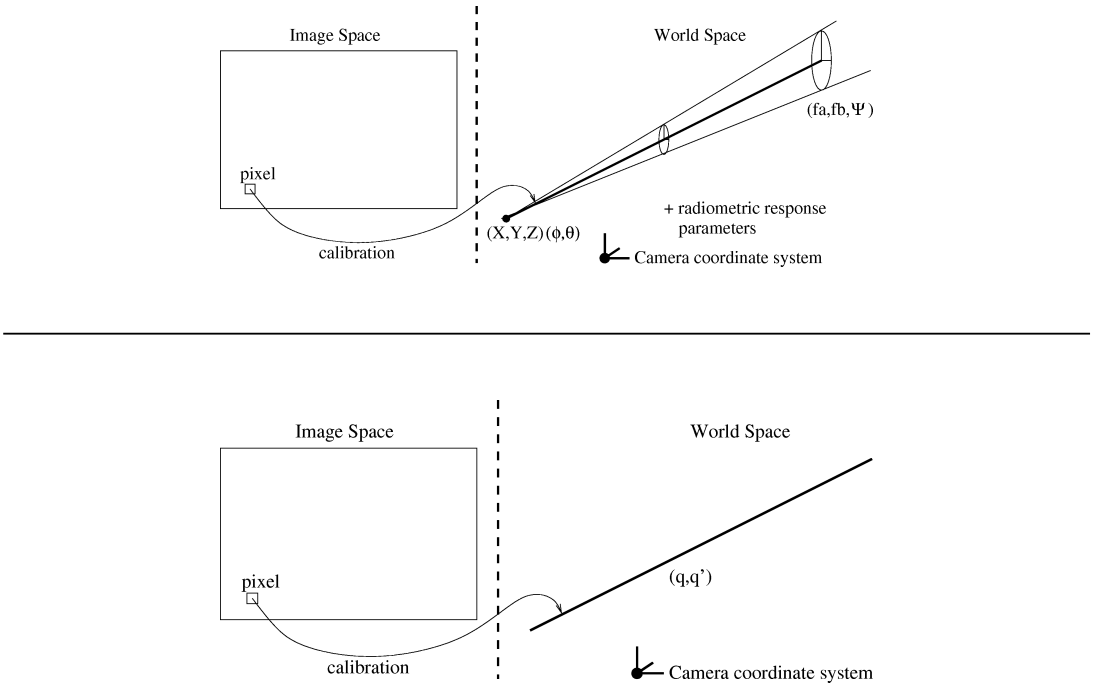


Figure 2.2: The top image shows the general camera model from Grossberg and Nayar (2001) which includes a space sampled in a direction centered at the ray. The bottom image shows the simplified model by Pless (2003) which only uses the ray and its direction parameterized as a Plücker line. (Image source: Pless (2003))

in the imaging system is a virtual element that measures the direction and radiometric response of an incoming light ray. These virtual elements are called "raxels".

Later, Pless (2003) takes the general camera model and extends the model, using it to estimate motion of an imaging system. Pless first simplifies the raxel model to the ray that a given pixel samples in space. The differences between the simplification by Pless (2003) and Grossberg and Nayar (2001) can be seen in Figure 2.2. This model will be referred to as the Pless Model.

The Pless model uses Plücker coordinates to describe an arbitrary line in space that a pixel samples. Plücker coordinates are a parameterization of lines in projective 3-space. Plücker coordinates are advantagous because they provide convenient methods for common transformations and projections performed in computer vision. The Plücker coordinates representation of a line is a 6-vector that is broken into to two 3-vectors. The first 3-vector is the direction of the line $\mathbf{q}$, and the second 3-vector is the moment of the line $\mathbf{q}'$. If a point on the line $\mathbf{X}$ and a ray in the direction of the line $\mathbf{q}$ are known, then the Plücker coordinates of the line can be formed as follows (Pless, 2003)

$$\mathbf{L} = \begin{pmatrix} \mathbf{q} \\ \mathbf{q} \times \mathbf{X} \end{pmatrix} = \begin{pmatrix} \mathbf{q} \\ \mathbf{q}' \end{pmatrix} \tag{2.5}$$

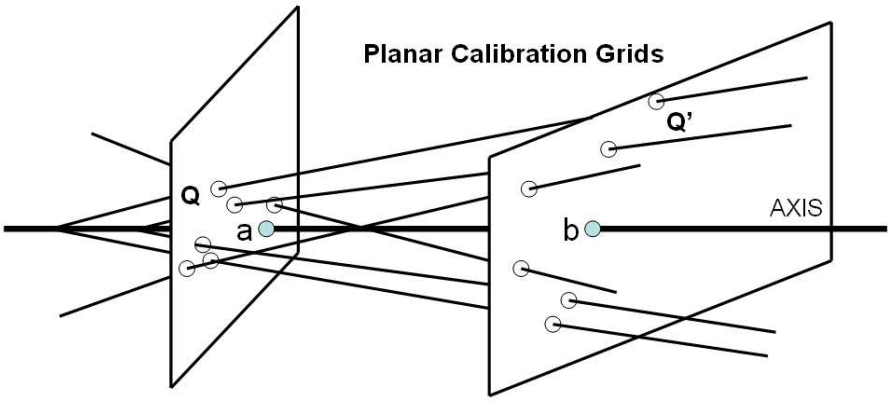


Figure 2.3: An example of the axial camera model with a single camera. Rays of points on a planar calibration board intersect the axis of the axial camera. (Image adapted from source: Sturm and Ramalingam (2004a))

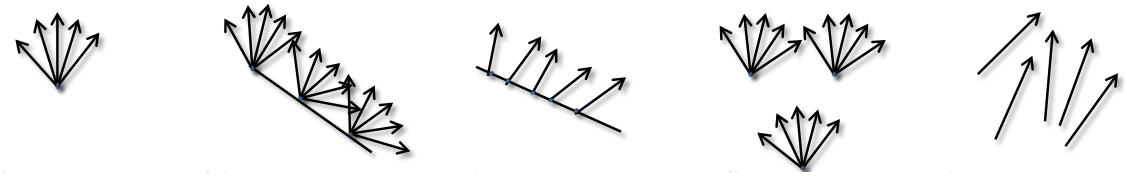


Figure 2.4: From left to right: The perspective camera. An array of perspective cameras modeled as an axial camera. A single axial camera. An array of cameras modeled as a general camera. A single general camera. (Image adapted from source: Jordt (2014))

where $\mathbf{L}$ is the line in Plücker coordinates. The vector $\mathbf{q}$ can be of any magnitude. Thus, in the Pless model, imaging can be modeled if, for each pixel, a point and the direction of the line the pixel samples is known or can be found.

2.1.3 Axial Camera Model

While the perspective camera model and the general/Pless model show opposite sides of the spectrum with regard to assumptions of the imaging system, there exists other intermediate models. One of which is called the *axial camera* model. The axial camera is also a non-central camera model. However, the axial camera model contains the constraint that there exists an axis such that all incoming rays into the camera intersect. This model can be applied to a single non-central camera or a stereo system of two perspective cameras. An example of these two can be seen in Figure 2.3. This model and a calibration procedure based on the model were developed in the paper Sturm and Ramalingam (2004a).

To summarize the differences between the camera models described in this section, a visualization of the different types of rays in the models can be seen in Figure 2.4.

2.2 Underwater Imaging

When submerging a camera underwater, many assumptions used above water are not valid underwater. There are geometric and radiometric effects that occur underwater that need to be accounted for. In addition, a camera typically needs to be enclosed in a protective housing in order to prevent damage to the camera. A housing usually has some sort of viewing port for the light rays to enter the housing and into the camera center. This viewing port can also cause geometric effects on the incoming rays. This section focuses on the theory behind underwater imaging and how to model its effects.

Different media have different *refractive indicies*. The refractive index of a medium describes how light propogates through it with respect to light in a vacuum. The refractive index of a medium can be calculated by

$$n = \frac{c}{v} \tag{2.6}$$

where n is the refractive index, c is the speed of light in a vacuum, and v is the phase velocity of light through the medium in question.

When light propagates from one medium to another, it is bent or *refracted* at the optical interface between the two media. The amount refracted is dependent on the refractive indices of the two media. This phenomena is illustrated by Snell's Law which describes the change in propagation direction from one media to another by

$$\sin(\theta_{i+1}) = \frac{\mu_i}{\mu_{i+1}} \cdot \sin(\theta_i) \tag{2.7}$$

where θ_i is the angle of incidence of the incoming ray from medium i with refractive index μ_i and θ_{i+1} is the refraction of the ray after crossing the interface into medium $i+1$ with refractive index μ_{i+1}. The angle of incidence and the refraction are measured with respect to the normal of the interface. This effect is visualized in Figure 2.5

According to these established principles, a camera placed inside of a protective housing with a window port of some material (typically acrylic or glass) will image rays that have been refracted twice: once at the interface between water and the port outside of the housing, and again at the interface between air and the port inside of the housing. These geometric effects need to be accounted for in order to accurately model viewing geometry to estimate 3D structure of a scene.

An underwater camera housing can utilize different types of viewing ports. Two of the most common types of ports are the flat port and the dome viewing port. Figure 2.6 shows the two different types of ports and the refraction that occurs in each. As can be established from Equation 2.7, any incoming ray with a non-zero incidence angle is refracted. In the case of a flat shaped port, refraction occurs twice – once at each interface.

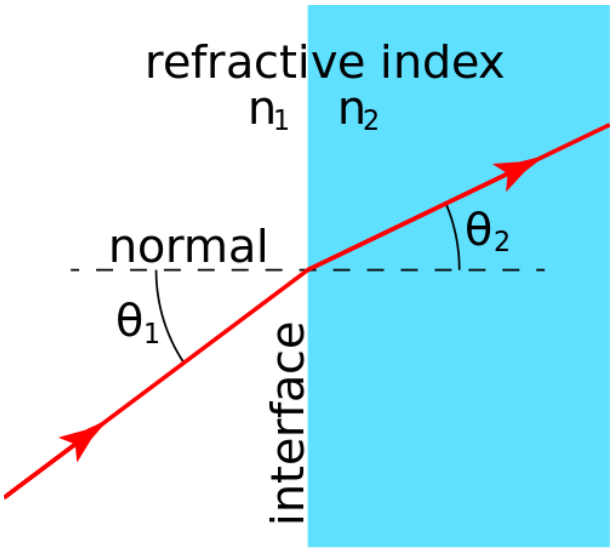


Figure 2.5: Diagram showing refraction at the optical interface as a light ray propagates through a medium change. (Image source: WikiCommons (2015))

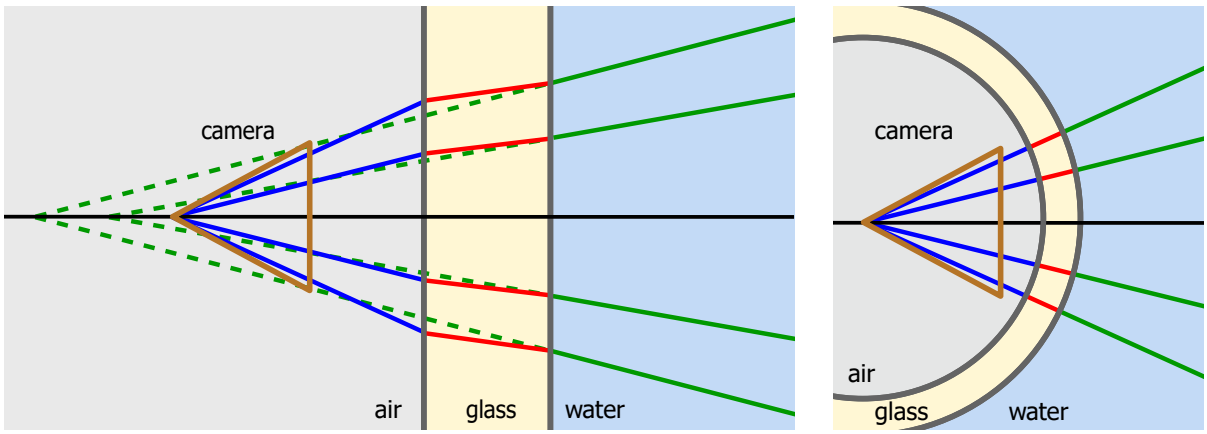
In contrast, in the case of a dome port in Figure 2.6b, rays enter the port normal to the refractive interface. Therefore, no refraction occurs. However, this concept is theoretical. For this to work in practice, the camera center needs to be perfectly aligned with the dome center and the dome must be perfectly spherical. Of course, this is usually not the case. Dome ports usually exhibit some deviations from a perfect sphere due to imprecise machining, and maintaining the camera at the center of the dome is a practical burden. Further, good quality dome ports are particularly expensive due to the precise machining needed for manufacturing. Thus, dome ports are not the best practical choice for a low-cost underwater 3D reconstruction system, especially if refraction in a flat port can be modeled.

In addition to the geometric effects, there are radiometric distortion effects that occur underwater. Photons underwater collide with water molecules and cause absorption and scattering of photons (Jordt, 2014). These effects are wavelength dependent and therefore have an effect on color and intensity of an underwater image. However, in this project, radiometric effects are not considered.

2.2.1 Accounting for Refraction

There have been different methods in the literature to account for the geometric effects of refraction on viewing geometry. In general, three different techniques have been used to account for refraction in underwater imaging.

First, the perspective camera model can be be used although it has been shown that it is invalid underwater and there exists a systematic model error (Sedlazeck and Koch, 2012). Some methods calibrate a perspective camera above and below water to examine how much of the error is absorbed in the camera parameters, and adapt the focal length and distortion parameters (Fryer



(a) (b)

Figure 2.6: Refraction effects with respect to a (a) flat port and (b) a perfectly dome-shaped port. Rays are illustrated as follows: rays in water are in green, rays in between the port material are in red, and rays in air into the camera are in blue. The dashed green lines shown in (a) illustrate the path the incoming rays from water without considering refraction. (Image source: Cotugno et al. (2016))

and Fraser, 1986; Lavest et al., 2000) accordingly. Bryant et al. (2000) uses underwater calibration images of a checkerboard and adds a coefficient for radial distortion.

Second, the general or axial camera models can be used. In this case, refraction can be explicitly modeled as opposed to the previous case. In the same paper as formulating the general camera model, Grossberg and Nayar (2001) also proposes a calibration system and states that general cameras are defined by their caustics. The caustic of an imaging sensor is the surface such that the set of all camera rays are tangent. A paper by Sturm and Ramalingam (2004b) extends this work by making calibration more generic by using three or more images of calibration objects where viewing positions are unknown. The camera used here is only described by its viewing rays. In Ramalingam et al. (2006), a generic structure-from-motion algorithm is introduced based on the work from Sturm and Ramalingam (2004b) where rays are clustered to approximate perspective cameras. These ray clusters are used to synthesize a perspective camera plane. This is a general concept that of synthesized perspective cameras used in Ramalingam et al. (2006), other papers, and particularly in this project, will be referred to as a *virtual camera*. Lastly, Mouragnon et al. (2009) introduces a complete 3D reconstruction system that uses a catadiotropic camera where each raxel is known from calibration.

Thirdly, an explicit, physical model for refraction can be used. Using a general camera model to account for refraction relies on having independent arbitrary perspective centers and directions for each ray, which increases complexity of the problem (Jordt, 2014). Thus, creating a physical model for refraction in the camera housing can reduce complexity and make calibration and 3D reconstruction less difficult. One of the earliest methods of forming a physical refraction model is Treibitz et al. (2008). Treibitz et al. make the significant assumption that the viewing port of the underwater housing is very thin. Therefore, the effect of refraction is approximated by only having one refraction interface – an air-water refractive interface. In addition, they assume that

the viewing port is orthogonal to the optical axis of the camera. In Li et al. (1997), a complete physical model is developed using a photogrammetric approach. Rays are optically traced and a stereo-rig is used to triangulate the traced rays in water. Sedlazeck and Koch (2011) also introduces a complete physical model with no assumptions on inclination angle between the viewing port normal and the camera optical axis. Further, they explicitly model the thickness of the view port as well as the distance from the camera center to each of the interfaces. They provide a virtual camera error function for calibration similar to Sturm and Ramalingam (2004b) except compute a virtual perspective camera for *each* pixel. Transformation for each pixel into the virtual camera requires solving a costly computation for a caustic point for each virtual camera. However, this computation is still more efficient than solving a non-linear optimization for forward projection for the reprojection error.

The paper by Agrawal et al. (2012) analyzes refractive geometry with up to n layers of successive parallel refraction. Their work has significant value in the field of underwater imaging. One of the most important findings of the paper is that a perspective camera behind a n refractive planes corresponds to an axial camera. The axis of the axial camera is a line that is parallel to the normal of the refractive planes (interfaces) and goes through the camera's perspective center. The main constraint of this theory is that the successive refractive planes should be parallel. However, the orientation of the camera does not need to be fronto-parallel which allows for cheaper, consumer-grade equipment. Agrawal et al. (2012) provide a calibration routine that only requires the intrinsics of the camera and coordinates of the 2D-3D correspondences of an imaged calibration board in both the image and world coordinate frame to be known. The calibration provides estimates for the interface normal, distance to each interface, and potentially the indices of refraction if they are unknown. Further, they introduce a novel method for analytical forward projection. Forward projection differs on the number of refractive planes and the indices of refraction of the first and last media. In the case of standard underwater imaging, there are two refractive planes and the first and last indices of refraction differ. In this case, it is shown that forward projection of a 3D point into the image can be found by solving a 12^{th} degree polynomial. While this can be expensive, it is still much more efficient than previous forward projection techniques that required non-linear optimization.

The work produced by Agrawal et al. (2012) was then integrated into the previous work by Anne Jordt/Sedlazeck et al. (Sedlazeck and Koch, 2011, 2012). The realization that the refractive camera is an axial camera allowed for computation of the virtual camera based on the axial camera axis rather than solving for a caustic at each pixel. This greatly increased efficiency and decreased complexity of the problem. Jordt/Sedlazeck et al. then continued their work with the new refractive geometry insights (Jordt-Sedlazeck and Koch, 2012, 2013; Jordt-Sedlazeck et al., 2013), finally culminating in a complete underwater 3D reconstruction pipeline in Jordt et al. (2016). The pipeline includes underwater refractive calibration, structure-from-motion, bundle adjustment, and dense 3D reconstruction via a plane-sweep method.

In this project, the refraction model and 3D reconstruction methods proposed build on the insights developed from Agrawal et al. (2012) and Jordt et al. (2016).

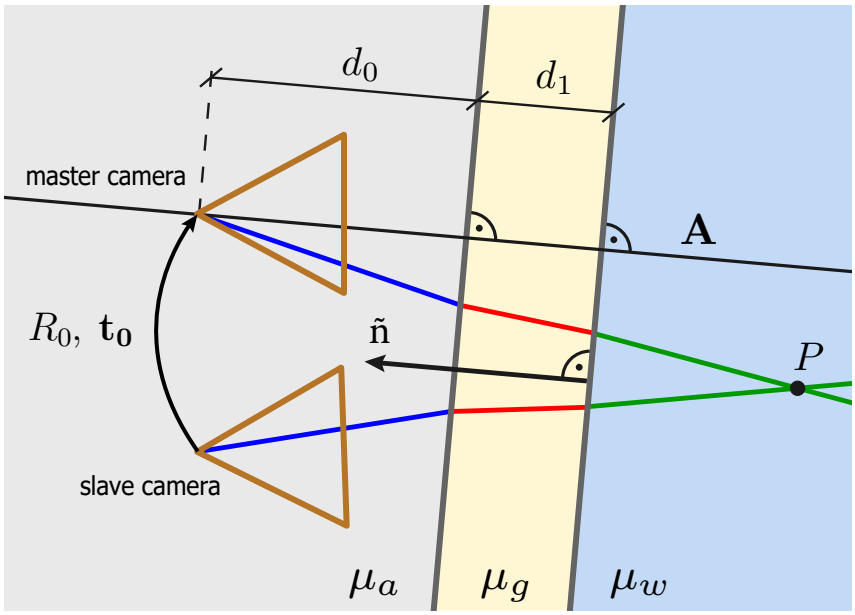


Figure 2.7: The proposed refraction model for a stereo-rig with a flat viewing port. Illustrated are light rays in water (green); light rays in the viewing port material (red); light rays in air entering the camera (blue); relative orientation between the two cameras $R_0, \mathbf{t}_0$; the parameters of the camera housing. An exemplary 3D point seen in both cameras is shown as $\mathbf{P}$. (Image source: Cotugno et al. (2016))

2.2.2 Refraction Model

In this work, a stereo-camera is used inside of a housing with a flat viewing port. The refraction model proposed here requires two calibrations. The first calibration is in air for the intrinsics of the cameras and the relative pose between the two. The second is underwater for the parameters of the housing that allows for explicit modeling of refraction. The model will be explained in detail in this section.

The following parameters are used to entirely account for refraction in the proposed system:

- The normal $\tilde{\mathbf{n}}$. The normal is in the local coordinate frame of the master camera.
- The distance from the camera center of the master camera to the first (inner) refractive interface d_0 measured in the direction of $\tilde{\mathbf{n}}$.
- The thickness d_1 of the viewing port.
- The indicies of refraction for air μ_a, glass μ_g, and water μ_w.

The auxiliary parameter $\mathbf{A}$ is the axis of the axial camera a the master camera location. It is parallel to the normal $\tilde{\mathbf{n}}$ and passes through the camera center of the master camera. All of these parameters with the addition of the relative transformation parameters between cameras in the stereo-rig can be seen in Figure 2.7.

Ray Tracing

Using the physical model parameters listed above, the ray for any pixel can be traced from the camera to a ray in water, explicitly taking into account refraction. Ray tracing for the underwater housing rig follows Snell's Law shown in Equation 2.7. Beginning from a ray in air can be back projected from its pixel coordinate in homogeneous representation $\mathbf{x} = (x, y, 1)^\top$ by (Hartley and Zisserman, 2004)

$$\tilde{\mathbf{X}}_a = \frac{K^{-1}\mathbf{x}}{\|K^{-1}\mathbf{x}\|_2} \tag{2.8}$$

where $\tilde{\mathrm{X}}_a$ is the normalized ray in air inside of the housing and K is the calibration matrix. The formula in Equation 2.8 is the typical case for a perspective camera. However, in this case of this project where a fisheye lens is used, the generic fisheye model by Kannala and Brandt (2006) is used to back project a pixel to its ray in air.

Once this ray is known, the housing parameters can be used to trace this ray into the next medium. This procedure can be done using (Agrawal et al., 2012)

$$\tilde{\mathbf{X}}_g = \frac{\mu_a}{\mu_g} \cdot \tilde{\mathbf{X}}_a + \frac{-\mu_a \tilde{\mathbf{X}}_a^\top \tilde{\mathbf{n}} - \sqrt{\mu_a^2 \left(\tilde{\mathbf{X}}_a^\top \tilde{\mathbf{n}}\right)^2 - \left(\mu_a^2 - \mu_g^2\right) \tilde{\mathbf{X}}_a^\top \tilde{\mathbf{X}}_a}}{\mu_g} \cdot \tilde{\mathbf{n}} \tag{2.9}$$

where $\tilde{\mathbf{X}}_g$ is the ray traced into the glass port. The ray should be normalized by $\tilde{\mathbf{X}}_g = \frac{\tilde{\mathbf{X}}_g}{\|\tilde{\mathbf{X}}_g\|_2}$. The formula in Equation 2.9 can be used to continue tracing a pixel's ray to its ray in water $\tilde{\mathbf{X}}_w$, while changing the previous rays and indices of refraction accordingly.

In addition to rays in each medium, the intersection of the rays with each refractive interface can be determined. The most important of the two intersections is the last and thus the equation for which is displayed in Equation 2.10 (Agrawal et al., 2012).

$$\mathbf{X}_s = \frac{d_0}{\tilde{\mathbf{X}}_a^\top \tilde{\mathbf{n}}} \tilde{\mathbf{X}}_a + \frac{d_1}{\tilde{\mathbf{X}}_g^\top \tilde{\mathbf{n}}} \tilde{\mathbf{X}}_g \tag{2.10}$$

The outer intersection point is labeled $\mathbf{X}_s$ because it is considered the *starting* point of the ray in water. This is an important note because throughout the course of this project, image points $\mathbf{x}$ are often represented by their point on the outer interface and traced ray in water $\left(\mathbf{X}_s, \tilde{\mathbf{X}}_w\right)$.

Another important note is that these point-ray representations of image points are in the coordinate frame of the camera (master or slave). If needed, they can be transformed into the world coordinate frame by

$$^{\mathrm{WC}}\mathbf{X}_s = \mathrm{R}^{\mathrm{CC}}\mathbf{X}_s + \mathbf{C} \qquad ^{\mathrm{WC}}\tilde{\mathbf{X}}_w = \mathrm{R}^{\mathrm{CC}}\tilde{\mathbf{X}}_w \tag{2.11}$$

where $\left(^{\mathrm{WC}}\mathbf{X}_s, ^{\mathrm{WC}}\tilde{\mathbf{X}}_w\right)$ and $\left(^{\mathrm{CC}}\mathbf{X}_s, ^{\mathrm{CC}}\tilde{\mathbf{X}}_w\right)$ are the point-ray representations in the world coordinate frame and the camera coordinate frame respectively. $\mathbf{R}$ and $\mathbf{C}$ are the rotation and

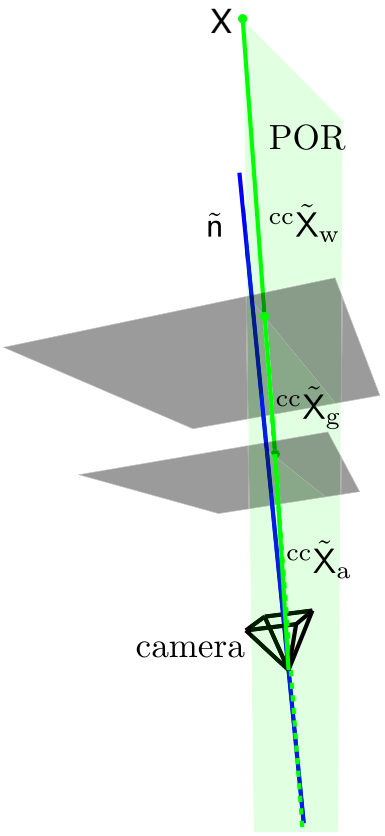


Figure 2.8: The plane of refraction constraint theorized by Agrawal et al. (2012). All rays $\tilde{\mathbf{X}}_a$, $\tilde{\mathbf{X}}_g$, and $\tilde{\mathbf{X}}_w$ and the interface normal $\tilde{\mathbf{n}}$ exist in the same plane. (Image adapted from: Jordt et al. (2016))

translation from camera coordinate system into the world coordinate system .

Forward Projection

In earlier studies (Sedlazeck and Koch, 2012) forward projection of a 3D point into the image was solved numerically using an optimization scheme for the two intersection points of the refractive interfaces. This optimization scheme was exceedingly costly and would have made any reconstruction system too long for any realistic application.

However, in Agrawal et al. (2012) additional useful characteristics of the refractive camera besides the ones already discussed were found. The most important in regard to forward projection is what they called the plane of refraction (POR) constraint. The POR constraint says that all traced, refracted rays sampled by a pixel lie in the a common plane as the axial camera axis $\mathbf{A}$. Further, all PORs intersect the axis $\mathbf{A}$, shown in Figure 2.7. This theory is visualized in Figure 2.8 and the equation for the constraint is shown in Equation 2.12 (Jordt et al., 2016).

$$\left(\mathrm{R}^{\top\,\mathrm{WC}}\mathbf{X} - \mathrm{R}^{\top}\mathbf{C}\right)^{\top}\left(\tilde{\mathbf{n}} \times^{\mathrm{CC}} \tilde{\mathbf{X}}_w\right) = 0 \tag{2.12}$$

Equation 2.12 shows that a 3D point in the world coordinate system ${}^{\mathrm{WC}}\mathbf{X}$ transformed into the camera coordinate system lies within the POR. This development is used in Agrawal et al.

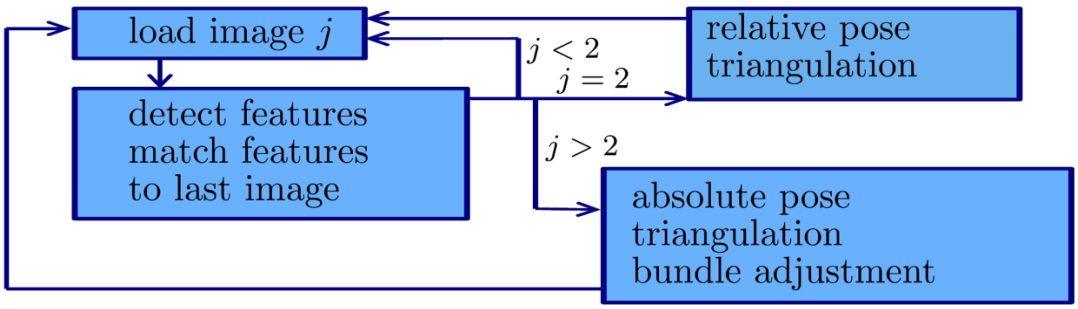


Figure 2.9: A general pipeline for sequential SfM. (Image source: Jordt et al. (2016))

(2012) to contrain forward projection of a 3D point into the camera to be on the POR. The paper shows that if the first and last indices of refraction differ – which is the case here ($\mu_a \neq \mu_w$) – then the image coordinates of a 3D point forward projected into the image can be known by solving a 12^{th} degree polynomial.

This development significantly increases efficiency of forward projection. Although, in the scope of a SfM system, particularly in bundle adjustment where image observations can be up to hundreds of thousands of observations (depending on the scene), solving a 12^{th} degree polynomial for each point is still costly.

2.3 Structure-from-motion

In general, structure-from-motion (SfM)is a technique to estimate 3D structure of a scene given an input of 2D images of the scene. Typically, the input images can be sequential, which can give prior knowledge to the method, though there are variations for unordered images. The output of SfM is usually the poses of each camera in the 3D scene and a sparse point cloud of the scene. The output from SfM can be used as input into a dense 3D reconstruction system to create a final, dense 3D model of the scene.

As mentioned, there are different variations of SfM. In this project, the *sequential* approach is used. This is because images are taken in a a video sequence of the scene. The general pipeline for SfM will be outlined here and can be seen in Figure 2.9.

Consider the case of a calibrated perspective camera that takes a sequential set of images of a scene. The first camera is considered at the origin of the world coordinate frame. First, the initial image ($j = 1$) is input into the system and features are detected and described. Then, the second image ($j = 2$) input into the system and features are detected and described. Then, the features from image $j = 2$ are matched to the features from image $j = 1$ to obtain 2D-2D feature correspondences.

With the knowledge of epipolar geometry (see Figure 2.10) and a set of 2D-2D correspondences, the essential matrix of a calibrated camera (known intrinsics) can be estimated in a RANSAC framework. The essential matrix has 3D structure information encoded. Decomposing the essential matrix reveals the relative pose of the second camera with respect to the first. This

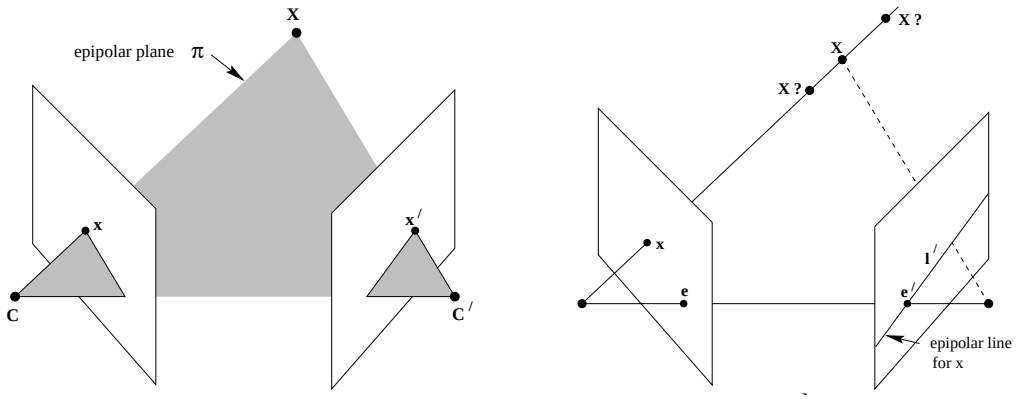


Figure 2.10: Given two cameras $\mathbf{C}$ and $\mathbf{C}'$ and matching features $\mathbf{x}$ and $\mathbf{x}'$ in each image respectively, the epipolar geometry is depicted above. The ray sampled by the point $\mathbf{x}$ corresponds to a line $\mathbf{l}'$ in C′ that passes through the epipole $\mathbf{e}'$. (Image source: Hartley and Zisserman (2004))

gives inital 3D structure to the scene. With this knowledge, features can be back projected and triangulated to obtain 2D-3D correspondences.

Once initial 3D structure has been developed, the subsequent images ($j > 2$) can be iteratively added to the scene. In each iteration, features are detected, described, and matched to the previous features of prior images. This gives 2D-3D correspondences to the features in the current image. Using these correspondences, the camera projection matrix can be estimated, also within a RANSAC framework. This is called the absolute pose, as it is the pose of the camera relative to the scene. New points can be triangulated to add 3D points to the scene. Lastly, a bundle adjustment is used to jointly optimize the 3D points and the poses of all cameras in the scene.

This general framework is adapted to an underwater setting. While there are some fundamental differences for some of the steps, the general framework is the same. Most importantly, standard epipolar geometry becomes invalid in the refractive camera model. Rays in water sampled by a image points in one image do not correspond to lines in the corresponding image, but rather a curves. Instead, the generalized epipolar constraint proposed by Pless (2003) is used. Further, a different method for absolute pose requires a new technique to deal with refraction. These methods are described in further detail in Chapter 3.

3 Methodology

This section outlines the methods used for the SfM pipeline in this project. The general pipeline for the proposed underwater SfM scheme can be seen in Figure 3.1.

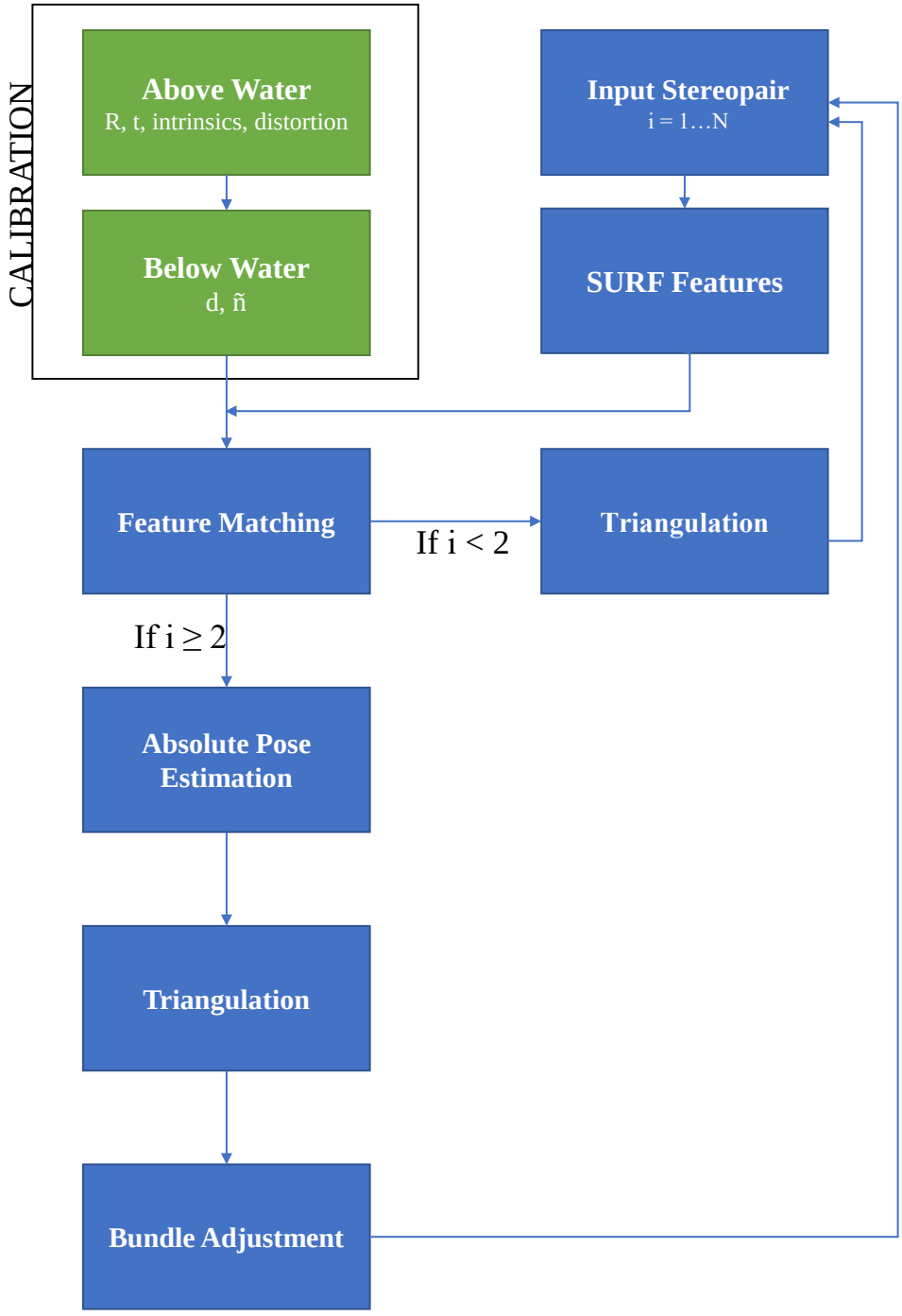


Figure 3.1: The general underwater SfM pipeline. Calibration formulation was done in previous work (Cotugno et al., 2016) and is shown in green. All other modules are formulated and implemented in this work.

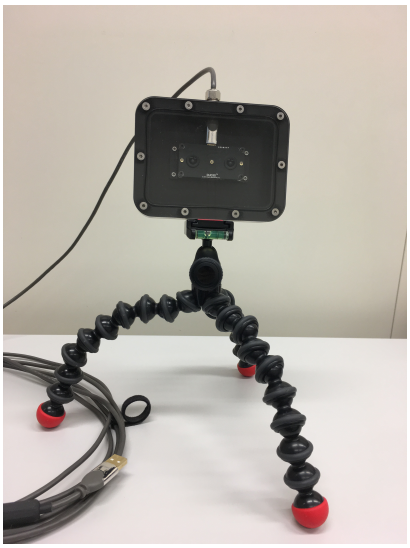


Figure 3.2: The DUO MLX within the custom built housing.

The general pipeline is similar to the one in Figure 2.9. However, in this project, a stereo camera is used. After calibration, the relative pose between the two cameras within a stereopair is known. This information is exploited in this project to effectively eliminate the relative pose estimation module. Therefore, in the first iteration, features are detected and matched within the stereopair and triangulated to obtain initial 2D-3D correspondences. On the next iteration, absolute pose can be estimated with these correspondences. Lastly, because the relative pose within a stereopair is known, the scale of the scene is fixed by the known baseline within the stereopair.

The entire software is implemented in MATLAB and uses many MATLAB toolboxes.

3.1 Equipment & Setup

This project uses a DUO MLX stereo camera sensor from DUO3D. The sensor's dimensions are $52 \times 25 \times 13$ mm with a 30 mm baseline, making the camera very compact. The maximum resolution of the sensor is 752×480 and the sensor images in grayscale. The housing for the camera is custom built with a thin acrylic flat viewing port. The viewing port has a thickness of approximately 3.8 mm. The camera within the housing port can be seen in Figure 3.2. The camera is interfaced via a USB port and through Robot Operating System (ROS) middleware.

The calibration is performed in two stages using the procedure developed in Cotugno et al. (2016). First, an in-air calibration is performed to obtain the camera intrinsics and the relative transformation parameters from the master camera to the slave camera. In-air calibration is performed using Kalibr (Furgale et al., 2013), a calibration toolbox for multi-camera rigs. An april grid (Olson, 2011) is used for the calibration board. Once calibrated in air, the camera is placed in the housing and calibrated underwater to obtain the housing parameters (see: Section 2.2.2). The refractive indices are assumed to be known and fixed. The refractive indices are obtained from Polyanskiy (2008). The values are $\mu_a = 1$, $\mu_g = 1.49$, $\mu_w = 1.33$ for air,

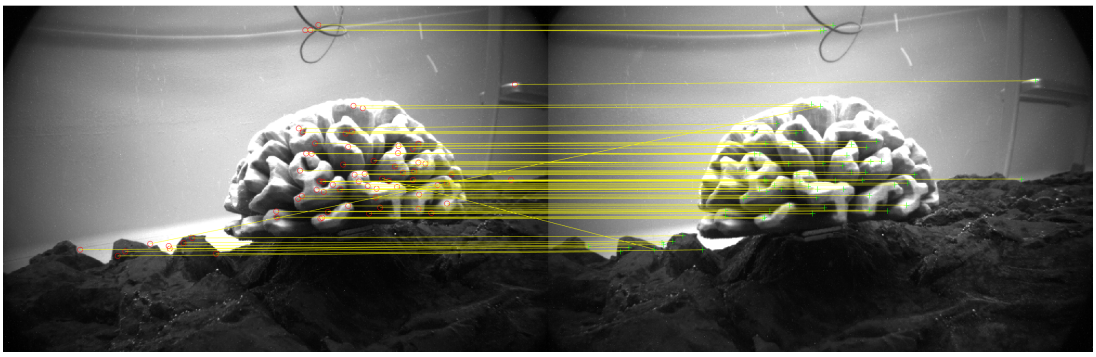


Figure 3.3: This image shows an example of incorrect feature matches within a stereopair. The incorrect matches are those that cross the otherwise straight lines.

acrylic, and fresh water, respectively. Further information on the calibration procedure can be found in Cotugno et al. (2016).

3.2 Feature Detection & Matching

The process for detecting and matching features is described in this section. The basis for many 3D reconstruction techniques, especially SfM is finding matching feature points across images. These feature correspondences are used to estimate the 3D structure of the scene. Therefore, accuracy and robustness of feature detection and description are critical. In particular, with an image sequence of a moving camera (or camera rig), features should be invariant to scale, orientation, and illumination changes.

Two of the most popular methods of this type are the scale-invariant feature transform (SIFT) (Lowe, 1999) and the speeded up robust features (SURF) (Bay et al., 2008). Both methods detect features and describe them and are quite similar in methodology. However, only SURF is implemented in MATLAB and therefore is used as the feature detector and descriptor in this project.

When the first stereopair is input into the system, SURF features are detected in both images. These features are then matched using the MATLAB Computer Vision Toolbox feature matching function. Due to underwater conditions (scarce illumination, particles in the water, no background texture), less features are detected than a standard in air image. In addition, features are described and matched less accurately. Therefore, matching typically returns a limited amount of matches between a stereopair than would be in-air. The standard range of matches seen in the test are between 35 - 100 depending on the view of the scene.

Further, matching often produces incorrect matches. This occurs in-air as well. Because these matches are used to triangulate 3D points, and later to estimate pose of the subsequent stereopairs, it is important that incorrect matches are removed. An example of incorrect matches can be seen in Figure 3.3.

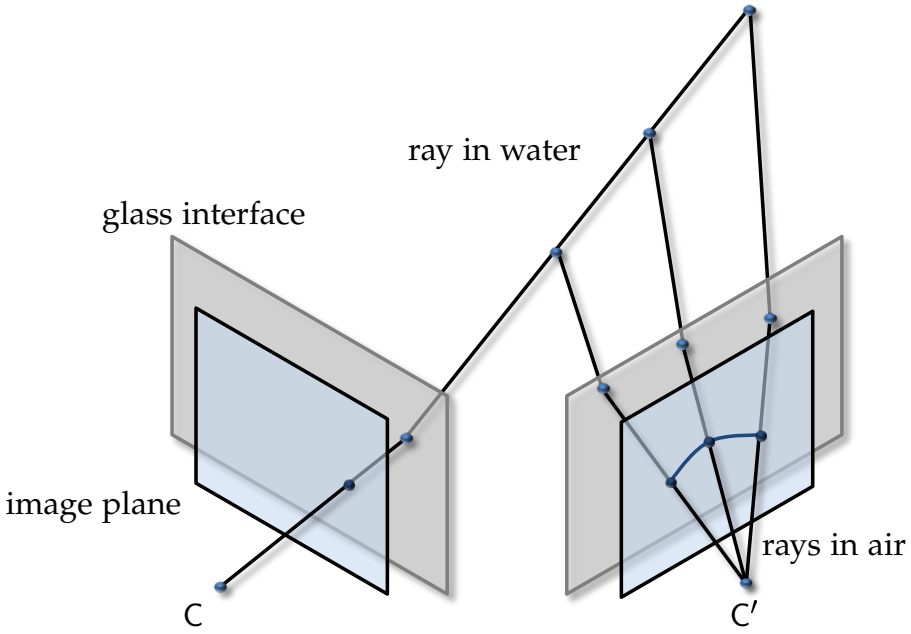


Figure 3.4: Generalized epipolar geometry illustrated with a refractive camera. Corresponding Plücker lines intersect. The figure also shows that in the refractive camera case, a point and its corresponding ray correspond to an epipolar curve as opposed to an epipolar line in the perspective case. (Image source: Jordt (2014))

Typically, in a stereo rig in-air, images can be rectified such that epipolar lines are horizontal (see Figure 2.10). Feature matching can then be done by searching horizontally along the corresponding image. Further, in the case of a monocular camera, feature matches can be evaluated based on the distance of the feature to the epipolar line in the corresponding image (distance from $\mathbf{x}'$ to $\mathbf{l}'$ in Figure 2.10). However, in an underwater refractive camera, standard epipolar geometry is invalid.

In the general camera model proposed by Pless (2003) and described in Section 2.1.2, a generalized essential matrix is formed by Pless that can be used to describe geometry between two general cameras. The generalized essential matrix is used to form the generalized epipolar constraint (GEC). This constraint is based on the theory that the Plücker lines sampled by corresponding features in two images intersect. The equation for the epipolar constraint is shown in Equation 3.4. The theory behind the GEC is visualized in Figure 3.4. This theory can be adapted to the case here with a refractive camera.

For instance, take a pair of matching features $\mathbf{x}$ and $\mathbf{x}'$ from the master camera and slave camera of a stereopair, respectively. Back projecting these features into their respective points on the outer interface and the rays in water gives $\left(\mathbf{X}_s, \tilde{\mathbf{X}}_w\right)$ and $\left(\mathbf{X}'_s, \tilde{\mathbf{X}}'_w\right)$. Transforming the point-ray representation into Plücker coordinates gives

$$\mathbf{L} = \begin{pmatrix} \tilde{\mathbf{X}}_w \\ \tilde{\mathbf{X}}_w \times \mathbf{X}_s \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{X}}_w \\ \mathbf{M} \end{pmatrix} \qquad \mathbf{L}' = \begin{pmatrix} \tilde{\mathbf{X}}'_w \\ \tilde{\mathbf{X}}'_w \times \mathbf{X}'_s \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{X}}'_w \\ \mathbf{M}' \end{pmatrix} \tag{3.1}$$

.

The GEC constraint states that corresponding Plücker lines should intersect. To relate the two, they need to be within the same coordinate frame. If the the Plücker line from $\mathbf{x}$ is in the world coordinate frame, the Plücker line from $\mathbf{x}'$ should be transformed into the world coordinate system by (Pless, 2003)

$${}^{\mathrm{WC}}\mathbf{L}' = \begin{pmatrix} \mathrm{R}\tilde{\mathbf{X}}'_w \\ \mathrm{R}\mathbf{M}' - [\mathbf{C}]_\times \mathrm{R}\tilde{\mathbf{X}}'_w \end{pmatrix} \tag{3.2}$$

where R is the rotation into the world coordinate frame and $[\mathbf{C}]_\times$ is the skew-symmetric matrix representation of the translation into the world coordinate frame.

Once transformed, the intersection between the two Plücker lines can be determined by (Jordt, 2014)

$$\tilde{\mathbf{X}}_w^{\top\mathrm{WC}}\mathbf{M}' + \mathbf{M}^{\top\mathrm{WC}}\tilde{\mathbf{X}}'_w = 0 \tag{3.3}$$

These two equations together form the GEC equation (Jordt, 2014)

$$0 = \mathbf{L}^\top \underbrace{\begin{pmatrix} -[\mathbf{C}]_\times \mathrm{R} & \mathrm{R} \\ \mathrm{R} & 0_{3\times3} \end{pmatrix}}_{\mathrm{E_{GEC}}} \mathbf{L}' \tag{3.4}$$

where the matrix $\mathrm{E_{GEC}}$ is the generalized essential matrix described previously.

In the case of this project, a calibrated stereo rig is used. Therefore, the relative pose between the two cameras (R and $\mathbf{C}$) is known and fixed. This provides a fixed $\mathrm{E_{GEC}}$ for the stereo rig. The $\mathrm{E_{GEC}}$ is then used to remove matches that score above an empirically determined threshold. The threshold used is $5E-4$. However, the threshold can be adjusted accordingly.

3.3 Triangulation

Once correct matches within a stereopair have been determined, they can be triangulated to obtain 3D points in the scene.

In the first iteration of the pipeline, triangulation is only performed using the images of the stereopair. Therefore, for each feature correspondence, only two rays are used to estimate the feature point. However, when more views are added that also contain the same feature correspondence, the rays from the new views can be used as well to estimate the 3D point. Often, more views of a 3D point leads to a better estimate of the 3D point.

Triangulation can be formed based on the following constraint

$$\mathbf{X} = \mathbf{X}_{s_i} + \kappa_i \tilde{\mathbf{X}}_{wi}, \quad \kappa_i \in \mathbb{R} \tag{3.5}$$

where $\mathbf{X}$ is the 3D point associated to a feature point in the i^{th} view represented by its point-ray pair $\left(\mathbf{X}_{s_i}, \tilde{\mathbf{X}}_{w_i}\right)$ and κ_i is the distance from the outer surface point to the 3D point along the ray.

Using this constraint, the formula used to triangulate a 3D point can be formed. The equation is a linear least-squares equation to find the optimal 3D point estimate that fits the rays. The least-squares equation used can be seen in Equation 3.6 (Jordt, 2014).

$$\epsilon = \underbrace{\min}_{\mathbf{X},\kappa_1,\ldots,\kappa_n} \sum_{i=1}^{N} |\mathbf{X}_{s_i} + \kappa_i \tilde{\mathbf{X}}_{wi} - \mathbf{X}|^2 \tag{3.6}$$

The problem in Equation 3.6 can be solved by stacking a linear system of equations $\mathrm{A}\mathbf{x} = \mathbf{b}$ in the fashion

$$\underbrace{\begin{bmatrix} 1 & -\tilde{\mathbf{X}}_{w_1} & 0 & \cdots & 0 \\ \vdots & 0 & -\tilde{\mathbf{X}}_{w_2} & 0 & \vdots \\ & \vdots & 0 & \ddots & \\ 1 & 0 & \cdots & & -\tilde{\mathbf{X}}_{w_N} \end{bmatrix}}_{\mathrm{A}} \underbrace{\begin{bmatrix} \mathbf{X} \\ \kappa_1 \\ \kappa_2 \\ \vdots \\ \kappa_N \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{X}_{s_1} \\ \mathbf{X}_{s_2} \\ \vdots \\ \mathbf{X}_{s_N} \end{bmatrix}}_{\mathbf{b}} \tag{3.7}$$

where $\mathbf{x}$ is the vector of unknowns and $\mathbf{b}$ is the vector of observations. In this case, the additional unknowns κ_i are auxiliary unknowns and not used further in the SfM pipeline.

The system of equations is solved in MATLAB using the backslash operator. The solution is then in the form of $\mathbf{x} = \mathrm{A}\backslash\mathbf{b}$.

3.4 Absolute Pose Estimation

Once 2D-3D correspondences have been determined within the first stereopair via triangulation, the next stereopair can be input to add to the scene structure. Absolute pose estimation involves estimating the pose of a camera relative to an established 3D representation of a scene. Since the scene has been established with the master camera of the first stereo pair at the scene origin, with a second camera (slave camera) and some initial 3D triangulated points, there exists enough information to estimate the pose of the next stereopair. The pose of the camera consists of 6 degrees of freedom (DOF): 3 for the rotation, and 3 for the translation.

The absolute pose problem is also known as the PnP (perspective-n-point) problem in the perspective case. There are many well-known solutions to solving the problem. Minimal solvers for the the problem are the case where only 3 points are needed (called the $P3P$ problem).

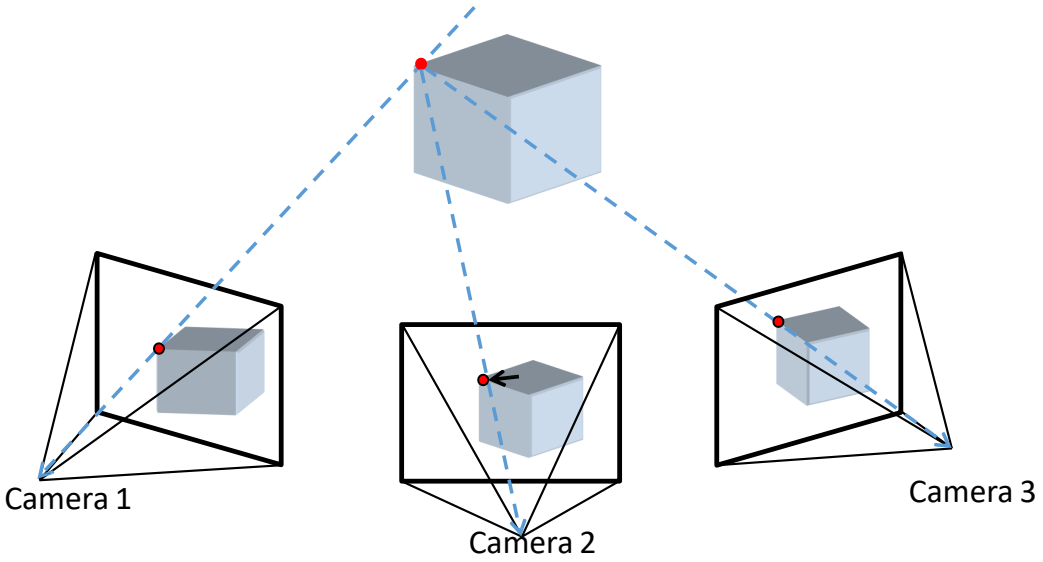


Figure 3.5: Example of persepctive triangulation from $N = 3$ views. In the refractive case, the point-ray representation of each pixel is used to triangulate a 3D point. (Image source: Geiger (2016))

Well-known solutions are in Fischler and Bolles (1981) and Kneip et al. (2011).

In the case of a general camera, the problem is known as the $gPnP$ (general PnP) problem. Minimal solutions for this problem using 3 points can be found in Nistér (2004) and Lee et al. (2016). There exists little literature on absolute pose for a refractive camera other than Haner and Åström (2015), where only one refractive plane is used. Other solutions, including an iterative solution and a solution based off the process used for calibration in Agrawal et al. (2012) can be found in Jordt (2014). Additionally, an analysis of multiple different methods was performed in Jordt (2014). The conclusion of the analysis was that the methods by Nistér (2004) and the adaptation of Agrawal et al. (2012) were – while minimal in the case of Nister – very sensitive to noise.

Thus Jordt (2014) proposes to use an iterative approach to solve for the absolute pose based on known 2D-3D correspondences. The iterative approach requires $c \geq 3$ correspondences. However, due to noise in image observations, $c = 7$ correspondences are used (Jordt et al., 2016). This iterative solution is used within a RANSAC framework to obtain an initial solution and remove outliers. Lastly, the initial solution obtained within the RANSAC framework is nonlinearly optimized in a Levenberg-Marquardt optimization scheme.

This method was attempted for this system. However, in the case where very few correspondences exits, this solution was not providing an accurate enough initial solutions for Levenberg-Marquardt. Thus, a new solution is proposed.

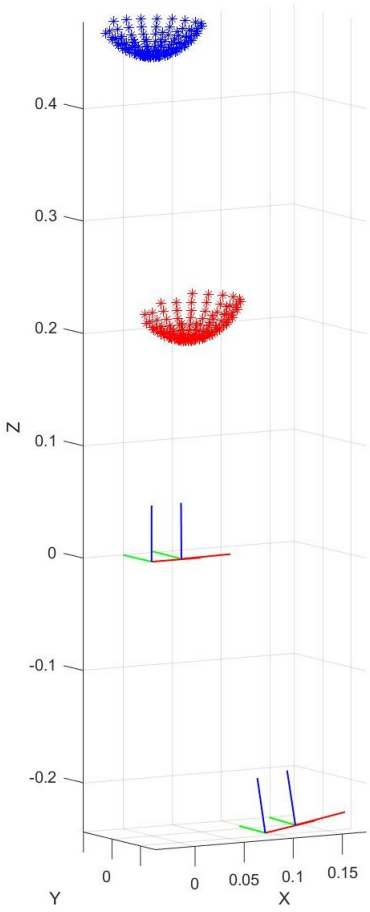


Figure 3.6: An example of 3D-3D correspondences. The first stereopair, considered at the origin of the world coordinate frame, has triangulated points to obtain 3D points in the world coordinate frame, shown in blue. The next stereopair triangulates matches to the previous stereopair to obtain matches in the local coordinate frame, shown in red. The difference between world coordinate frame and local camera frame is exaggerated for effect.

3.5 Initial Solution

The new solution again exploits the stereopair configuration used in this project. Because relative pose within a stereopair is known, local camera coordinate frame 3D points can be triangulated within any stereopair view. Therefore, when adding the next stereopair to the scene, features matched to the scene can be triangulated within the new stereopair to obtain local camera coordinate frame 3D points. These 3D points correspond to 3D points already within the world coordinate frame of the scene. These correspondences will be referred to as 3D-3D correspondences. This concept is illustrated in Figure 3.6.

Once 3D-3D correspondences have been established, they can be used to estimate the transformation parameters between the two point sets. This transformation is then pose of the new stereopair in the scene. Solving for the transformation is essentially a point cloud registration

problem. In Lorusso et al. (1995) four different methods for solving transformation parameters between two 3D point sets are analyzed: using singular value decomposition (SVD), using orthonomral matrices, using unit quaternions, and using dual quaternions. They conclude that using the SVD method, further described in Umeyama (1991), provides the best overall accuracy and stability. Therefore, this method is used in this project.

The SVD method finds the transformation parameters based on the similarity of the 3D-3D correspondences when translated to their centroids. The method is described as follows. First, the centroid for each 3D point set $\mathbf{X}_1$ and $\mathbf{X}_2$ is computed by

$$\overline{\mathbf{X}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{X}_i \tag{3.8}$$

.
where N is the number of corresponding points. The points are then translated to its centroid by

$$\tilde{\mathbf{X}} = \mathbf{X} - \overline{\mathbf{X}} \tag{3.9}$$

.
Next, the correlation between the two translated point sets $\tilde{\mathbf{X}}_1$ and $\tilde{\mathbf{X}}_2$ is computed by

$$\mathrm{H} = \frac{1}{N}\tilde{\mathbf{X}}_2\tilde{\mathbf{X}}_2^{\top} \tag{3.10}$$

.
The 3×3 matrix H is then decomposed using SVD to obtain

$$\mathrm{SVD}(\mathrm{H}) = \mathrm{U}\Lambda\mathrm{V}^{\top} \tag{3.11}$$

where the optimal rotation can be extracted by

$$\hat{\mathrm{R}} = \mathrm{UV} \tag{3.12}$$

However, it is possible that, in the case of planar data sets or large noise, a reflection can be found as opposed to a rotation matrix. A reflection occurs when $\det(\hat{\mathrm{R}}) = -1$. To deal with this, Umeyama (1991) sets the third column of V negative. Thus, the optimal rotation is then found by

$$\hat{\mathrm{R}} = \mathrm{U}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & -1\end{bmatrix}\mathrm{V} \tag{3.13}$$

Finally, the optimal translation can be found by

$$\hat{\mathbf{t}} = \overline{\mathbf{X}}_2 - \hat{\mathrm{R}}\overline{\mathbf{X}}_1 \tag{3.14}$$

This method provides the initial of the absolute pose of the new stereopair. However, there are often outliers in matched features across stereopairs. To deal with this, this method is wrapped in a RANSAC framework to remove outliers and to find the best-fitting transformation parameters.

RANSAC requires an evaluation function of the model fit within each RANSAC iteration to provide the best solution. Typically this function is the reprojection error. However, as has already been discussed, the projection of a 3D point into the image plane involves solving a 12^{th} degree polynomial. This inefficiency would make RANSAC, the subsequent optimization, and especially bundle adjustment very costly. Thus, a different error function should be sought.

In Jordt (2014) two error functions are proposed, the *virtual camera error* (briefly mentioned previously) and the *angular error*. The virtual camera error finds a perspective camera for each pixel to compute the error in. The angular error is the error between a ray from the corresponding point on the outer interface $\mathbf{X}_s$ to a 3D point transformed into the local coordinate frame, and the corresponding point's ray in water $\tilde{\mathbf{X}}_w$. Jordt (2014) concludes that the virtual camera error performs the best. Therefore, the virtual camera error is used throughout this project, and will be described in further detail.

3.5.1 Virtual Camera Error

As stated previously, the virtual camera error uses the concept of creating a virtual perspective camera at every pixel. This allows for a computationally efficient perspective projection of 3D points into the virtual camera where the error can be evaluated, analogous to the reprojection error. This section will go into detail in how the virtual camera is defined and how the error can be computed.

The virtual camera can be formed for any pixel if the housing parameters (see: Section 2.2.2) are known. The virtual optical axis is parallel to the interface normal $\tilde{\mathbf{n}}$ and goes through the real camera center. Therefore, the virtual image plane is parallel to the refractive interface. The concept of the virtual camera can be seen in Figure 3.7.

To transform into the virtual camera, a rotation and translation need to be formed. The rotation R_v is defined by a rotation axis and a rotation angle. The rotation axis is computed by using Equation 3.15a. It is the cross product between the interface normal and the real optical axis. The rotation angle is formed using Equation 3.15b. It is the dot product between the interface normal and the real optical axis.

$$\mathrm{R}_{v_{\text{axis}}} = \tilde{\mathbf{n}} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.15a}$$

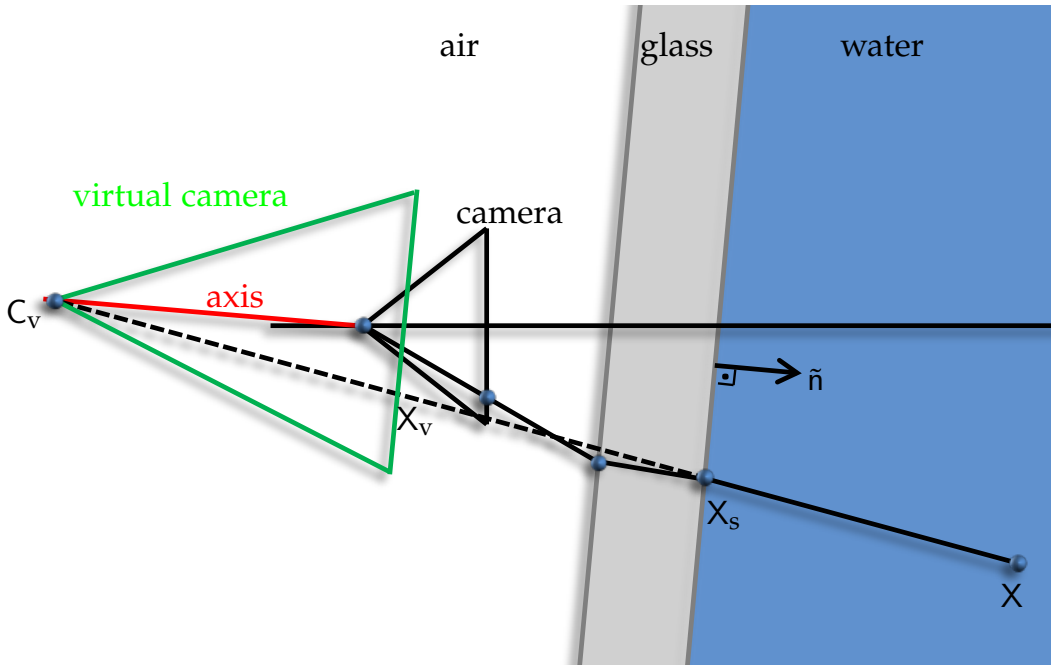


Figure 3.7: Illustration of the virtual camera. (Adapted from: Jordt et al. (2016))

$$\mathrm{R}_{v_\theta} = \arccos \tilde{\mathbf{n}} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.15b}$$

The center of the virtual camera $\mathbf{C}_v$ is the intersection of the incoming ray from water and the virtual camera axis.

Once the virtual camera transformation parameters have been defined, the error can be computed. A 2D-3D correspondence must be known to compute the error. Both the 2D image point and 3D world coordinate frame correspondence must be transformed into the virtual camera. This procedure is outlined as follows.

First, the 3D world point is transformed into the local coordinate frame of the camera.

$$^{\mathrm{CC}}\mathbf{X} = \mathrm{R}^{\top\,\mathrm{WC}}\mathbf{X} - \mathrm{R}^{\top}\mathbf{C} \tag{3.16}$$

Then, the 3D point in the camera coordinate system is transformed into the virtual coordinate system by

$$\mathbf{X}_v = \mathrm{R}_v^{\top\,\mathrm{CC}}\mathbf{X} - \mathrm{R}_v^{\top}\mathbf{C} \tag{3.17}$$

where $\mathbf{X}_v$ is the 3D point represented in the virtual coordinate system.

Similarly, the image observation must be transformed into the virtual coordinate frame. The image observation is first back projected to its point-ray representation $\left(\mathbf{X}_s, \tilde{\mathbf{X}}_w\right)$. This is already in the camera coordinate system, so no intermediate transformation is necessary. The point on the outer interface is used as the image point representation and transformed into the virtual coordinate system by

$$\mathbf{X}_{vs} = \mathrm{R}_v^\top \mathbf{X}_s - \mathrm{R}_v^\top \mathbf{C} \tag{3.18}$$

where $\mathbf{X}_{vs}$ is the image observation representation in the virtual coordinate system.

The error can be computed once both points of the 2D-3D correspondence has been transformed into the virtual coordinate frame. To compute the error, both points are perspectively projected into the virtual image plane. As seen in Equation 2.1, perspective projection requires a known focal length. The focal length of the virtual camera is the distance from the camera optical center to the outer interface computed as

$$f_v = d_0 + d_1 \tag{3.19}$$

where d_0 and d_1 are defined in Section 2.2.2 and f_v is the virtual focal length. Thus the virtual camera error is computed by (Jordt, 2014)

$$\mathbf{g}_v = \begin{pmatrix} f_v \frac{\mathbf{X}_{v_X}}{\mathbf{X}_{v_Z}} - f_v \frac{\mathbf{X}_{vs_X}}{\mathbf{X}_{vs_Z}} \\ f_v \frac{\mathbf{X}_{v_Y}}{\mathbf{X}_{v_Z}} - f_v \frac{\mathbf{X}_{vs_Y}}{\mathbf{X}_{vs_Z}} \end{pmatrix} \tag{3.20}$$

where the subscripts X, Y, Z refer to the coordinate in each dimension and $\mathbf{g}_v$ is the vector of error in the virtual camera.

Lastly, the total error for k correspondences can be computed by Equation 3.21.

$$E_v = \sum_{k=1}^{K} \|\mathbf{g}_{v_k}\| \tag{3.21}$$

There are two main advantages to using the virtual camera error. First, the error computation is efficient as it only requires three Euclidian transformations and two perspective projections. This is much more efficient than solving a 12^{th} degree polynomial for each 3D point. Second, many of the values can be precomputed prior to optimization or to RANSAC. Since only the housing parameters and the image observations are needed, the virtual coordinate system transformation $(\mathrm{R}_v, \mathbf{C}_v)$ and the image coordinate in the virtual image plane $\mathbf{X}_{vs}$ can be precomputed.

Once the initial solution has been solved for and inliers have been determined within the RANSAC (Fischler and Bolles, 1981) framework, the solution is optimized within a non-linear Levenberg-Marquardt optimization scheme using all of the inliers. This optimization is done to combat with the typical noise levels of image observations in underwater images. Thus, initial

solutions are optimized such that they are more accurate when input into the bundle adjustment. Optimization is done using the *lsqnonlin* MATLAB function.

3.6 Bundle Adjustment

Bundle adjustment jointly optimizes all poses and 3D points of a scene. It is a non-linear optimization problem, typically solved using the Levenberg-Marquardt algorithm. This section describes how bundle adjusment is formed in this work.

As stated in the previous section, the reprojection error is very inefficient to optimize over. This is particularly true in bundle adjustment where all 3D points and poses are optimized. Thus, the virtual camera error is used as the objective function for the Levenberg-Marquardt optimization.

The unknowns of the optimization are the 3D points and the rig poses. The observations are the image points. The vector of unknowns is then

$$\mathbf{x} = \begin{bmatrix} \mathbf{X}_1, & \cdots, & \mathbf{X}_K, & \mathrm{R}_1 & \mathbf{C}_1, & \cdots, & \mathrm{R}_N & \mathbf{C}_N \end{bmatrix}^\top \tag{3.22}$$

where $\mathbf{X}_k$, $k = \{1, ..., K\}$ are the K 3D points, and $(\mathrm{R}_i, \mathbf{C}_i)$, $i = \{1, ..., N\}$ are the N rig poses. Rig poses are estimated and optimized based on the pose of the master camera. Errors in the slave camera are computed using the known relative pose from the master to slave camera. Therefore all unknowns are encoded in the 3D point in the virtual camera coordinate frame, $\mathbf{X}_v$ (see: Equation 3.17).

A rotation matrix has 9 parameters, but only 3 degrees of freedom (DOF). Therefore, to minimize the parameters to be optimized, a reparameterization is performed. Rotations are often parameterized using Euler angles. Euler angles are convenient but suffer from ambiquities as a rotation can be represented in more than one way (Terzakis et al., 2014). Quaternions are often used as a more robust parameterization of rotations. However, quaternions must be constrained to unit length during optimization to represent a true rotation. Thus, to avoid using Lagrangian multipliers for the unit length constraint of quaternions, an axis-angle parameterization is used. The axis-angle representation of a rotation is a 3-vector where the normalized vector is the axis to be rotated about and the norm of the vector is magnitude of the rotation angle. The axis-angle representation is a unique solution for every rotation

The known observations are the image points which are parameterized by their virtual camera coordinates. Thus the vector of observations is the vector of virtual camera coordinates $\mathbf{X}_{vs_{k_i}}$ of each k 3D point seen in rig view i.

Once formulated, bundle adjustment is performed using the *lsqnonlin* MATLAB function after each new pose is estimated. While analytical derivatives for the virtual camera error function exist, numerical derivates are used in this project.

4 Results & Discussion

In this chapter, the results of the developed system are displayed and discussed. The chapter begins with intermediate results from the feature matching module. Next, an explanation of data simulation to test the remaining modules is described. Then, results of the absolute pose estimation on the simulated data are displayed and discussed. Lastly, results are displayed and discussed for the whole system on both simulated data and real data.

4.1 Feature Matching

This section shows the results of using the GEC to remove poor feature matches within a stereopair. Features are removed if the GEC error (see: Equation 3.4) of a feature matches is larger than a defined threshold. As stated previously, the threshold used was determined empirically and is currently set at $5E-4$.

In Figure 4.1 an example of the results of using the GEC for removing bad matches is shown. In this specific case, all of the incorrect matches are removed.

4.2 Data Simulation

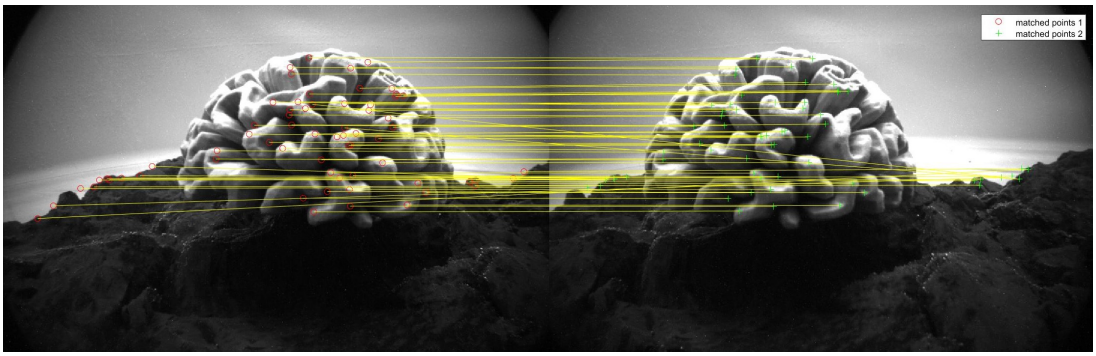
To test different parts of the system, data was simulated. Data was simulated to somewhat resemble the real data scene consisting of a coral with a spheroid shape.

First, a sphere is sampled to create 3D world space points. As in the proposed system, the master camera of the first stereopair is placed in the center of the world coordinate frame. New stereopairs are added by sampling a trajectory that moves in a circular motion around the sphere, keeping the Y axis constant. Stereopairs are sampled at approximately 6 cm, measured from the location of the master camera center. This baseline was chosen because the stereopair baseline is approximately 3 cm, and thus, an image is taken at approximately every 3 cm.

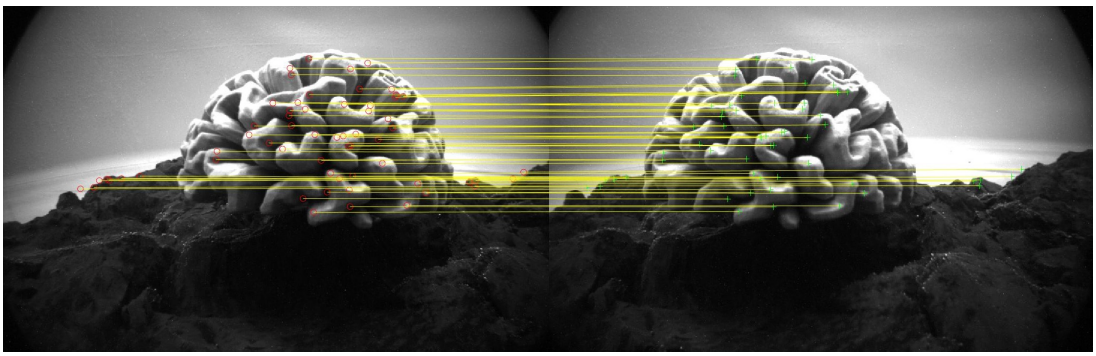
Then, 3D points are forward projected into each view. To do this, 3D points are first checked if they are valid within a viewpoint. There are two validity checks for a viewpoint. First, the angle between the viewing ray and the normal of the 3D point is computed. If the angle is greater than 80° it is considered invalid. Remaining valid points are then forward projected into each image. If the projected points are outside the dimensions of the image, they are considered invalid.

Since the indicies of the projected 3D points are kept, then the true matches between each image is known. The proposed system is then tested using the known true matches and their true location within each image.

Two sets of simulated data are formed. One set with a small sphere and one with a large sphere. In the smaller sphere, image observations are concentrated around the center of the image. Conversely, in the larger sphere, image observations are spread throughout the image. These two settings allow to test the system with features in a limited area in the image and how



(a) Initial Feature Matches



(b) Feature Matches After GEC

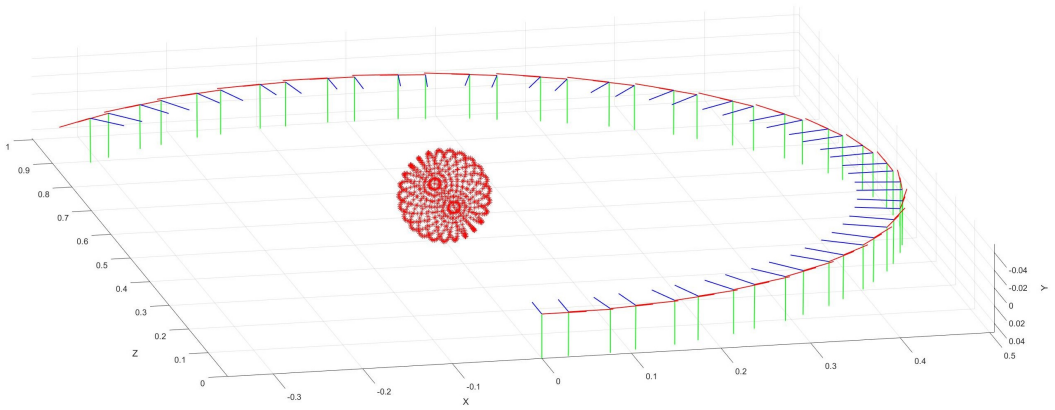
Figure 4.1: Results after using the GEC to remove bad matches within a stereopair. Initial feature matches shown in (a) and cleaned matches shown in (b).

it compares to when features are more thoroughly distributed in the image. The small sphere data set will be referred to as the small set and the large sphere as the large set.

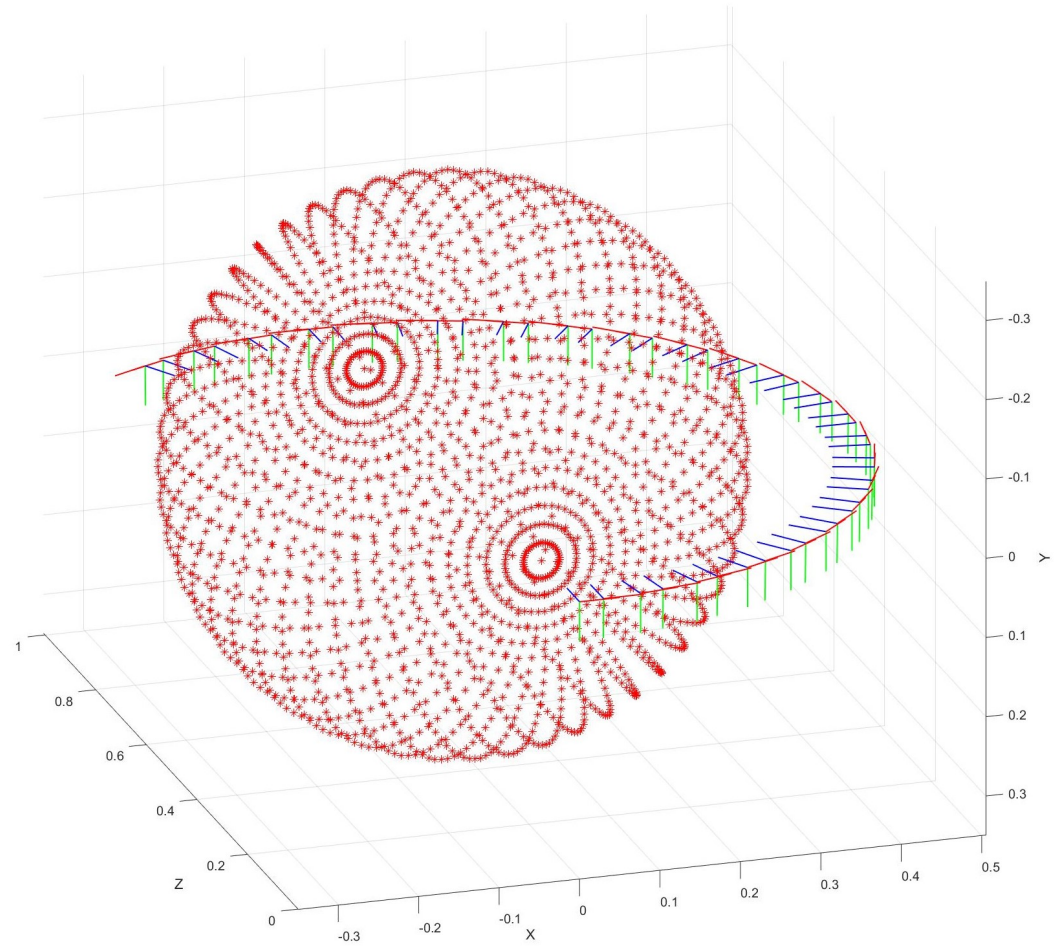
The simulated data can be seen in Figure 4.2 and the parameters of the simulated data can be seen in Table 4.1. Further, an example of the image coverage from both simulated data sets can be seen in Figure 4.3.

Table 4.1: Parameters of the two simualted data sets. Min distance is the minimum distance from the camera center to a 3D point.

	Radius	**Min Distance**	**# of 3D Points**
Small Set	5 cm	45 cm	576
Large Set	35 cm	15 cm	2401

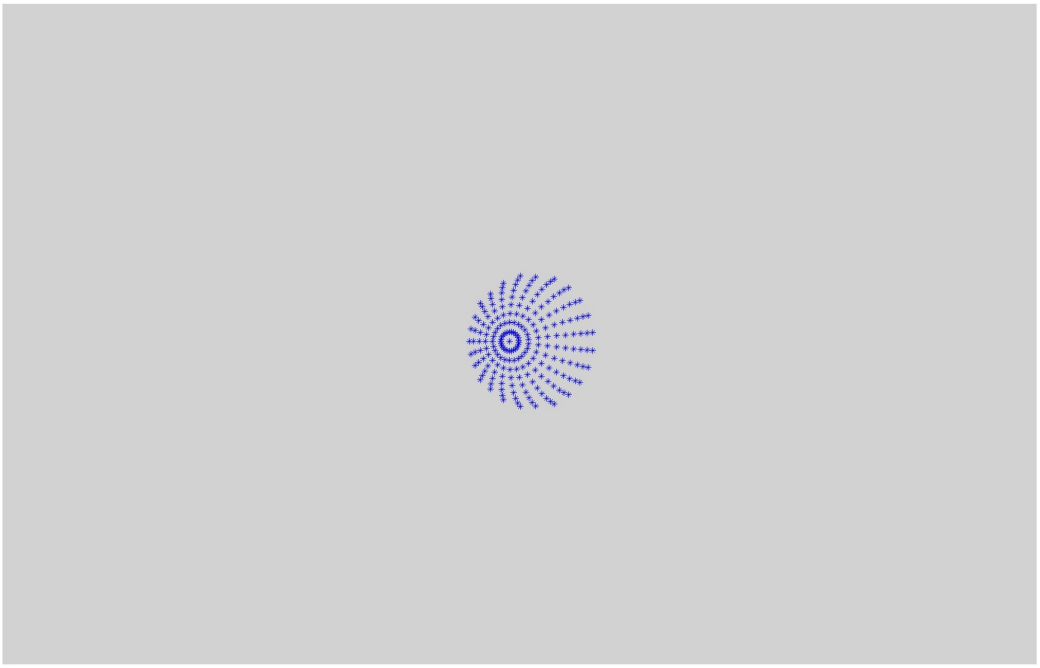


(a)

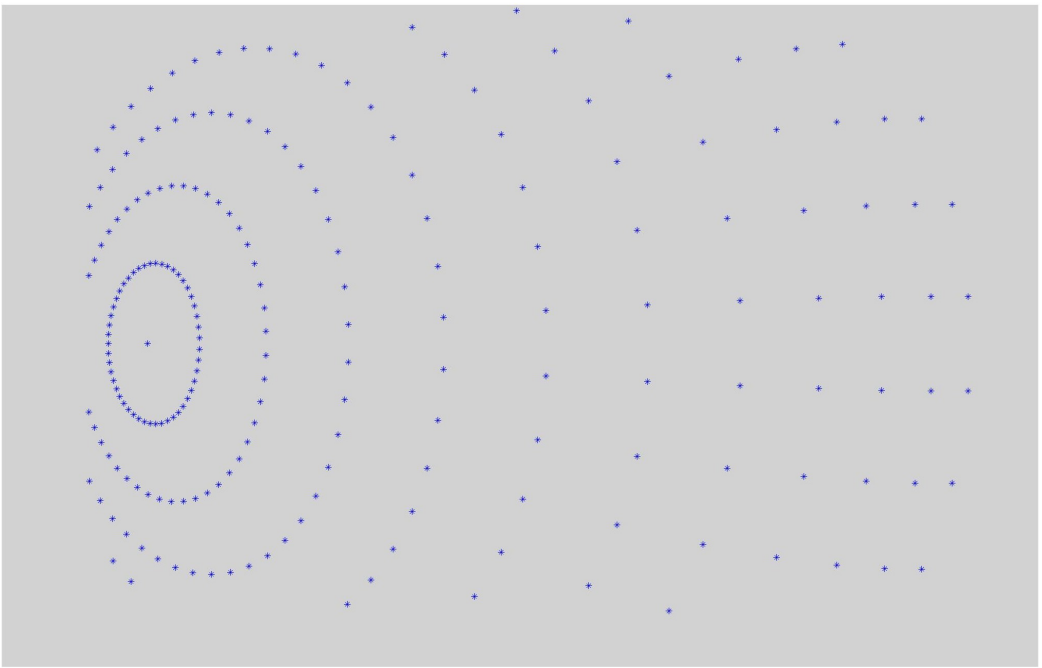


(b)

Figure 4.2: Simulated data sets for the small sphere (a) and large sphere (b). The sphere in the middle are the 3D world space coordinates and each axis triple represents a camera. Blue is the Z axis, green is the Y axis, and red is the X axis of each camera.



(a)



(b)

Figure 4.3: Example image coverage of the simulated data sets for the small sphere (a) and large sphere (b).

4.3 Absolute Pose

In this section, results of absolute pose estimation are displayed and discussed. The absolute pose estimation is tested on both the large sphere and small sphere data sets. Image observations are manipulated with 6 different levels of normal noise, $\sigma = \{0, .2, .4, .6, .8, 1\}$.

Each simulated data set is tested on three different methods to compute the absolute pose. First, absolute pose is evaluated just using the initial pose estimate. Second, absolute pose estimation is evaluated using the initial estimates from the previous method, but optimized in a Levenberg-Marquardt optimization scheme. On the last method, the initial pose and the optimization are wrapped in a RANSAC framework. However, in the last evaluation, outliers are added to the image measurements. Outlier representation is 10% for both small and large sphere test sets. In all methods, 50 tests are run and errors are averaged over all 50 tests.

4.3.1 Initial Estimate

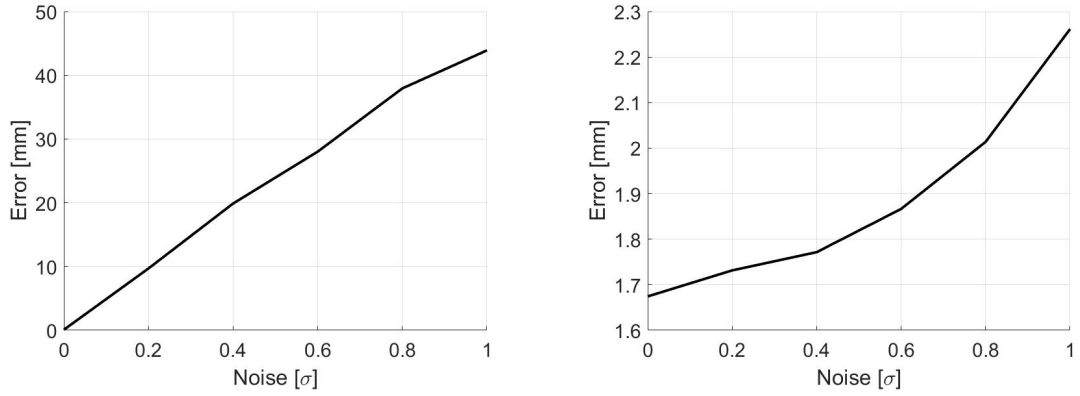
First, results on both data sets using the initial estimate are displayed. Translation errors are in Figure 4.4. Rotation errors are in Figure 4.5. Reprojection errors are in Figure 4.6. Additionally, errors with $\sigma = 0$ are displayed in Table 4.2.

The first thing to note are initial errors with $\sigma = 0$. In the small data set, initial errors are quite small. In the large data set, initial errors are much larger. Typically, the case for $\sigma = 0$, errors should be very close 0. There are a few different reasons for this. One of the main causes of the large discrepancy between the two, and the considerable offset in the large data set, is due to how errors are distributed over the image.

For instance, take the first stereopair with $\sigma = 0$. Triangulating matches and reprojecting them back into the image already contains errors correlated to their location. This effect is visualized in Figure 4.7. In the large data set where features are well distributed within the image, there are larger errors moving away from the center. In the small data set, because most features are clustered near the center, there is little error. These implicit errors can be due to a few different causes. First, and most importantly, there are numerical issues involved with solving the forward projection of 3D points into the image (see: Section 2.2.2) where a $12th$ degree polynomial needs to be solved. Further, these implicit errors propagate into triangulation. Triangulation within a stereopair is sensitive to errors within correspondences due to the narrow baseline within the stereopair. Both errors combined contribute to these offset errors when no noise exists.

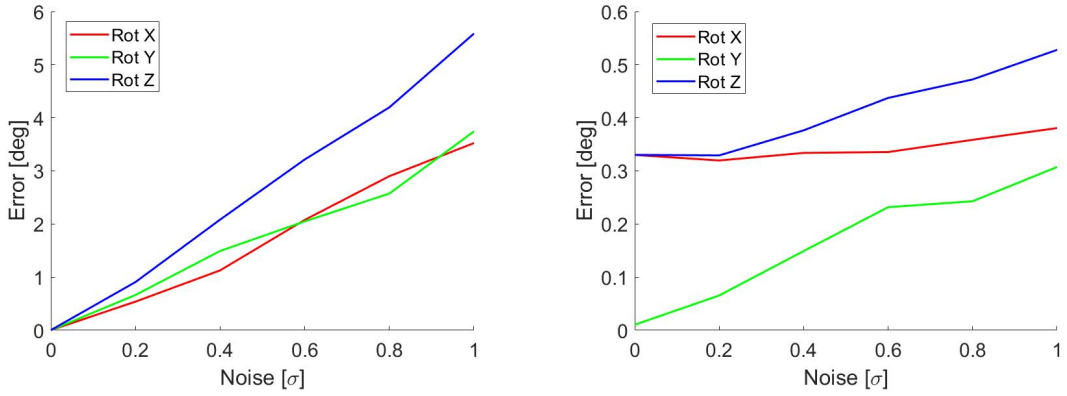
Table 4.2: Errors on both data sets when $\sigma = 0$ using the initial estimate only. Errors are in mm, degrees, and pixels for translation, rotation, and reprojection, respectively.

	Translation	**Rot X**	**Rot Y**	**Rot Z**	**RE**
Small Set	0.08	0.003	0.001	0.002	0.031
Large Set	1.67	0.330	0.010	0.330	1.67



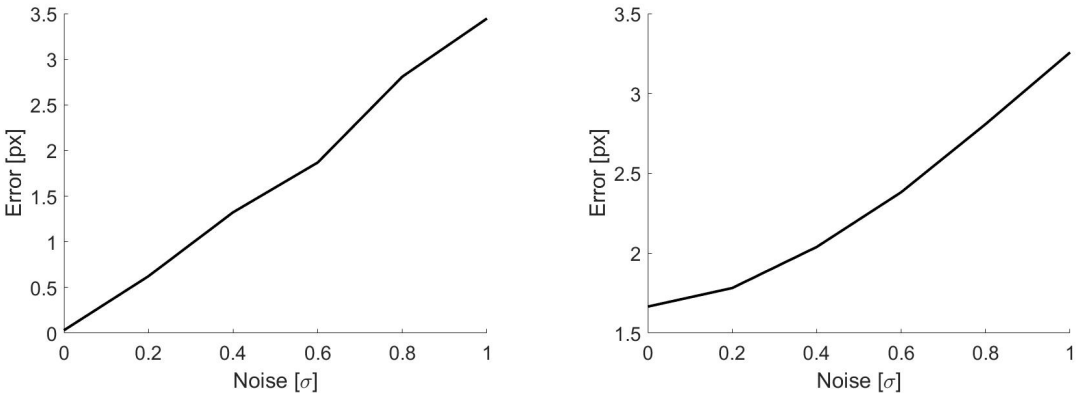
(a) (b)

Figure 4.4: Translation errors on the small (a) and large (b) data sets on the initial estimate.



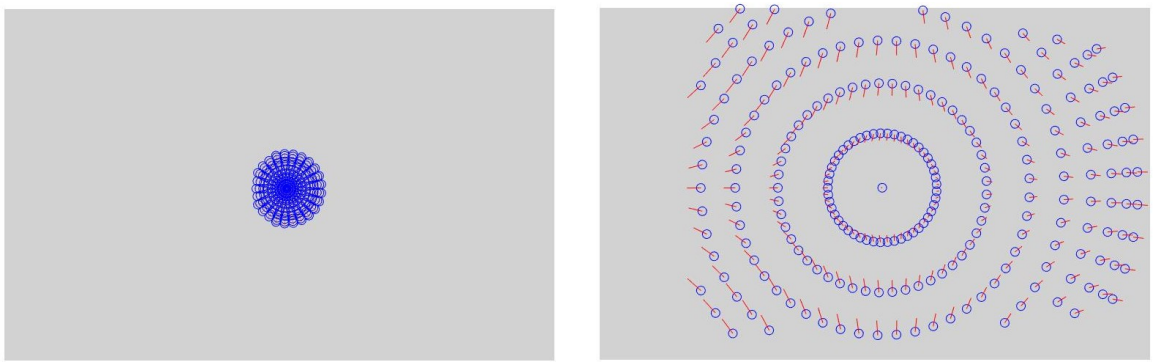
(a) (b)

Figure 4.5: Rotation errors on the small (a) and large (b) data sets on the initial estimate.



(a) (b)

Figure 4.6: Average reprojection errors on the small (a) and large (b) data sets on the initial estimate.



(a) (b)

Figure 4.7: Reprojection error vectors on points in the small (a) and large (b) data sets. Original points are the blue circles and their reprojection error vectors are in red. Error vectors are scaled by 50 for visualization.

The absolute pose estimation method used is a least-squares estimation (Umeyama, 1991). Therefore, the initial estimate is the optimal transformation parameters between the two 3D point sets. Since there is much less error in the small set, the method estimates a more accurate solution when there is no noise. Conversely, the initial pose estimation on the large set has larger errors. Therefore, the fit is less accurate.

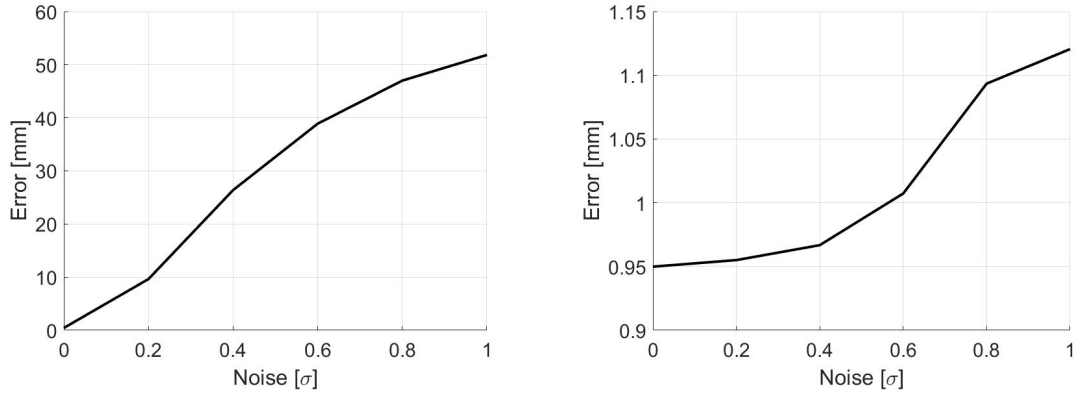
However, pose estimation using points thoroughly distributed throughout the image is more robust to noise in the image observation. Figures 4.4 and 4.5 show that, at $\sigma = 1$, the large sphere data set estimates a much more accurate solution: 2.3 mm in the large set and 45 mm in the small set. Further, the range in errors between $\sigma = 0$ and $\sigma = 1$ are much smaller in the large sphere data set. This is in agreement with the theory that using features well distributed throughout the image leads to better pose estimation. The reason for this is because viewing geometry between the two stereopairs is better suited for pose estimation when matching rays covers more of the image.

4.3.2 Optimization

The initial estimates in the previous section are then optimized over the virtual camera error in a Levenberg-Marquardt optimization scheme. The results for the optimized absolute pose estimates are displayed in this section. Translation errors are in Figure 4.8. Rotation errors are in Figure 4.9. Reprojection errors are in Figure 4.10. Additionally, errors with $\sigma = 0$ are displayed in Table 4.3.

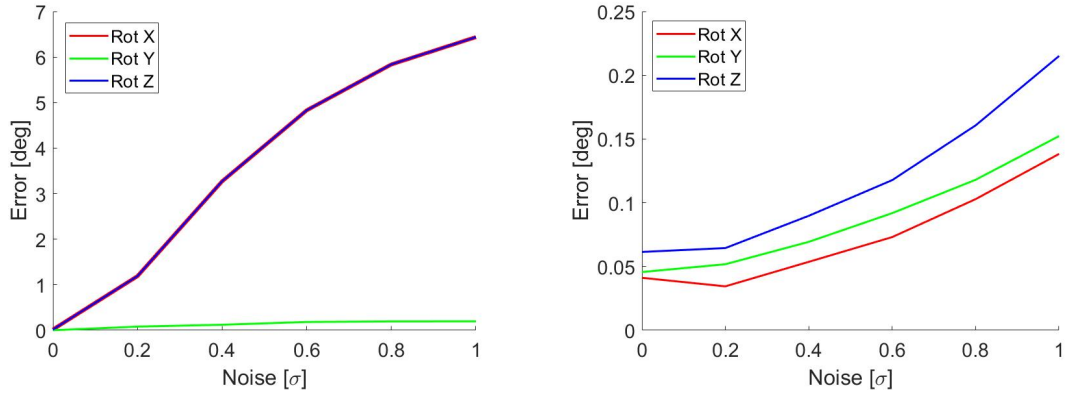
Table 4.3: Errors on both data sets when $\sigma = 0$ with optimization. Errors are in mm, degrees, and pixels for translation, rotation, and reprojection, respectively.

	Translation	**Rot X**	**Rot Y**	**Rot Z**	**RE**
Small Set	0.44	0.015	0.001	0.015	0.016
Large Set	0.95	0.041	0.046	0.061	0.44



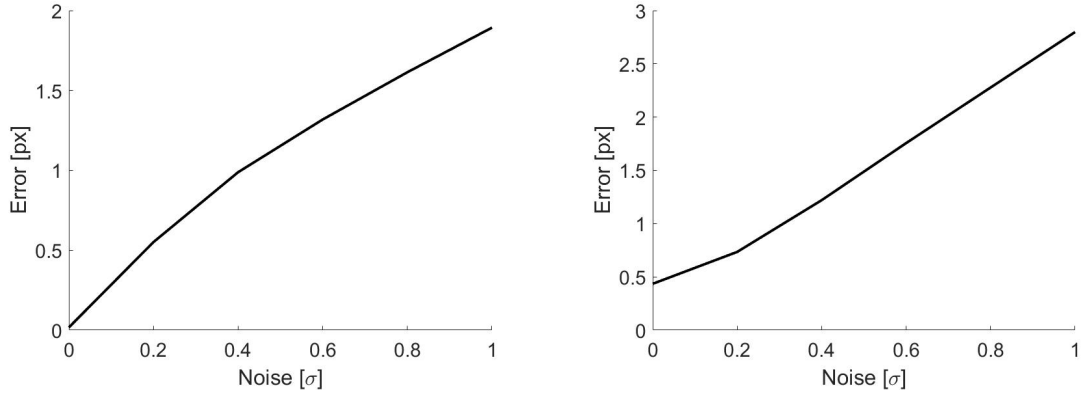
(a) (b)

Figure 4.8: Translation errors on the small (a) and large (b) sphere data sets with optimization.



(a) (b)

Figure 4.9: Rotation errors on the small (a) and large (b) sphere data sets with optimization.



(a) (b)

Figure 4.10: Average reprojection errors on the small (a) and large (b) sphere data sets with optimization.

Again, looking at errors with $\sigma = 0$, there are still initial errors. In the large set, accuracy increases slightly. However, in the small set, accuracy decreases. This could be due to the optimization scheme converging to an incorrect local minimum. The slight decrease (.031 px to .016 px) in reprojection error could support this claim.

Looking further over varying noise levels, Figures 4.8(b) and 4.9(b) show that optimization increases accuracy when compared to the initial estimate results. This is the expected result. When comparing the ranges of the the errors (difference from $\sigma = 0$ and $\sigma = 1$) between the initial estimate and the optimized estimates on the large set, it appears that optimizing the initial estimate results in a more robust solution. The same cannot be said for the small set. It appears that the optimization algorithm fixes the Y axis and moves the other two axes to converge to a solution. Thus, for bad image coverage, the optimization scheme is not robust to noise.

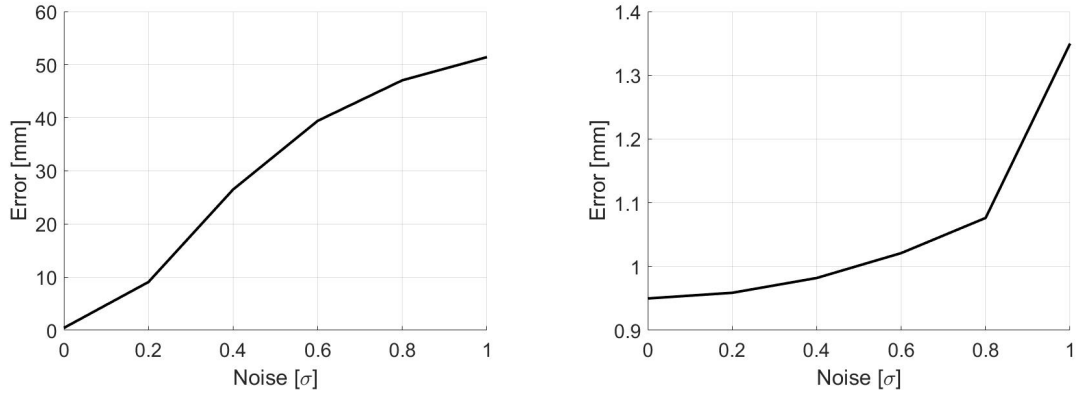
Lastly, as expected, reprojection error decreases in both sets. This is to be expected because the optimization is over the virtual camera error. The most significant decrease is seen in the $\sigma = 0$ case for the large data set where reprojection error decreases from 1.67 px to .44 px after optimization.

4.3.3 With RANSAC

Lastly, errors are displayed with the optimized initial estimates within a RANSAC framework. However, in this section, outliers are introduced into feature matches across the stereopair. Outliers are not introduced within a stereopair as they are removed using the GEC (see Section 3.2).

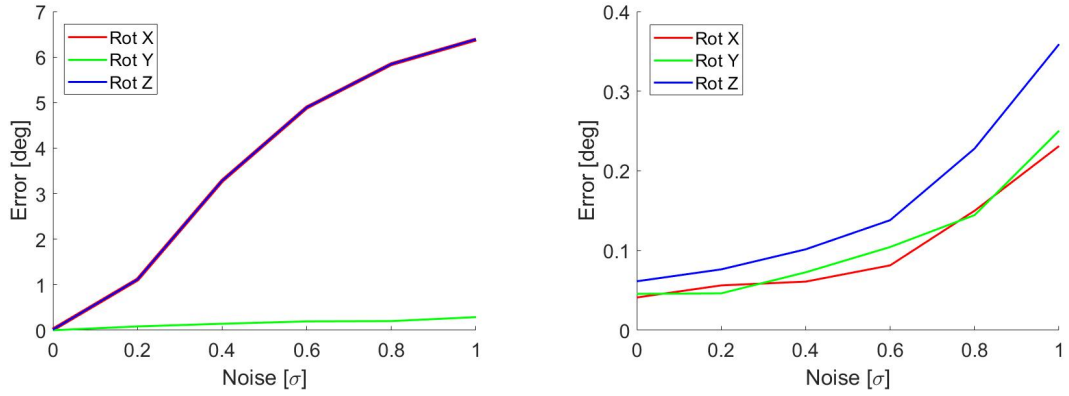
Table 4.4: Errors on both data sets when $\sigma = 0$ with RANSAC. Errors are in mm, degrees, and pixels for translation, rotation, and reprojection, respectively.

	Translation	**Rot X**	**Rot Y**	**Rot Z**	**RE**
Small Set	0.44	0.015	0.001	0.015	0.02
Large Set	0.95	0.041	0.046	0.061	0.44



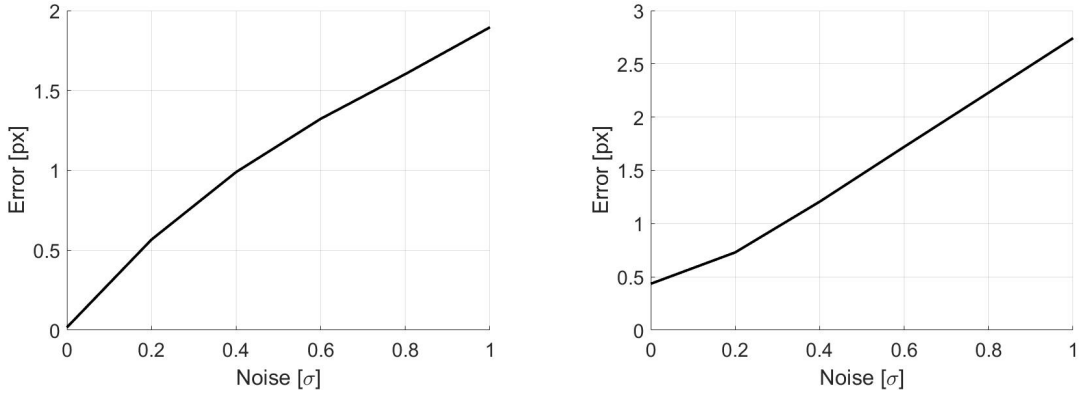
(a) (b)

Figure 4.11: Translation errors on the small (a) and large (b) sphere data sets with RANSAC.



(a) (b)

Figure 4.12: Rotation errors on the small (a) and large (b) sphere data sets with RANSAC.



(a) (b)

Figure 4.13: Average reprojection errors on the small (a) and large (b) sphere data sets with RANSAC.

With $\sigma = 0$, Tables 4.3 and 4.4 show that the RANSAC framework is correctly removing outliers as the results are the same. As noise levels increase, results are similar to results with optimization only. The main difference here is that accuracy degrades slightly as noise levels increase. This is to be expected because as noise levels increase, RANSAC has difficulty distinguishing between true outliers and highly noisy data. Sometimes, too many points are thrown out, resulting in poor geometry for pose estimation.

4.4 Structure from Motion

In this section results are displayed and discussed on the full SfM system. The system is tested on a simulated data set and a real data set. Testing on the simulated data set is only performed using the small sphere set. On the real data, only a qualitative analysis is available due to unknown true structure of the scene.

It is important to note the efficiency of the system. Due to the software being implemented in MATLAB and not currently optimized for efficiency, the system is very slow in the current state, bottlenecked by bundle adjustment. Performing the proposed SfM system on 15 views of a scene with approximately 500 object points takes > 4 hours. Therefore, testing and debugging is difficult and tedious. Still, as a result of optimizing over the virtual camera error, the system is much quicker than using the reprojection error. A test on 3 views showed an approximate 7 fold increase in processing time for bundle adjustment.

4.4.1 Simulated Results

Tests on the simulated data set were performed over 15 views of the scene. The quantitative results on the simulated data can be seen in Figure 4.14.

The plots in Figure 4.14 show the current system is failing as new views are introduced. As noise increases, errors in pose rapidly increase to the point of failure. Even with zero noise, there are still large errors.

There are a few possible reasons for failure. First, there is likely still a bug in the bundle adjustment software. Second, the MATLAB Levenberg-Marquardt optimization function is using numerical derivatives to compute the Jacobian matrix. Numerical derivatives can result in worse accuracy in optimization problems when compared to analytical derivatives as they are an approximation. Lastly, it is likely that the implicit errors discussed in Section 4.3 are accumulating as new views are added.

Qualitative visualizations of the reconstructed scene can be seen in Appendix B.

Table 4.5: SfM errors on simulated data when $\sigma = 0$. Errors are in mm, degrees, and pixels for translation, rotation, and reprojection, respectively.

Translation	**Rot X**	**Rot Y**	**Rot Z**	**RE**
10	0.933	0.921	1.168	0.035

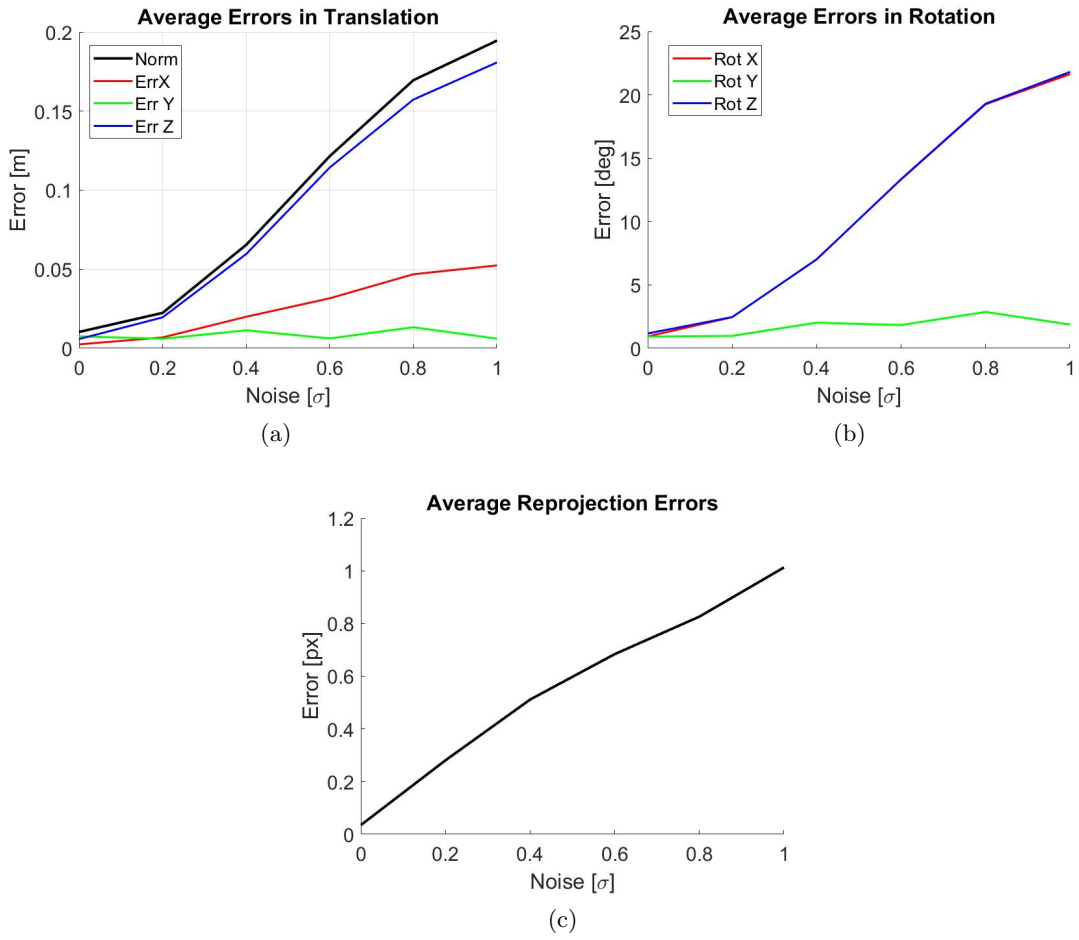


Figure 4.14: Results on the simulated data. Figure (a) shows the average translation errors over all poses. Figure (b) shows the average rotation errors over all poses. Figure (c) shows the average reprojection error.

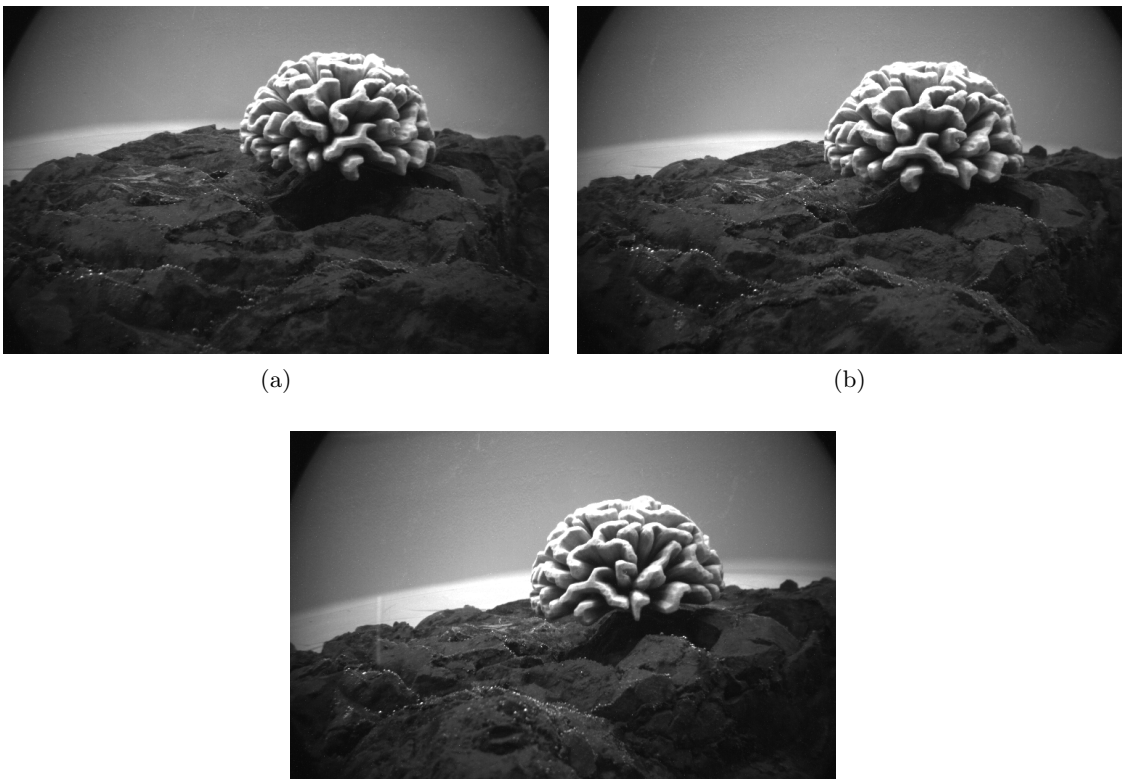
Further, looking at the reprojection errors, it seems that the bundle adjustment is still converging to a solution, though incorrect. This is most likely because of the bug in the bundle adjustment code.

4.4.2 Real Data Results

Lastly, results on a real data set are displayed in this section. An exemplary sequence of 3 images in the underwater setting is displayed in Figure 4.15. The image sequence contains 18 views.

The reconstruction of the underwater scene can is visualized in Figure 4.16.

As stated previously, it is difficult to evaluate the reconstruction quantitatively due to the true poses being unknown. However, the average reprojection error is .975 px. The system reconstructs parts of the coral and edges of the tile below the coral, though it is possible that these points have large errors. The underwater SfM had difficulty tracking features throughout multiple view sequences. Often, features are only tracked in 2 or 3 stereopairs (4 or 6 images).



(a) (b)

(c)

Figure 4.15: Exemplary underwater image sequence.

Additionally, features tracked over multiple stereopairs are extremely limited, ranging from only 10 to 30. It is possible to use sequences closer together, but the time limitation of bundle adjustment limits the amount of overlap between images to reconstruct a scene. Large overlap will contain many more images resulting in extremely long processing times.

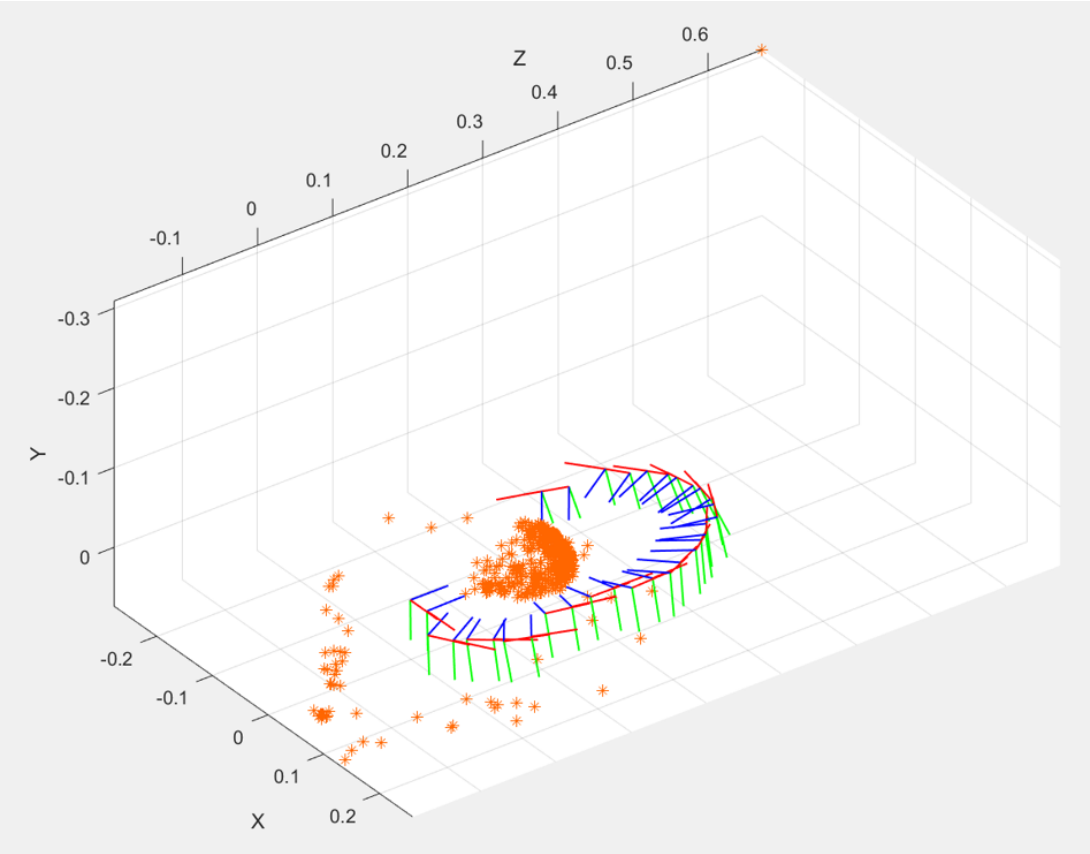


Figure 4.16: Reconstructed underwater scene.

5 Conclusion

To meet growing demands for automatic underwater reconstruction, this project attempts to develop an underwater structure from motion system. The work uses a compact, consumer-grade sensor and explicitly models refraction with calibrated parameters of the camera housing. Methods for robust underwater feature matching, triangulation, absolute pose estimation, and bundle adjustment are implemented.

The work quantitatively analyzes the proposed methods on simulated data and qualitatively on real underwater scene. Results show that feature matching with the generalized epipolar constraint allows for removal of incorrect matches within a stereopair. Further, the absolute pose estimation is tested on each step of the proposed method – the initial estimation, the initial estimation optimized using the Levenberg-Marquardt algorithm, and the optimized solution within a RANSAC framework. Absolute pose estimation is tested on two sets of simulated data where varying levels of noise are added to the image observations. Results show that, with good coverage of features within the image, the absolute pose estimation method is robust to noise. However, when features are concentrated near the center, pose estimation is less robust to noise. Further, with outliers introduced, the pose estimation method within a RANSAC framework robustly estimates the absolute pose with only a slight decrease in accuracy.

One of the most notable insights found from the absolute pose estimation analysis is the presence of errors when there is no noise in the image observations. There is less error in the data set with coverage clustered near the center than when observations are distributed throughout the image. This is due to numerical errors present when solving the 12^{th} degree polynomial for forward projection and also when triangulating within a stereopair. Further, the baseline within the stereopair is narrow. Narrow angles typically lead to less geometrically accurate solutions in triangulation.

Lastly, the full system is tested on simulated data and real data. Results on the simulated data show that the system begins to fail as noise is added to the image observations. The cause of failure is mainly due to three problems. First, there is likely a bug still in the bundle adjustment. Because bundle adjustment is extremely time consuming, testing and debugging the full system is difficult. Second, numeric derivatives for the bundle adjustment are used. Numeric derivatives can be less accurate than analytical derivatives. Lastly, implicit errors from forward projection and triangulation may be accumulating. The results on real data set show that it is reconstructing some parts of the scene, but quantitative evaluation is not possible as the poses are unknown.

In general, it is shown that underwater reconstruction is a complex, sensitive problem. In addition to the problems already mentioned, assumptions of known refractive indicies and effects of temperature and pressure underwater can have small effects on modeling refraction.

With regard to future work, a first step is to fix the bundle adjustment module. Further, implementing analytical derivatives can likely increase accuracy of the bundle adjustment. To increase efficiency of the system, a C++ implementation is recommended. While using the virtual cam-

era error greatly increased time cost of the bundle adjustment compared to using reprojection error, the bundle adjustment is still a significant bottleneck. It would also be interesting to estimate the housing parameters of the camera in the bundle adjustment.

Moreover, it is suggested to create an accurate ground truth data set on real world underwater data. The purpose of this is to not only test the proposed method in this project, but for other underwater reconstruction methods to be tested, compared, and possibly bench marked. Lastly, it would be interesting to test the system on a non-fisheye standard perspective camera. This would limit the complexity of the problem and it would be interesting if accuracy of results increase.

References

Agrawal, A., Ramalingam, S., Taguchi, Y., and Chari, V. (2012). A theory of multi-layer flat refractive geometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3346–3353. IEEE.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359.

Bryant, M., Wettergreen, D., Abdallah, S., and Zelinsky, A. (2000). Robust camera calibration for an autonomous underwater vehicle.

Cotugno, J., Corinne, S., and Nater, A. (2016). Diving into 3d - underwater multi-view stereo.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.

Fryer, J. and Fraser, C. (1986). On the calibration of underwater cameras. *The Photogrammetric Record*, 12(67):73–85.

Furgale, P., Rehder, J., Maye, J., and Schneider, T. (2013). Kalibr. `https://github.com/ethz-asl/kalibr/wiki`.

Geiger, A. (2016). Computer vision lecture notes.

Grossberg, M. D. and Nayar, S. K. (2001). A general imaging model and a method for finding its parameters. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 108–115. IEEE.

Haner, S. and Åström, K. (2015). Absolute pose for cameras under flat refractive interfaces. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1428–1436.

Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

Jordt, A. (2014). *Underwater 3D Reconstruction Based on Physical Models for Refraction and Underwater Light Propagation*. PhD thesis, Universitätsbibliothek Kiel.

Jordt, A., Köser, K., and Koch, R. (2016). Refractive 3d reconstruction on underwater images. *Methods in Oceanography*.

Jordt-Sedlazeck, A., Jung, D., and Koch, R. (2013). *Refractive Plane Sweep for Underwater Images*, pages 333–342. Springer Berlin Heidelberg, Berlin, Heidelberg.

Jordt-Sedlazeck, A. and Koch, R. (2012). *Refractive Calibration of Underwater Cameras*, pages 846–859. Springer Berlin Heidelberg, Berlin, Heidelberg.

Jordt-Sedlazeck, A. and Koch, R. (2013). Refractive structure-from-motion on underwater images. In *IEEE International Conference on Computer Vision (ICCV)*.

Kannala, J. and Brandt, S. S. (2006). A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(8):1335–1340.

Kneip, L., Scaramuzza, D., and Siegwart, R. (2011). A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2969–2976. IEEE.

Lavest, J., Rives, G., and Lapresté, J. (2000). Underwater camera calibration. *Computer vision—ECCV 2000*, pages 654–668.

Lee, G. H., Li, B., Pollefeys, M., and Fraundorfer, F. (2016). Minimal solutions for pose estimation of a multi-camera system. In *Robotics Research*, pages 521–538. Springer.

Li, R., Li, H., Zou, W., Smith, R. G., and Curran, T. A. (1997). Quantitative photogrammetric analysis of digital underwater video imagery. *IEEE Journal of Oceanic Engineering*, 22(2):364–375.

Lorusso, A., Eggert, D. W., and Fisher, R. B. (1995). *A comparison of four algorithms for estimating 3-D rigid transformations*. University of Edinburgh, Department of Artificial Intelligence.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *International Conference on Computer Vision, 1999*, pages 1150–1157. IEEE.

Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., and Sayd, P. (2009). Generic and real-time structure from motion using local bundle adjustment. *Image and Vision Computing*, 27(8):1178 – 1193.

Nistér, D. (2004). A minimal solution to the generalised 3-point pose problem. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE.

Olson, E. (2011). Apriltag: A robust and flexible visual fiducial system. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407. IEEE.

Pless, R. (2003). Using many cameras as one. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–587. IEEE.

Polyanskiy, M. (2008). Refractive index database. `https://RefractiveIndex.INFO`.

Ramalingam, S., Lodha, S. K., and Sturm, P. (2006). A generic structure-from-motion framework. *Comput. Vis. Image Underst.*, 103(3):218–228.

Sedlazeck, A. and Koch, R. (2011). Calibration of housing parameters for underwater stereo-camera rigs. In *British Machine Vision Conference (BMVC)*, pages 1–11. Citeseer.

Sedlazeck, A. and Koch, R. (2012). Perspective and non-perspective camera models in underwater imaging – overview and error analysis. In *Outdoor and Large-Scale Real-World Scene Analysis*, pages 212–242. Springer.

Sturm, P. and Ramalingam, S. (2004a). A generic concept for camera calibration. *Computer Vision-ECCV 2004*, pages 1–13.

Sturm, P. and Ramalingam, S. (2004b). *A Generic Concept for Camera Calibration*, pages 1–13. Springer Berlin Heidelberg, Berlin, Heidelberg.

Terzakis, G., Culverhouse, P., Bugmann, G., Sharma, S., and Sutton, R. (2014). On quaternion based parameterization of orientation in computer vision and robotics. *Journal of Engineering Science & Technology Review*, 7(1).

Treibitz, T., Schechner, Y. Y., and Singh, H. (2008). Flat refractive geometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380.

WikiCommons (2015). Refractive index.

A Declaration of Originality

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Diving into 3D - Underwater 3D Reconstruction

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Cotugno	Jonathan

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zürich, 03.07.2017	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

B SfM Reconstruction of Simulated Data

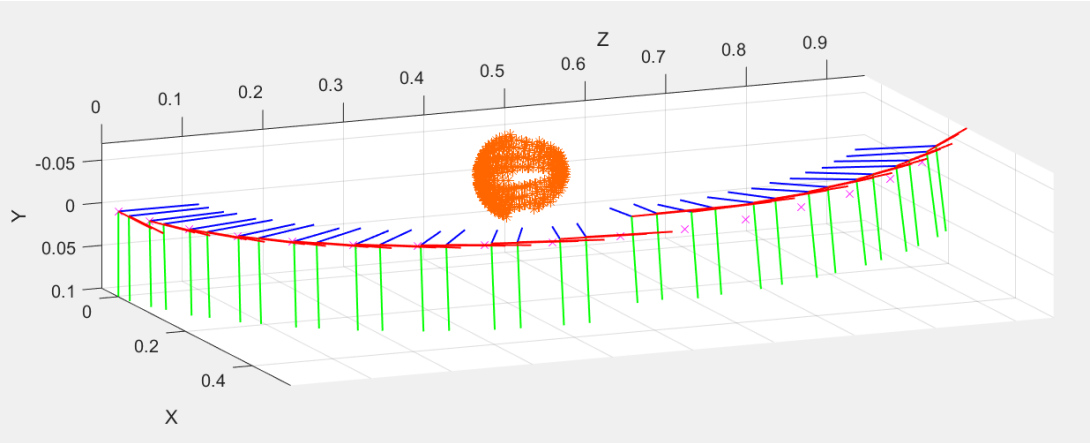


Figure B.1: SfM results on simulated data with noise level $\sigma = 0$. Magenta points are the true rig center locations.

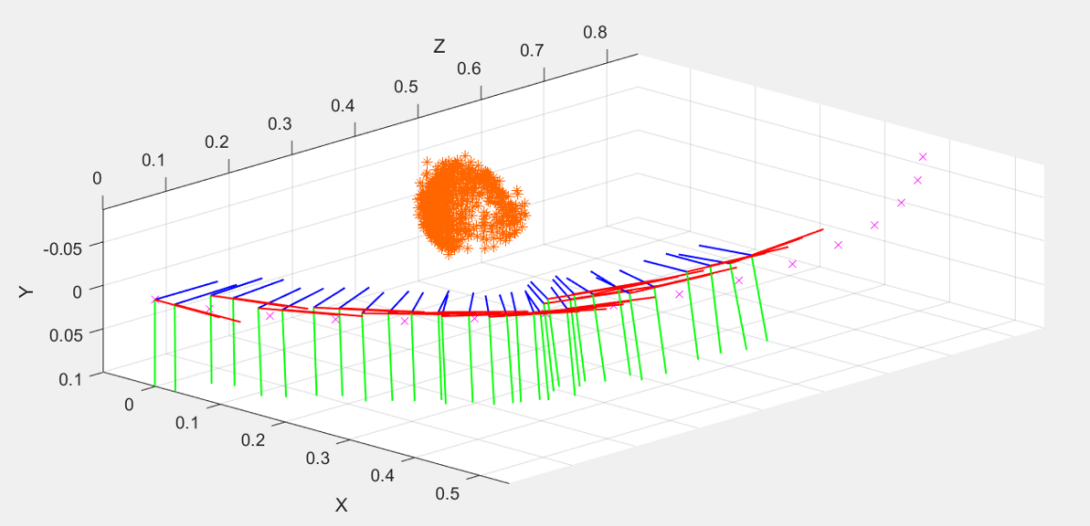


Figure B.2: SfM results on simulated data with noise level $\sigma = 1$. Magenta points are the true rig center locations.