

Underwater Camera Calibration and 3D Reconstruction

by

Timothy Ho Yeung Yau

A thesis submitted in partial fulfillment of the requirements for the
degree of

Master of Science

Department of Computing Science

University of Alberta

© Timothy Ho Yeung Yau, 2014

Abstract

The recovery of 3D information from 2D images is a well-studied problem in computer vision, with many competing methods that can achieve highly accurate results. However, relatively little attention has been paid to the problem of 3D reconstruction in underwater environments. When cameras are placed underwater they must be protected by a waterproof housing, often featuring a flat glass port through which the scene is viewed. Light rays passing from water into the housing are bent by refraction, a nonlinear process that renders the standard perspective camera model invalid. In spite of this, it is common practice in photogrammetry studies to treat refraction as a radial lens distortion, which can lead to errors in the measured 3D information.

Recent research has acknowledged that a physically-correct model of refraction is needed to obtain accurate 3D reconstructions in underwater environments. One important and necessary step is to calibrate the parameters of such a refraction model. In this thesis we develop a novel calibration method that exploits the dispersion of light, which is the angular separation of different wavelengths during refraction, and show that better accuracy is achieved compared to previous work. We then show how to adapt existing reconstruction algorithms to use the physical refraction model, and combine it with our calibration method to obtain a complete process for underwater 3D reconstruction. By analyzing the reconstruction results and comparing against the standard perspective model approximation, we identify the shortcomings of the latter and reveal situations where the refraction model is indispensable. Lastly, we apply the techniques developed in this thesis to real data captured by an underwater observatory and obtain high quality 3D reconstruction results.

Preface

Research for this thesis was conducted under the supervision of Dr. Minglun Gong at the Memorial University of Newfoundland and Dr. Herbert Yang at the University of Alberta. Portions of Chapters 3 and 4 were published as T. Yau, M. Gong, and Y.H. Yang, "Underwater Camera Calibration using Wavelength Triangulation," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, June 25-27, 2013, Portland, Oregon. I, Timothy Yau, was responsible for the concept, data collection and analysis, and manuscript composition, while my supervisors provided guidance and comments on the manuscript.

Acknowledgements

I would like to thank Dr. Herb Yang and Dr. Minglun Gong for their invaluable guidance and support throughout my research program. They provided me with much knowledge and insight, and encouraged me to keep me going in spite of the difficulties I encountered. This thesis would not be possible without them.

I also thank the members of the Computer Graphics Group for their friendship over the years, and for their helpful comments during our group meetings and discussions. The group members include Xida Chen, Rouzbeh Maani, Yi Gu, Ding Liu, Samaneh Eskandari, Mohsen Taghaddosi, and Neda Aslsabbaghpourhokmabadi.

Steve Sutphen from the Department of Computing Science provided technical assistance on many occasions. He was also responsible for the detailed design and assembly of the calibration device described in this thesis. Without his help, this research would have been much more challenging to carry out.

Underwater imagery and data was supplied by Ocean Networks Canada. I greatly appreciate the NEPTUNE team at the University of Victoria for the opportunity to work with the Folger Pinnacle observation platform prior to its deployment. In particular, Dr. Maia Hoeberechts coordinated my on-site visit and helped me to use the camera equipment.

I received financial support from the Natural Sciences and Engineering Research Council of Canada, Alberta Innovates Technology Futures, and the Faculty of Graduate Studies and Research. The generosity of all donors is gratefully acknowledged.

Lastly, I thank my parents for working so hard to raise me, and for loving and supporting me unconditionally.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Background	2
1.3	Contributions	5
2	Related Work	8
2.1	Structure-From-Motion	9
2.1.1	Triangulation	9
2.1.2	Epipolar Geometry and Feature Matching	11
2.1.3	Sparse Reconstruction	13
2.1.4	Dense Reconstruction	16
2.2	Refractive Reconstruction	19
2.2.1	Reconstructing refractive surfaces	19
2.2.2	Reconstructing scenes with refractive distortion	21
2.3	Underwater Stereo	24
3	Refractive Geometry	32
3.1	Refraction	32
3.2	Dispersion	34

3.3	Flat Refraction Model	35
3.3.1	Back Projection	36
3.3.2	Forward Projection	38
3.3.3	Perspective Approximation	39
3.3.4	Dispersion Analysis	42
3.3.5	Geometric Constraints	46
4	Underwater Camera Calibration	49
4.1	Calibration Method	50
4.1.1	Refraction Axis	51
4.1.2	Calibration Object Pose	53
4.1.3	Refraction Layer Thicknesses	55
4.1.4	Nonlinear refinement	55
4.2	Calibration Device	57
4.2.1	Design and Construction	57
4.2.2	Point Detection and Localization	59
4.2.3	Chromatic Aberration Correction	61
4.3	Experimental Results	61
4.3.1	Synthetic Data	62
4.3.2	Real Data	67
5	Underwater 3D Reconstruction	71
5.1	Reconstruction Method	72
5.1.1	Reference Implementation	72
5.1.2	Refraction-Corrected Implementation	72
5.2	Camera Array Calibration	77

5.2.1	Extrinsic Calibration	77
5.2.2	Intrinsic Calibration	78
5.3	Experimental Results	79
5.3.1	Evaluation Methodology	79
5.3.2	Synthetic Data Results	80
5.3.3	Real Data Results	98
5.3.4	Application to Seabed Images	109
6	Conclusions and Future Work	115
6.1	Future Work	118
	Bibliography	127
A	Supplementary Experimental Data	128

List of Tables

2.1	Tractability of light path triangulation problems.	20
4.1	Configurations for synthetic data experiments.	62
4.2	Refractive indices used in all experiments.	62
4.3	Estimated calibration parameters using real data.	69
5.1	Error values added to the calibration data.	95
5.2	Seawater conditions at time of capture of the undersea dataset. . .	109
5.3	Point cloud distance statistics after ICP alignment.	114
A.1	Calibration data for our 3D reconstruction experiments with a wa- ter tank.	132
A.2	Calibration data for the Folger Passage underwater observation platform camera array.	137

List of Figures

1.1	The underwater camera array deployed by Ocean Networks Canada under the NEPTUNE program [3].	3
1.2	Underwater view of a sponge captured by the camera array [3]. . .	3
2.1	Triangulation of a 3D point from multiple views.	9
2.2	Results of the Tomasi-Kanade factorization method.	14
2.3	Photo Tourism from internet image collections.	16
2.4	Example 3D reconstruction with PMVS2.	17
2.5	Patch representation in PMVS2.	18
2.6	Fluid surface reconstruction by Morris and Kutulakos.	19
2.7	Restricting the refraction problem with known vertical direction. .	22
2.8	Depth from refraction method by Chen <i>et al.</i> [11].	23
2.9	Rays imaged by a perspective camera in an underwater housing with a flat port.	24
2.10	Two-view underwater 3D reconstruction results obtained by Kang <i>et al.</i> [31].	27
2.11	Flat refractive geometry used by Agrawal <i>et al.</i> [2].	28
3.1	An illustration of Snell's Law.	33

3.2	Refractive index of distilled water as a function of wavelength [14, 41].	35
3.3	Flat refraction imaging model.	36
3.4	Formulation of forward projection of a scene point for one refraction.	38
3.5	Approximate focal length adjustment for a single refraction. . . .	40
3.6	Failure of radial distortion to approximate refraction.	41
3.7	Dispersion of light in the flat refraction imaging model, for a single refraction.	43
3.8	Dispersion of light in a multi-layer flat refraction imaging model. .	44
3.9	Light paths for scene points observed under two different wavelengths, yielding the dispersion and triangulation constraints. . . .	47
4.1	Effect of applying a local neighbourhood average over dispersion point pairs for estimating the refraction axis.	51
4.2	The calibration device that we constructed to emit two wavelengths of light in a known spatial pattern.	57
4.3	Close-up of an image of six illuminated points on the calibration device.	59
4.4	Synthetic data results for configuration I.	64
4.5	Synthetic data results for configuration III.	65
4.6	Reprojection error after nonlinear refinement for synthetic data configuration III.	66
4.7	Experimental setup.	67
4.8	Refraction axis estimation results.	69

5.1	Relationship between focal length and object distance.	75
5.2	Apparent shift in object position due to refraction.	76
5.3	Renderings of the scene used in our synthetic data experiments. . .	80
5.4	Error distribution and scale factor plots for two views.	84
5.5	Error distribution and scale factor plots for four views.	85
5.6	Error distribution and scale factor plots for eight views.	86
5.7	Reconstructed 3D point clouds overlaid on the ground truth ge- ometry.	89
5.8	Reconstructed 3D point clouds, with highlighted regions showing deviations from the ground truth geometry.	90
5.9	Reconstructed 3D point clouds, showing examples where Rdist+Fadj performs poorly.	91
5.10	Completeness measure for 3D reconstruction results.	91
5.11	Error distribution and scale factor plots for eight views.	93
5.12	Completeness measure for eight view 3D reconstruction results using feature points detected by Bundler.	94
5.13	Bundle adjustment results for two views.	95
5.14	Bundle adjustment results for four views.	96
5.15	Bundle adjustment results for eight views.	96
5.16	Reconstruction accuracy results for two views.	98
5.17	Reconstruction accuracy results for four views.	98
5.18	Reconstruction accuracy results for eight views.	99
5.19	Experimental setup for our real data experiments.	99
5.20	Capturing ground truth 3D model.	100

5.21	Examples of the images used in our real data 3D reconstruction experiments.	102
5.22	3D reconstruction results for the FlatRefract method.	102
5.23	3D reconstruction results for Rdist+Fadj.	104
5.24	Accuracy of real data 3D reconstructions.	104
5.25	Completeness of real data 3D reconstructions.	105
5.26	Camera array calibration result.	107
5.27	Camera array calibration result, after bundle adjustment.	108
5.28	A straight edge placed against the wall of our water tank.	108
5.29	Undersea dataset from the Folger Passage observatory operated by Ocean Networks Canada.	110
5.30	Qualitative comparison of the 3D reconstruction results.	112
5.31	Difference in scale between the FlatRefract and Rdist+Fadj reconstruction results.	113
5.32	Point cloud distance between the FlatRefract and Rdist+Fadj reconstruction results.	113
A.1	Error distribution and scale factor plots for two views, using feature points detected by Bundler.	129
A.2	Error distribution and scale factor plots for four views, using feature points detected by Bundler.	130
A.3	Completeness measure for two views, using feature points detected by Bundler.	131
A.4	Completeness measure for four views, using feature points detected by Bundler.	131

A.5	3D Reconstruction completeness results for simulated data with varying amounts of error in the initial refraction model parameters.	132
A.6	3D reconstruction results for Rdist+Fadj using two views (fixed t), overlaid on the ground truth model.	133
A.7	Accuracy of real data 3D reconstructions.	133
A.8	Completeness of real data 3D reconstructions.	134
A.9	Accuracy of real data 3D reconstructions.	134
A.10	Completeness of real data 3D reconstructions.	134
A.11	Calibration result for the Folger Passage underwater observation platform camera array, after bundle adjustment.	135
A.12	Additional views of the point cloud reconstructed with the FlatRefract method.	136
A.13	Reconstructed 3D model with the FlatRefract method, after applying Poisson Surface Reconstruction.	137

Chapter 1

Introduction

1.1 Motivation

Camera-based 3D reconstruction is the process of extracting three-dimensional geometric information about a scene from 2D images. Researchers in computer vision and photogrammetry have explored many different approaches to this problem, and these approaches are collectively termed Shape-from-X, indicating that the shape or geometric structure of the scene is being deduced using a particular property of the images or imaging process. Examples of X include scene shading [55], defocus blur [17], and specularities [38]. The most common approach, however, is known as Structure-from-Motion, where “motion” means that multiple images of the scene are captured with the camera in different positions; the camera may be physically moved, or multiple cameras may be set up as an array. The 3D structure of the scene is then obtained by identifying points seen from different angles, and triangulating them geometrically.

The Structure-from-Motion (SfM) approach has enjoyed great success, with

applications such as photo-tourism [48], terrain mapping [49], and motion capture [25]. There are nearly 60 different algorithms listed on the Middlebury Multi-View Stereo website, a standard benchmark for 3D reconstruction algorithms. On the test scenes, which measure about 10cm across, the best algorithms are able to achieve accuracies well within 1mm [43]. However, all of these algorithms are designed to be used on land. The emergence of underwater 3D reconstruction as an application area has presented a new set of challenges.

Cameras have been used in underwater environments for many different purposes, including: monitoring marine habitats [47], tracking fish populations [47], reconstructing archaeological sites [8], and inspecting industrial equipment [5]. Our work is motivated by the increasingly common practise of deploying multiple cameras aboard underwater remotely-operated vehicles (ROVs) and observation platforms. For example, Ocean Networks Canada has deployed an 8-camera array (Fig. 1.1) for observation of the sea bed off the coast of Vancouver Island [3]. This camera array was designed to enable SfM-based 3D reconstruction, giving biologists a way to make regular, non-intrusive, in-situ measurements of marine life forms.

1.2 Background

There are multiple challenges involved in underwater imaging and 3D reconstruction. Besides the engineering issues of enclosing cameras in watertight housings, configuring an array of cameras, and providing sufficient scene illumination, some computational challenges arise as well. Sea water often has high turbidity because of dissolved substances and suspended particles. This leads to backscattering and



Figure 1.1: The underwater camera array deployed by Ocean Networks Canada under the NEPTUNE program [3].



Figure 1.2: Underwater view of a sponge captured by the camera array [3].

attenuation of light, which reduces image contrast and limits the depth at which objects can be seen. Furthermore, different wavelengths of light are absorbed at different rates, causing depth-dependent color casts. Some recent work has been done to simulate and compensate for these effects [27, 45].

Our work is primarily concerned with the challenge introduced by refraction, the commonly-seen effect that, for example, causes a straw to appear bent when placed into a glass of water. This occurs because light waves propagate at a different speed in water than in air. A typical underwater camera housing has a flat piece of glass in front of the lens, resulting in two refractions as the light travels from water to glass to air. It was common in prior work to model refraction as a radial distortion, as this was found to be satisfactory for applications such as measuring fish [47]. More recently, there is a trend toward applying a physically-correct refraction model, in the interest of obtaining more accurate 3D reconstructions [2, 9, 19, 31, 27, 54].

The principles behind modelling refraction and incorporating it into existing SfM algorithms have been studied to some extent [9, 19]. Before 3D reconstruction can be carried out, however, the first step is to obtain the parameters of the refraction model. Similar to estimating the focal length and other characteristics of the camera lens, a calibration process is needed to ensure that the refraction model accurately describes how light propagates from the scene to the camera.

Earlier work on refractive calibration relied on nonlinear optimization procedures and heuristic initialization, which did not always prove reliable [19, 44]. In contrast, our work is inspired by the discovery of some mathematical properties of the refraction model that enable more efficient and reliable estimation of the model parameters [2]. We build upon these results by studying the phenomenon

of dispersion, which is the differential refraction of light according to wavelength. Dispersion often causes undesirable effects, such as color fringing around sharp edges in an image, but we show that it is in fact useful in the calibration process.

As for the actual 3D reconstruction, ambiguous results have been reported in the prior work. Some authors have experienced mixed success [19], while others reported success but without quantitative evaluation against ground truth using real data [9, 28, 31]. Still others indicated that a physically-correct refraction model may not be needed after all, applying the land-based methods directly and suggesting that they work well enough [30]. It is clear that a more comprehensive study is needed to quantitatively evaluate 3D reconstruction with and without a refraction model, and to determine whether the refraction model can lead to better results in practice.

1.3 Contributions

The goal of our work is to develop a practical solution for underwater 3D reconstruction. Our solution involves two main parts: first, a physically correct and accurate calibration method; and second, adaptations of existing 3D reconstruction algorithms to accommodate for refraction. For the first part, we:

- show that the dispersion of light is a measurable side-effect of refraction using a typical consumer-grade camera,
- derive mathematical constraints on the flat refraction camera model using dispersion,
- incorporate these constraints in a calibration procedure that achieves greater

accuracy compared to existing work, and

- demonstrate how this calibration can be achieved in practise with a novel calibration device.

For the second part, we:

- identify and implement the changes needed to use existing 3D reconstruction algorithms with the refraction model,
- evaluate the quality of reconstructions using the refraction model compared to a standard perspective camera model,
- analyze the impact of errors in calibration, and
- show that with an accurate calibration and a physically correct refraction model, high quality 3D reconstructions can be obtained in practise.

In the course of this work we also derive some computational techniques that are new, to the best of our knowledge. We implemented a method to forward project points in multi-layer refraction systems with an arbitrary number of refractions. We also created specialized techniques to process images of the light pattern emitted by our novel calibration device. Details on these techniques are given in the relevant sections.

The remainder of this thesis is organized as follows: in Chapter 2 we survey the existing work related to traditional 3D reconstruction, as well as 3D reconstruction in the presence of refraction. In Chapter 3 we look at refraction and dispersion in more detail, study the geometry of a flat refraction camera model, and derive new mathematical constraints. We then describe how these constraints

are used in a complete camera calibration procedure in Chapter 4, together with experiments and results. The penultimate chapter focuses on the application of the preceding concepts to multi-view underwater 3D reconstruction. Lastly, we close with a discussion and possible future work.

Chapter 2

Related Work

There is a vast amount of research on the topic of 3D reconstruction. A comparative evaluation of many different algorithms can be found on the Middlebury Multi-View Stereo website [43], and the associated paper provides a useful taxonomy for classifying these algorithms [46]. We will not attempt to give a comprehensive survey, but only to cover the key ideas such as triangulation, feature matching, and epipolar geometry, before expanding on a few specific algorithms most relevant to our work.

Having set the stage with the traditional SfM approach, we will describe a number of works that generalize the problem to include transparent, refractive objects. Some methods are intended to reconstruct the surface(s) of the object such as, for example, the ripples on a body of water. Other methods are designed to reconstruct a scene in spite of, or with the help of, a refractive object between the camera and the scene.

Lastly, we will give an overview of the research in underwater 3D reconstruction. These works generally assume that the refraction boundaries have a simple

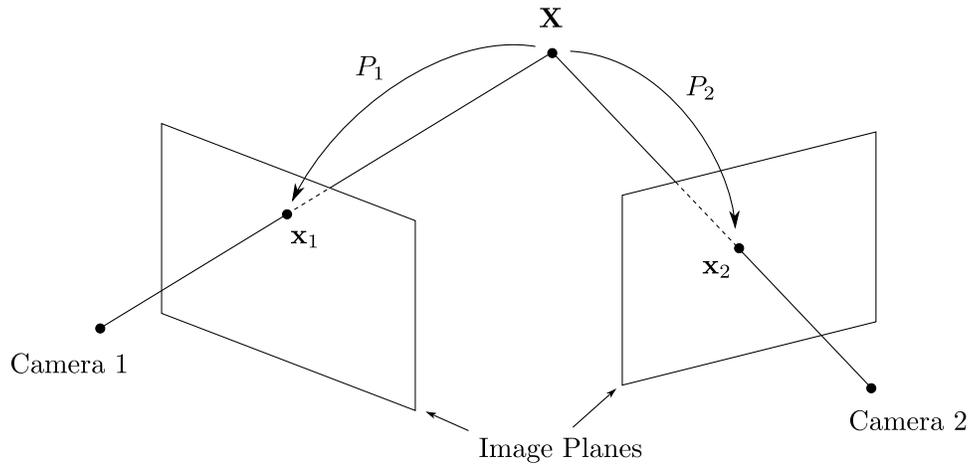


Figure 2.1: Triangulation of a 3D point from multiple views.

form, namely parallel planes, and that they are an integral part of the camera system. The goal is to accurately model the refractive imaging process in order to optimize the 3D reconstruction of the scene.

2.1 Structure-From-Motion

2.1.1 Triangulation

Consider a scene that is imaged from different viewpoints with a conventional camera. We wish to deduce the three-dimensional structure of the scene by observing how it changes as the camera is moved. Suppose that a point X in the scene is imaged to points x_1 and x_2 as shown in Figure 2.1. If the projection matrices of the cameras are P_1 and P_2 respectively, then we have two linear con-

straints of the form:

$$\begin{cases} P_1 \mathbf{X} = k_1 \mathbf{x}_1 \\ P_2 \mathbf{X} = k_2 \mathbf{x}_2 \end{cases} \quad (2.1)$$

where k_i is an unknown scale factor. For the general case of n views, we have n such equations that can be combined into a homogeneous linear system as in

$$\begin{bmatrix} P_1 & \mathbf{x}_1 & & & \\ & P_2 & \mathbf{x}_2 & & \\ & \vdots & & \ddots & \\ & & & & P_n & \mathbf{x}_n \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ -k_1 \\ -k_2 \\ \vdots \\ -k_n \end{bmatrix} = 0 \quad (2.2)$$

Given known P_i and x_i , equation 2.2 can be solved to obtain \mathbf{X} . This is known as the linear triangulation method, and it is most commonly used because of its simplicity. Although the algebraic error being minimized is not meaningful, it can be modified with an iterative adjustment step to give good performance in most cases [23]. Hartley and Sturm describe an optimal triangulation method [23] for two views based on solving a set of polynomial equations, but unfortunately it is difficult to extend to three or more views [29].

Another simple triangulation method is known as the midpoint method. Here we consider the rays corresponding to the back-projections of the image points \mathbf{x}_1 and \mathbf{x}_2 . In the ideal case these rays would intersect at a point, but in general they do not due to measurement noise. Therefore the point \mathbf{X} is estimated to be the midpoint along the perpendicular line joining the two rays. Writing $P_1 =$

$[M_1 | -M_1\mathbf{c}_1]$ and $P_2 = [M_2 | -M_2\mathbf{c}_2]$, where \mathbf{c}_i are the camera centres and M_i is a 3×3 matrix, the ray directions are given by $M_1^{-1}\mathbf{x}_1$ and $M_2^{-1}\mathbf{x}_2$. The common perpendicular of the two rays is $\mathbf{u} = (M_1^{-1}\mathbf{x}_1) \times (M_2^{-1}\mathbf{x}_2)$, so we have

$$\mathbf{c}_1 + \alpha_1 M_1^{-1}\mathbf{x}_1 + \beta \mathbf{u} - \alpha_2 M_2^{-1}\mathbf{x}_2 - \mathbf{c}_2 = 0 \quad (2.3)$$

This vector equation can be easily solved for the unknowns α_1, α_2 , and β (with \mathbf{c}_1 and \mathbf{c}_2 obtained from P_1 and P_2). The midpoint method has reasonable performance for Euclidean reconstruction, when the intrinsic camera parameters are known, but in general it is not as good as other methods [23]. It is, however, easy to implement for refraction because it considers only the back-projected rays and not the projection matrices.

2.1.2 Epipolar Geometry and Feature Matching

The preceding discussion assumes that we already know the correspondence between a scene point \mathbf{X} and the imaged points $\mathbf{x}_1, \mathbf{x}_2$. Moreover, it assumes that the camera projection matrices are known, which combine both the extrinsic parameters (rotation, translation) and intrinsic parameters (focal length, principal point). In the conventional framework without refraction, both of these issues are addressed through the concept of the fundamental matrix [24].

Given two views of a scene, any point observed in one image cannot appear arbitrarily in the other image. The back-projection of the point in the first image is a line which must project to a line in the second image. The fundamental matrix

F encodes this constraint, such that

$$\mathbf{x}_2^\top F \mathbf{x}_1 = 0 \quad (2.4)$$

whenever \mathbf{x}_1 and \mathbf{x}_2 correspond to the same point in the scene. Suppose we know \mathbf{x}_1 , then $F\mathbf{x}_1$ is the line in the second image on which \mathbf{x}_2 is constrained to lie. This allows the elimination of false matches and greater efficiency when searching for \mathbf{x}_2 .

Equation 2.4 also gives a way to compute the fundamental matrix. Given a sufficient number of putative correspondence pairs $\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$, we can estimate F with a linear method [24] and an outlier-removal strategy such as RANSAC [56]. The fundamental matrix found in this way can be used to perform triangulation and 3D reconstruction up to a projective ambiguity [24]. If the camera intrinsic matrices K_1, K_2 are known (or obtained using readily available software [4]), we can go further to compute the essential matrix E , which is related to the fundamental matrix by:

$$E = K_2^\top F K_1 . \quad (2.5)$$

The essential matrix can be decomposed to obtain the relative translation \mathbf{t} and rotation R of the two cameras via $E = [\mathbf{t}]_\times R$, where $[\cdot]_\times$ denotes the anti-symmetric cross product matrix [39]. Subsequently we can compute a Euclidean 3D reconstruction using the triangulation techniques described above.

The remaining part of this process is feature matching. Detecting salient features in an image and encoding them into descriptors that can be compared across viewpoints is a non-trivial problem. Some feature detectors include the Harris cor-

ner detector [22], the Scale Invariant Feature Transform (SIFT) [36], and DAISY [52]. The feature descriptors computed by SIFT and DAISY both consist of histograms of image gradients sampled around the feature point.

2.1.3 Sparse Reconstruction

Structure-from-Motion algorithms can be divided into two groups based on the expected density of points in the final reconstruction. Sparse reconstruction is concerned with recovering the position of distinctive feature points, together with the pose of the cameras (rotation and translation), rather than producing a dense 3D model of the scene. This is useful for applications such as image registration and organizing photo collections [48], and also as input to a subsequent dense reconstruction step.

The Tomasi-Kanade factorization method is an example of sparse reconstruction from a set of tracked points in a video stream [53]. In this seminal paper, the authors cast the Structure-from-Motion problem as a matrix factorization problem by analyzing the rank deficiency of the measurement matrix W . For orthographic projection cameras, this yields an elegant solution to obtain the motion matrix M and structure S using the singular value decomposition (SVD). The matrices are

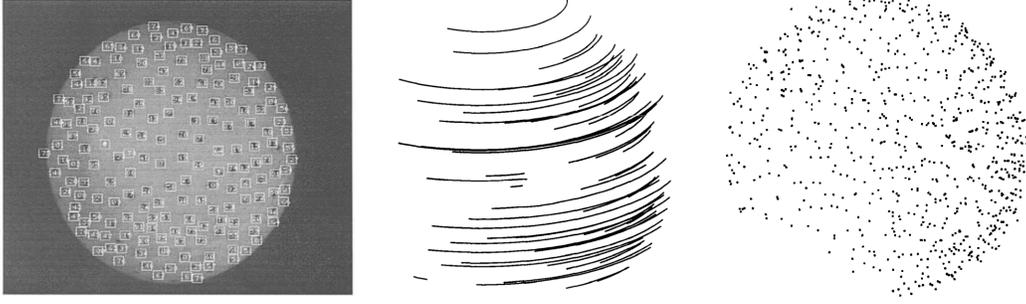


Figure 2.2: Results of the Tomasi-Kanade factorization method. Left to right: a ping-pong ball with marked features, a set of feature tracks as the ball is rotated, final reconstructed points. Source: [53]

defined as follows:

$$W = \begin{bmatrix} u_{1,1} & \cdots & u_{1,N} \\ \vdots & \ddots & \vdots \\ u_{F,1} & \cdots & u_{F,N} \\ v_{1,1} & \cdots & v_{1,N} \\ \vdots & \ddots & \vdots \\ v_{F,1} & \cdots & v_{F,N} \end{bmatrix} \quad M = \begin{bmatrix} \mathbf{r}_{1,1} \\ \vdots \\ \mathbf{r}_{F,1} \\ \mathbf{r}_{1,2} \\ \vdots \\ \mathbf{r}_{F,2} \end{bmatrix} \quad S = \begin{bmatrix} X_1 & \cdots & X_N \end{bmatrix} \quad (2.6)$$

where $(u_{f,k}, v_{f,k})$ is the measured image point number $k \in \{1, \dots, N\}$ in frame $f \in \{1, \dots, F\}$, $\mathbf{r}_{f,j}$ is the j -th row of the rotation matrix for frame f , and X_k is the 3D point number k . These definitions are such that we have $W = MS$, and the task is to factorize W into the unknown M and S .

The key result discovered by Tomasi and Kanade is that $\text{rank}(W) \leq 3$ in the absence of noise. When noise is present, W can be decomposed as $W = U\Sigma V^\top$ by SVD. The closest rank-3 approximation is obtained by $\widehat{W} = U'\Sigma'V'^\top$ which keeps only the largest three singular values and the corresponding columns of U

and V . This yields $\widehat{W} = M'S'$ where $M' = U'(\Sigma')^{1/2}$ and $S' = (\Sigma')^{1/2}V'^\top$, which are equal to the true M and S but with an unknown change of basis. The ambiguity is resolved by constraining the rows of M' to be unit vectors and rows from the same rotation matrix to be orthogonal.

Unfortunately, this factorization method only applies to orthographic projection, and not perspective projection as found in most cameras. Various methods have been proposed that iteratively correct for the projective depth in the measurement matrix [12, 50], but they are more complex and will not be discussed here.

An alternative to the factorization approach is to use a nonlinear optimization algorithm to find the values of all the camera parameters and 3D point positions simultaneously. This technique is known as Bundle Adjustment [24]. Let θ_i represent all of the (intrinsic and extrinsic) parameters of the i -th camera and proj be a function that projects a 3D point onto the image, then the objective is to compute:

$$\min_{\substack{\theta_i, \mathbf{X}_j \\ \forall i, j}} \sum_{k, l} d(\text{proj}(\theta_k, \mathbf{X}_l), \mathbf{x}_l)^2, \quad (2.7)$$

where d is a geometric distance in image space. The error being minimized is called the reprojection error, since it measures the distance between the observed point \mathbf{x}_l and the projection of the estimated 3D point \mathbf{X}_l . One successful application of bundle adjustment is the open-source software Bundler, developed by Noah Snavely [48]. Internally, Bundler relies on specialized optimization software developed by Lourakis and Argyros called sba [35], which uses the well-known Levenberg-Marquardt algorithm and exploits the sparsity structure of the problem. Sparsity comes from the fact that many parameters do not directly influence each



Figure 2.3: Photo Tourism from internet image collections. The SfM system that simultaneously computes the scene structure and camera viewpoints is an implementation of bundle adjustment called Bundler. Source: [48]

other; for example, the parameters of a camera are not affected by the positions of points not visible to it. This allows SfM to be applied on a large scale, such as reconstructing landmarks from thousands of tourist photographs [48]. There are some mechanics involved in strategically adding images, performing bundle adjustment in phases, and filtering outliers, but the underlying method for initializing the bundle adjustment is based on feature matching and pose estimation as described previously.

2.1.4 Dense Reconstruction

Dense reconstruction methods aim to recover as much of the observed surfaces as possible, usually assuming fully-calibrated cameras with known pose. A number of different methods are catalogued by Seitz *et al.* [46] based on the several distinguishing characteristics including scene representation, photo-consistency measure, and the reconstruction algorithm itself. One of the older but still top-performing methods is PMVS2 by Furukawa and Ponce [18]. We chose to adapt this open-source software for underwater 3D reconstruction because of its effectiveness and relative simplicity.



Figure 2.4: Example 3D reconstruction with PMVS2. Left to right: an example input image, detected features, initial patches, final patches after expansion and filtering, mesh model. Source: [18]

The scene representation in PMVS2 consists of a set of small rectangular patches. As shown in Figure 2.5, each patch is parameterized by its center coordinate c and a normal direction n , approximating a local tangent plane of the true surface. Each patch is also assigned a photometric discrepancy score that measures the difference in its appearance between two images. This is computed by overlaying a square grid on the patch, projecting the points onto two images, and calculating the normalized cross-correlation between the images sampled at those points (with interpolation). Under the assumption of Lambertian reflectance, a patch with low photometric discrepancy is more likely to accurately represent the surface.

The overall algorithm consists of three distinct steps: patch initialization, expansion, and filtering. In the initialization step, salient features are detected and matched across pairs of images. Feature pairs that satisfy the epipolar geometry are triangulated into patches, and additional views are added where the patch is also visible. The patches then undergo nonlinear optimization to minimize the photometric discrepancy by adjusting the center and normal. Only those patches having a sufficient number of views with low photometric discrepancy (below a

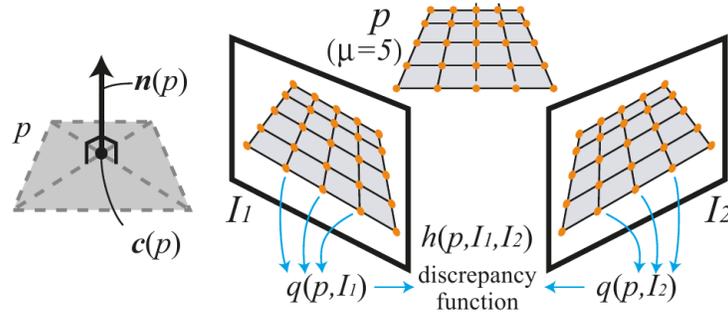


Figure 2.5: Patch representation in PMVS2. Photometric discrepancy is computed by projecting a sampling grid from the patch onto the images. Source: [18]

predefined threshold) are retained.

Subsequently, the algorithm alternates between the expansion and filtering steps for a fixed number of iterations. Expansion means that new patches are created next to existing patches. Each image is divided into a fine grid of square cells, and the objective is to reconstruct at least one patch in each cell. Thus, given a patch whose projection is adjacent to an empty cell, a new patch is created by intersecting the back-projected ray of that cell with the plane of the original patch. The center and normal of the new patch are then adjusted to minimize photometric discrepancy, similar to the initialization step.

Finally, the filtering step applies three different heuristics to remove outliers. The first filter examines patches that fall in the same cell but are not neighbours, removing the ones that have higher photometric discrepancy and fewer views. The second filter removes patches that have too few views when visibility is computed while accounting for depth. The last filter removes patches with too few neighbouring patches lying in adjacent cells. More details about these heuristics are provided in the original paper [18].

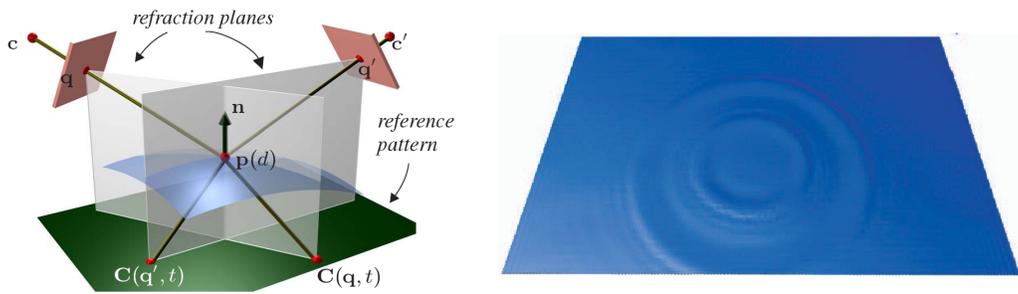


Figure 2.6: Fluid surface reconstruction by Morris and Kutulakos. Source: [37]

2.2 Refractive Reconstruction

2.2.1 Reconstructing refractive surfaces

One of the limitations of traditional SfM methods is that they cannot properly handle scenes containing transparent objects that refract light. Such objects do not have a fixed appearance, and instead derive their appearance from the surroundings by changing the direction and intensity of light rays. While the focus of this thesis is not on reconstructing the geometry of refractive surfaces, there are some interesting and relevant works in this area.

Morris and Kutulakos showed that it is possible to determine the height and normal direction of a point on the surface of a transparent, refracting fluid from two views. Whereas this is an ill-posed problem for a single view, the addition of a second view constrains the space of possible solutions to a discrete set. With reference to Figure 2.6, the key idea is that two light rays from different viewpoints passing through $\mathbf{p}(d)$ on the surface must be refracted according to the same normal direction \mathbf{n} . Suppose $C(\mathbf{q}, t)$ is a function that maps an observed image point \mathbf{q} at time t to a point on the reference pattern. The point \mathbf{p} might be postulated to lie at a distance d along the camera ray for image point \mathbf{q} . This completely

One reference point ($M = 1$)				Two or more ref. points ($M \geq 2$)			
	$K = 1$	$K = 2$	$K \geq 3$		$K = 1$	$K = 2$	$K \geq 3$
$N = 1$				$N = 1$	✓+		
$N \geq 2$	✓+			$N = 2$	✓+		
				$N = 3$	✓+	✓	
				$N \geq 4$	✓+	✓+	

Table 2.1: Tractability of light path triangulation problems. A check mark ✓ indicates that the problem is tractable, and a plus sign + indicates that it is tractable even when the refractive index (or if a deflection is a reflection) is unknown. Unmarked problems are intractable. Source: [33]

determines the refraction angle and \mathbf{n} , which in turn determines how the ray for \mathbf{q}' is refracted. Therefore, one needs only search for a d where the second camera ray is refracted consistently with $\mathbf{C}(\mathbf{q}', t)$. The function \mathbf{C} is obtained by tracking feature points on the reference plane, in this case a checkerboard pattern [37].

This work was expanded into a more general theory of light path triangulation by Kutulakos and Steger [33]. They analyzed the tractability of reconstructing reflective and refractive surfaces using light path consistency, and introduced a system to characterize such reconstruction problems based on the number of viewpoints N , the number of deflections K along the path, and the number of reference points M . A deflection can be either a refraction or reflection, and reference points are on the light path segment farthest from the camera. In their work, a light path triangulation problem is called tractable if the solution space lies on a 0-dimensional manifold, *i.e.* a discrete set of points. The tractability of various problem configurations is summarized in Table 2.1.

2.2.2 Reconstructing scenes with refractive distortion

Sometimes the surface of a refractive object is not of interest, but it causes distortions of the scene that we wish to reconstruct. These distortions may be desirable, for example as part of a multi-axial imaging system [1], or they may be undesirable as in the case of underwater imaging. In either case, the goal is the same: to obtain an accurate model of the distortion and thereby recover the true geometry of the scene. This will typically entail a calibration procedure in which the parameters of the cameras and of the refractive object are estimated (one such set of parameters is detailed in Section 3.3). We will restrict our discussion to the case where the refractive object is planar, as is often true in underwater imaging applications.

Chari and Sturm studied the geometry of the stereo camera configuration, where the scene is separated from the cameras by a single refractive planar surface. By formulating the refraction of rays in terms of quadratic *lifted coordinates*, they derived a 12×12 refractive fundamental matrix analogous to the ordinary fundamental matrix for two views [10]. This result is theoretically interesting and the authors described how it could be used to calibrate the camera pose and refractive surface parameters, but no experimental results are presented. As far as we know there has not been any follow-up work to implement a practical algorithm from these ideas.

A different approach taken by Chang and Chen is to restrict the problem by assuming that some of the calibration parameters are known. Specifically, they used cameras with built-in Inertial Measurement Units (IMUs) to provide the vertical direction of each view, which is perpendicular to the single horizontal refraction surface (such as a still water surface). The relative poses of multiple views, as well

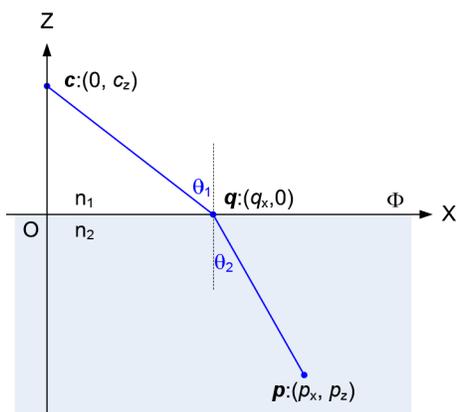


Figure 2.7: Restricting the refraction problem with known vertical direction allows a simplified global coordinate system [9]. In three dimensions there is a Y axis orthogonal to the page.

as the 3D points beneath the refraction surface, can then be computed by solving a system of linear equations.

This result can be understood by writing the entire problem in terms of a coordinate system aligned with the refraction plane Φ , as shown in figure 2.7. Since the vertical direction Z is known at camera c , the incident angle θ_1 of any camera ray with Φ is also known. Therefore, the refracted ray direction can be computed, and the point position p_x (and p_y in three dimensions) becomes a linear function of the camera height c_z and the point depth p_z . The remaining unknown rotation around the Z axis is also linear, as it is given by a matrix multiplication with the x and y components of the camera ray direction. A set of linear equations is obtained by equating the point positions from correspondences between two views and eliminating p_z , and the equations can be solved to obtain the unknown camera center c and rotation about the Z axis.

The main limitation of this approach is the need to measure the vertical direction of each view. More generally, the pitch and roll of each view with respect to

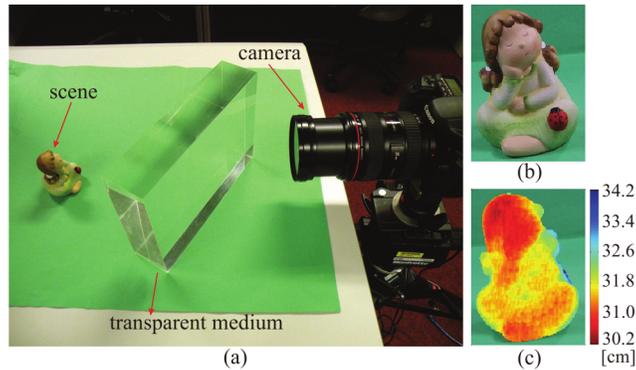


Figure 2.8: Depth from refraction method by Chen *et al.* [11], showing (a) the experimental setup, (b) an image of the scene without refraction, and (c) reconstructed depth.

a common refraction plane must be measured, and an IMU would be inadequate to do so if the refraction plane were not perpendicular to the earth’s gravity, as is the case with most underwater imaging systems. Moreover, the cameras used for underwater imaging may not share a common refraction plane.

In contrast to the previous two methods, in which refraction is modelled from multiple viewpoints, Chen *et al.* proposed a method for 3D reconstruction from a single viewpoint by manipulating the refractive object [11]. Figure 2.8 shows their experimental setup, in which a block of refractive material with parallel planar faces is placed between the camera and the scene. By capturing two images of the scene with and without the refractive object in place, the depth of point correspondences can be computed from the displacement due to refraction. The effect is similar to narrow-baseline stereo, although the image with refraction is not a single-viewpoint perspective projection [54] (see Section 2.3 below).

This method also has a calibration phase in which the pose of the refractive object is estimated. Due to the camera and the scene being in the same medium, the distance from the camera to the refractive object does not matter and only the

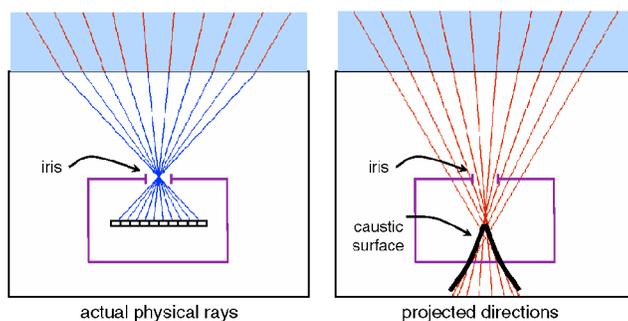


Figure 2.9: Rays imaged by a perspective camera in an underwater housing with a flat port. (Left) The physical rays imaged by the camera. (Right) The same rays if they did not undergo refraction.

object's rotation is needed. The main idea is that the images of a single point with and without refraction lie on a line that also contains the vanishing point of the normal direction of the refraction planes. This happens to be very similar to our idea described in Section 4.1.

2.3 Underwater Stereo

In this section we take a slightly different approach in which the refractive medium is considered as a part of the camera model, rather than a part of the scene. Such a situation arises naturally when cameras are mounted inside individual underwater housings. Compared to the methods described in 2.2.2, this is more general in that the refraction model is not restricted to a single plane shared by all cameras. The methods discussed here include those that focus on calibration, reconstruction, or both.

Although refraction had been considered in the field of photogrammetry for some time, in underwater applications the effect was often ignored, and allowed to be absorbed into the intrinsic camera parameters [47]. Some earlier work on

modelling refraction used camera housings with complex shapes, and focused on optimizing the resulting large set of parameters [34]. More recently, Treibitz *et al.* gave an analysis of the refraction effect of a single planar interface [54]. They showed that the bundle of rays imaged by a perspective camera through such a plane does not correspond to a single-viewpoint (SVP) camera. That is, when the rays on the far side of the interface are extended back toward the camera, they do not meet at a single point but instead form a caustic surface (Figure 2.9). It is therefore erroneous to model such a camera system as a perspective camera, even with radial distortion corrections (see 3.3).

Given that a physically-accurate refraction model should be used, Treibitz *et al.* proposed a method to obtain its parameters under some simplifying assumptions. First, only a single flat refraction is modelled, even though real underwater housings have glass of nonzero thickness. Second, the imaging plane is assumed to be parallel to the refraction plane. Third, calibration objects of known size must be placed parallel to the imaging and refraction planes, and at a known distance from the camera. Under these conditions and with at least two calibration objects imaged, their method is able to compute the distance from the camera center to the refraction interface using nonlinear optimization [54].

The assumptions in Treibitz *et al.*'s calibration method are quite restrictive and may be problematic for multi-view 3D reconstruction, for example if some cameras must be set at an angle to the refraction interface. Gedge *et al.* developed a more general calibration method that accounts for multiple cameras with independent refraction interfaces, and computes the interface angles in addition to the camera-to-interface distances [19]. The main idea is to perform the calibration on pairs of images, identifying corresponding feature points on a calibration

pattern and minimizing the perpendicular distance between back-projected rays (this is a nonlinear optimization). Furthermore, once the refraction parameters are obtained, Gedge *et al.* proposed a 3D reconstruction algorithm in which feature correspondences are matched along epipolar curves rather than epipolar lines. Unfortunately, the authors had difficulty obtaining accurate refraction parameters for more than two cameras using their method.

A similar calibration method was proposed by Sedlazeck and Koch with several differences: the camera housing is modelled as thick glass with two refractions, a calibration pattern is not required, and a specialized error function is used [44]. Their method proceeds by computing corresponding features in an underwater stereo image pair, and running an iterative sequence of nonlinear optimizations over the triangulated points, camera poses, and refraction parameters. An interesting aspect of this work is the derivation of a “virtual camera” error function, where each 3D point is projected using an imaginary perspective camera positioned on the refraction caustic surface (see above). This avoids the need to compute the refractively-projected point on each optimization iteration, and the authors claim improved efficiency. However, the runtime of this method reached 3 hours, and the results with real data are not compared to ground truth.

Kang *et al.* developed another optimization procedure for two-view calibration in the restricted case where the (single) refraction interface is assumed to be known or parallel to the image plane [31]. They showed that if the camera rotations are known, together with a set of image correspondences, then the minimization of reprojection error with respect to camera translations, 3D points, and camera-to-interface distances can be formulated as a convex optimization. Additionally, an algorithm based on Differential Evolution can be used to estimate

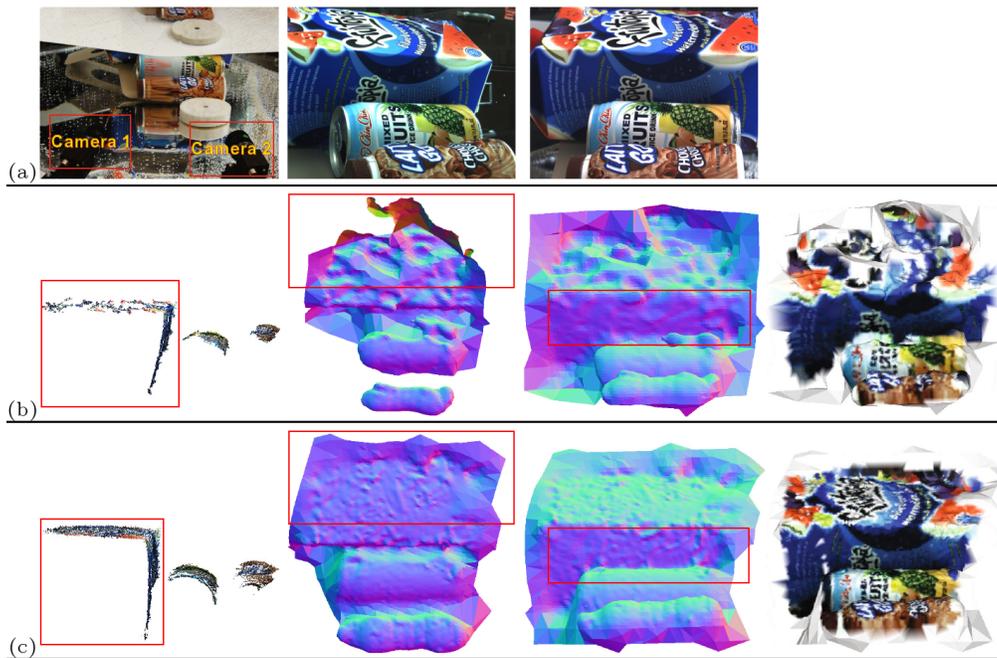


Figure 2.10: Two-view underwater 3D reconstruction results obtained by Kang *et al.* [31]. (a) The experimental setup. (b) Reconstruction without accounting for refraction. (c) Reconstruction using their method. The highlighted areas indicate where the result is less noisy and more geometrically correct compared to (b).

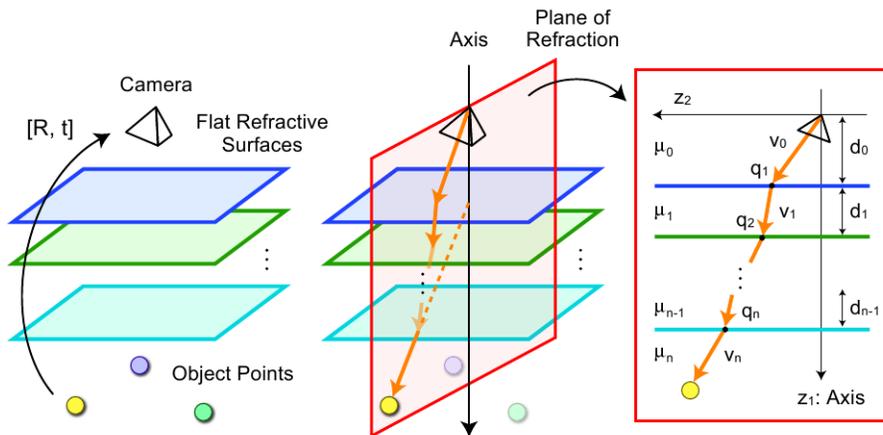


Figure 2.11: Flat refractive geometry used by Agrawal *et al.* [2]. (Left) A camera observing a scene through n flat refraction layers. (Middle) Each refracted light path lies entirely on a single plane, and all the planes intersect at a common axis. (Right) Layer thicknesses and refractive indices in the coordinate system of the light path plane.

the camera rotations, with the convex optimization as a sub-procedure to evaluate each trial. Figure 2.10 shows fairly good 3D reconstruction results that Kang *et al.* were able to obtain using this method and a modified version of PMVS, but they did not provide a quantitative comparison with ground truth.

To our best knowledge, the first method to fully and efficiently calibrate the parameters of a flat refraction camera model was proposed by Agrawal *et al.* [2]. Their work was based on the insight that such a model corresponds to an axial camera, where all of the imaged rays pass through a single line perpendicular to the refraction planes, which they call the “axis of refraction.” Furthermore, the entire path of a refracted light ray lies on a single plane called the “plane of refraction.” With this formulation, they showed that all the refraction parameters of a multi-layer flat refraction model can be computed by solving two systems of linear equations.

We will give a brief overview of their derivation, with reference to the concepts illustrated in Figure 2.11. In the first step, suppose a known 3D point \mathbf{P} (*e.g.* a point on a checkerboard pattern) in the scene is transformed into the camera coordinate system by unknown rotation R and translation \mathbf{t} . Let \mathbf{v}_0 be the direction of the initial ray from the camera for this point, and let \mathbf{A} be the direction of the axis. Then the “coplanarity constraint” states that the transformed point must lie on the plane of refraction containing both \mathbf{v}_0 and \mathbf{A} :

$$\text{Coplanarity: } (R\mathbf{P} + \mathbf{t})^\top (\mathbf{A} \times \mathbf{v}_0) = 0. \quad (2.8)$$

This equation can be rewritten to be linear in terms $[\mathbf{A}]_\times R$ and $\mathbf{A} \times \mathbf{t}$, from which all three unknowns can be recovered except for the translation component parallel to \mathbf{A} . The linear system can be constructed with either 11 points or 8 points, the latter using a solution based on essential matrix computation.

The second step relies on the axis of refraction being known. We now work in a new coordinate system with orthogonal basis $\{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3\}$ defined such that \mathbf{z}_1 is parallel to the axis of refraction, \mathbf{z}_2 lies on the plane of refraction, and $\mathbf{z}_3 = \mathbf{z}_1 \times \mathbf{z}_2$ (all vectors on the plane have zero third coordinate). Let \mathbf{p} represent \mathbf{P} transformed by R and \mathbf{t} and mapped into the new coordinate system. Writing \mathbf{v}_n for the final refracted ray direction from the camera, and \mathbf{q}_n for the ray intersection point with the last refraction interface, the equation

$$\mathbf{v}_n \times (\mathbf{p} + \alpha \mathbf{z}_1 - \mathbf{q}_n) = 0 \quad (2.9)$$

$$\Rightarrow \mathbf{v}_n \times (\alpha \mathbf{z}_1 - \mathbf{q}_n) = -\mathbf{v}_n \times \mathbf{p} \quad (2.10)$$

states that the last section of the light path is parallel to the final refracted ray direction. Here α represents an unknown shift along the axis direction, since this value was not recovered in the first step. Let $\mathbf{v}_1, \dots, \mathbf{v}_n$ be the directions of all the light path segments, and observe that they can be easily calculated from known \mathbf{v}_0 and \mathbf{A} . Let d_0, \dots, d_{n-1} be the unknown layer thicknesses and $c_i = \mathbf{v}_i^\top \mathbf{z}_1$, then we can express $\mathbf{q}_n = \sum_{i=0}^{n-1} \mathbf{v}_i \frac{d_i}{c_i}$ and

$$\mathbf{v}_n \times \begin{bmatrix} \frac{\mathbf{v}_0}{c_0} & \dots & \frac{\mathbf{v}_{n-1}}{c_{n-1}} & \mathbf{z}_1 \end{bmatrix} \begin{bmatrix} d_0 \\ \vdots \\ d_{n-1} \\ \alpha \end{bmatrix} = -\mathbf{v}_n \times \mathbf{p}. \quad (2.11)$$

This is the gist of Agrawal *et al.*'s linear solution to refractive calibration, and it is followed by nonlinear optimization to minimize the reprojection error. Unlike the method of Sedlazeck and Koch, the reprojection error is directly computed by solving an analytic equation, which is quartic for a single refraction and 12th-degree for two refractions. They also show how to extend the basic method to recover the refractive indices as well.

In a recent series of works, Jordt-Sedlazeck *et al.* made significant progress toward a system for underwater camera calibration and 3D reconstruction. They extended Agrawal *et al.*'s calibration method by introducing a genetic algorithm optimization stage coupled with an analysis-by-synthesis approach [27]. By optimizing not just the grid points on a checkerboard pattern, but rendering the full checkerboard and comparing every pixel with the captured image, they claim to achieve high accuracy in calibrating the refraction parameters. However, they did not compare their results with Agrawal *et al.*'s method, and their experiments with

real data did not have ground truth available.

Jordt-Sedlazeck *et al.* subsequently developed methods for 3D reconstruction accounting for refraction. Their approach for recovering dense depth maps, called refractive plane sweep, is based on back-projecting pixels from the captured images onto a series of hypothesized depth planes [26]. The back-projections are compared on the depth planes using normalized cross correlation (NCC) and sum of absolute difference (SAD) metrics, and a low discrepancy indicated by these metrics means that the depth hypothesis for that point is correct. The main advantage of this method appears to be efficiency, since refractive forward projection of points is avoided. For two views, fast runtimes of about 20 seconds are reported. Accuracy appears to be good with synthesized images, but once again no quantitative evaluation using real images with ground truth is performed.

Since their 3D reconstruction method assumes that all calibration parameters are known, Jordt-Sedlazeck and Koch also developed a bundle adjustment method to compute these parameters for image sequences [28]. They used a modified version of the virtual camera error function from their previous work [44] to derive an efficient optimization scheme with an analytic Jacobian. Simulated results show that their method achieves lower reconstruction and camera localization errors compared to those using a perspective camera model (*i.e.* ignoring refraction), but the errors still appear to be somewhat high given that the input images are noise-free. They also evaluate their method on real images, but the evaluation is mostly qualitative and lacks ground truth comparisons.

Chapter 3

Refractive Geometry

In this chapter we describe the mathematical foundations of our work. We follow a simple geometric optics approach in which we consider the propagation of light in terms of rays. Such an approach is common in the literature [2, 10, 28, 54]. First we cover some basic principles of refraction and dispersion, then we present the flat refraction model and derive some new properties that we use in calibration and 3D reconstruction.

3.1 Refraction

Refraction is the phenomenon that occurs when light passes through the boundary, or interface, between two dissimilar materials. Depending on the physical properties of the materials, the light waves will propagate at different speeds. The refractive index μ of a material is defined as the ratio of the speed of light c in

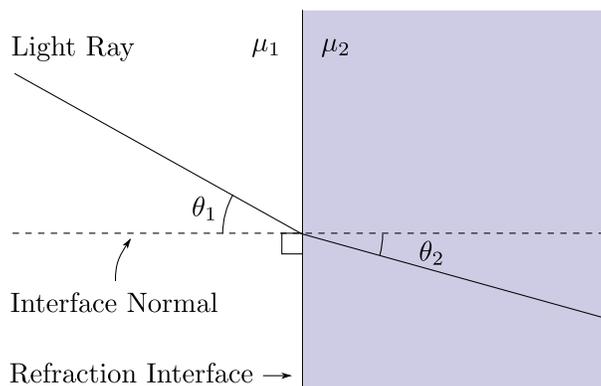


Figure 3.1: An illustration of Snell's Law. Here, $\mu_2 > \mu_1$.

vacuum to the speed of light v_{material} in the material:

$$\mu \triangleq \frac{c}{v_{\text{material}}} .$$

The result is that the light changes direction at the material interface, when we consider light in terms of rays. This effect can be derived in a number of ways, including Fermat's principle and Maxwell's equations, but for our purposes it suffices to consider a descriptive characterization in the form of Snell's law. Suppose a light ray passes through the interface of two materials with indices of refraction μ_1 and μ_2 , and makes angles of θ_1 and θ_2 with the normal direction on the corresponding sides of the interface (assume that the interface is locally planar). Snell's law states that:

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

This equation is illustrated in Figure 3.1. Some significant properties of refraction are:

- Snell’s law is symmetric with respect to which material the light is entering and exiting (this is known as the *Principle of Reversibility*). Therefore we may trace a ray backwards from the camera to the scene point from which it originates.
- The light path and interface normal lie entirely on a plane, regardless of the refraction angles.

Note that there are some materials that violate these properties, such as birefringent crystals. However, they are not usually found in imaging applications, so we will not consider them further.

Measuring the refractive index of a material is a fairly straightforward process requiring a precise setup. In this thesis we use a free online database of refractive indices [41] that draws its data from scientific sources, and we cite the original source where appropriate.

3.2 Dispersion

The speed of light propagation generally depends not only on the material it is passing through, but also on the frequency of the light itself. Therefore, the refractive index of a material is more correctly denoted as a function of frequency or, by common convention, a function of the wavelength of light in vacuum. Such functions can be highly complex, but within the wavelength range of visible light they can be modelled by the empirical Sellmeier equation of the form:

$$n^2(\lambda) = 1 + \sum_i \frac{B_i \lambda^2}{\lambda^2 - C_i} \quad (3.2)$$

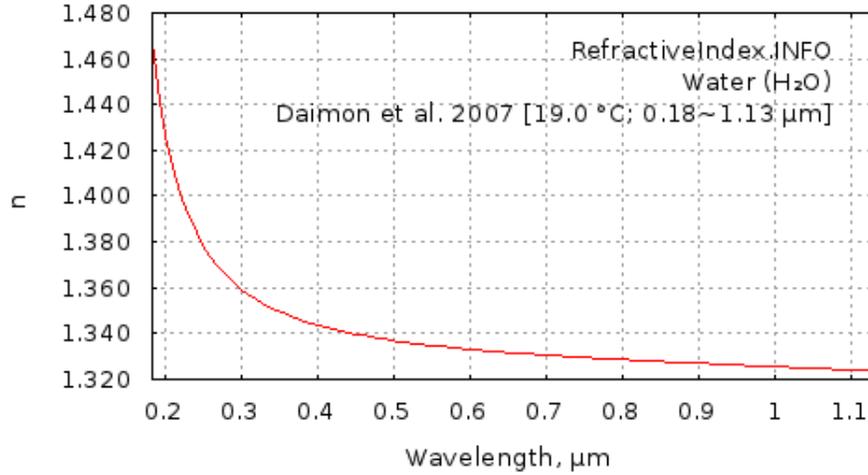


Figure 3.2: Refractive index of distilled water as a function of wavelength [14, 41].

where λ is the light wavelength. For common materials including water and glass, the coefficients B_i, C_i are positive numbers such that $n(\lambda)$ is a decreasing function. Figure 3.2 displays the refractive index of water from ultraviolet to near infrared wavelengths. Combining the equation for $n(\lambda)$ with Snell’s law (3.1), we see that when light passes from air to water, shorter wavelengths are refracted with a smaller angle than longer wavelengths. This is responsible for the “spreading out” of a light beam into the familiar rainbow pattern, and this phenomenon is called dispersion.

3.3 Flat Refraction Model

Our work is based on a model of refraction for parallel planar interfaces. It corresponds to a regular perspective camera placed in an underwater housing, featuring a flat glass port in front of the camera lens. While underwater imaging is the primary application for our work, the model can employ an arbitrary number of

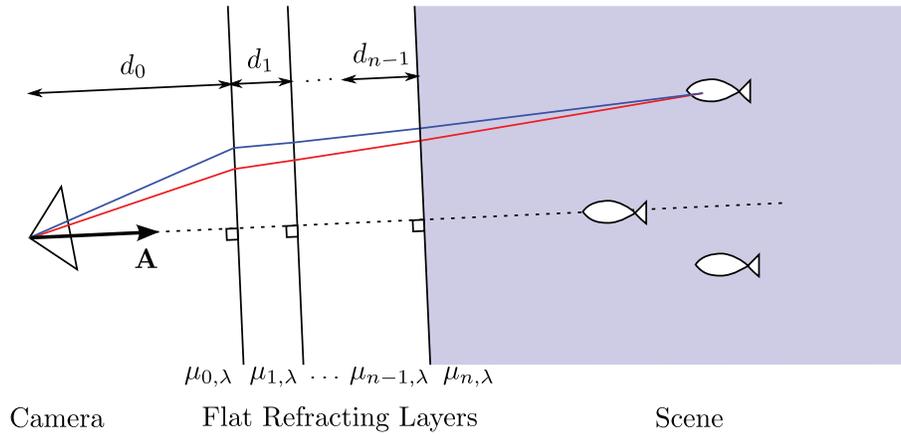


Figure 3.3: Flat refraction imaging model.

refraction planes. The same imaging model has previously been used by a number of different authors [2, 28, 19].

Figure 3.3 shows the components and parameters of the flat refraction model. A pinhole perspective camera observes a scene through $n+1$ refracting layers with parallel planar interfaces. We denote the refractive index of layer i for wavelength λ by $\mu_{i,\lambda}$, $i \in [0, n]$. Vector \mathbf{A} is the axis of refraction, which is the common perpendicular of the refraction interfaces passing through the camera center, and the layer thicknesses d_i are measured along this direction. Note that the optical axis of the camera need not be aligned with the axis of refraction. Also observe that because of the wavelength dependence of the refractive indices, light of different wavelengths emitted by the same point in the scene will take different paths to the camera.

3.3.1 Back Projection

Let $\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_n$ be the direction vectors for each segment along the light path from the camera to the scene. The task of computing \mathbf{v}_n from \mathbf{v}_0 , assuming all

flat refraction parameters are known, is similar to ray tracing [15]. Consider the first refraction from \mathbf{v}_0 to \mathbf{v}_1 , and let \mathbf{n} be the normal direction anti-parallel to \mathbf{A} , pointing from the refraction interface to the camera. Furthermore, write $\mathbf{m} = \frac{(\mathbf{n} \times \mathbf{v}_0)}{\|\mathbf{n} \times \mathbf{v}_0\|} \times \mathbf{n}$ for the direction of the component of \mathbf{v}_0 perpendicular to \mathbf{n} . We can decompose \mathbf{v}_1 into the components parallel and perpendicular to \mathbf{n} :

$$\mathbf{v}_1 = -(\mathbf{v}_1 \cdot \mathbf{n})\mathbf{n} + \|\mathbf{v}_1 \times \mathbf{n}\|\mathbf{m}. \quad (3.3)$$

Snell's law gives:

$$\|\mathbf{v}_1 \times \mathbf{n}\|\mathbf{m} = \frac{\mu_0}{\mu_1} \|\mathbf{v}_0 \times \mathbf{n}\|\mathbf{m} \quad (3.4)$$

$$= \frac{\mu_0}{\mu_1} (\mathbf{n} \times \mathbf{v}_0) \times \mathbf{n} \quad (3.5)$$

$$= \frac{\mu_0}{\mu_1} (\mathbf{v}_0 + \mathbf{v}_0 \cdot \mathbf{n}) \quad (3.6)$$

and

$$\mathbf{v}_1 \cdot \mathbf{n} = \sqrt{1 - \|\mathbf{v}_1 \times \mathbf{n}\|^2} \quad (3.7)$$

$$= \sqrt{1 - \frac{\mu_0^2}{\mu_1^2} \|\mathbf{v}_0 \times \mathbf{n}\|^2}. \quad (3.8)$$

Substituting into (3.3) and rearranging gives:

$$\mathbf{v}_1 = \frac{\mu_0}{\mu_1} \mathbf{v}_0 + \left(\mathbf{v}_0 \cdot \mathbf{n} - \sqrt{1 - \frac{\mu_0^2}{\mu_1^2} \|\mathbf{v}_0 \times \mathbf{n}\|^2} \right) \mathbf{n}. \quad (3.9)$$

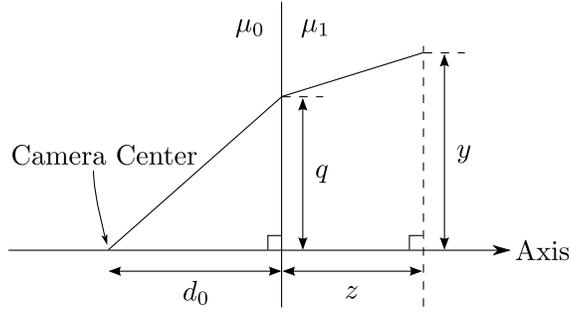


Figure 3.4: Formulation of forward projection of a scene point for one refraction.

Thus we can apply this formula recursively to obtain \mathbf{v}_n . The point \mathbf{q}_n where the light path intersects the refraction plane farthest from the camera is given by:

$$\mathbf{q}_n = \sum_{i=0}^{n-1} \mathbf{v}_i \frac{d_i}{-\mathbf{n} \cdot \mathbf{v}_i}. \quad (3.10)$$

3.3.2 Forward Projection

Computing the forward projection of a scene point onto the camera image plane is not a trivial task. Consider a single refraction of a ray as shown in Figure 3.4. Recall that the incident and refracted rays as well as the surface normal, which is the axis of refraction, always lie on a single plane called the plane of refraction. Therefore we can use a coordinate system on the plane of refraction, where y and z are displacements of the scene point in the perpendicular and parallel directions with respect to the axis, measured from the point where the axis intersects the refraction interface. Given the camera-interface distance d_0 and refractive indices μ_0, μ_1 (ignoring dispersion for the moment), q satisfies the following quartic equation [21]:

$$Nq^4 - 2Nyq^3 + \left(Ny^2 + \frac{z^2 \mu_1^2}{\mu_0^2} - d_0^2 \right) q^2 + 2d_0^2 yq - d_0^2 y^2 = 0 \quad (3.11)$$

where $N = \frac{\mu_1^2}{\mu_0^2} - 1$. Of the four possible solutions, there is exactly one in the interval $[0, y]$ that is the correct solution. Once the point (d_0, q) has been found, it is a simple matter to transform it back into the camera coordinate system and apply standard perspective projection to obtain the image point.

Agrawal *et al.* derived similar equations for the case of two refractions. They showed that when $\mu_0 = \mu_2$ the equation is a quartic polynomial, and when $\mu_0 \neq \mu_2$ the equation is a 12th-degree polynomial [2]. However, they did not perform such derivations for more than two refractions, and it appears that the degree of the equation grows very quickly. We implemented a more general forward projection technique based on optimization, described in Subsection 4.1.4.

3.3.3 Perspective Approximation

It has been noted in previous works that a regular perspective camera model can accommodate for the refraction effect to a significant degree [30, 47]. Two components of the model are mainly responsible, and these are the focal length and radial distortion parameters. Either of these components by itself can somewhat accommodate for refraction, but when calibrated together (*e.g.* using [4]), even better 3D reconstruction results can be obtained [30].

The first component of the perspective approximation, which is called “focal length adjustment,” is to apply a scaling factor to the true focal length of the camera. This is illustrated in Figure 3.5, showing two cameras with focal lengths f and f' and image planes parallel to the refraction plane. The first camera with focal length f images a physical ray that is refracted and crosses the image plane at point x , while the other camera observes an imaginary extension of the ray as if there were no refraction, but also images it at the same point x . The refraction

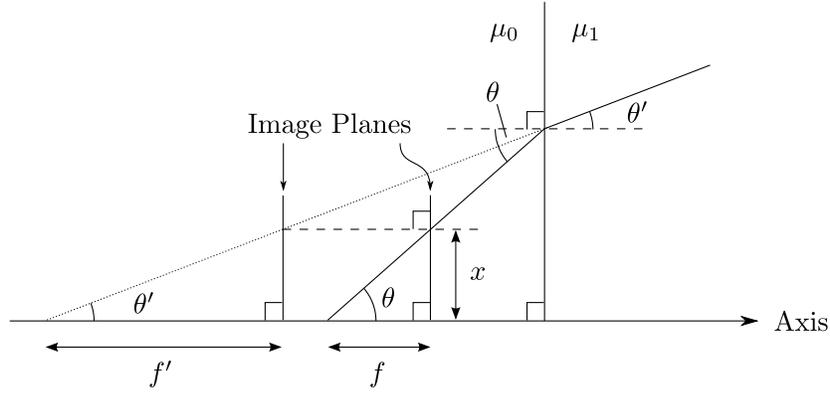


Figure 3.5: Approximate focal length adjustment for a single refraction.

angles θ , θ' are equal to the ray angles at the first and second cameras respectively, and relate x to the focal lengths f and f' :

$$\tan(\theta) = \frac{x}{f}, \quad \tan(\theta') = \frac{x}{f'}. \quad (3.12)$$

Using Snell's law and the approximation $\tan(\theta) \approx \theta \approx \sin(\theta)$ when the angle θ is small,

$$\mu_0 \sin(\theta) = \mu_1 \sin(\theta') \quad (3.13)$$

$$\Rightarrow \mu_0 \frac{x}{f} \approx \mu_1 \frac{x}{f'} \quad (3.14)$$

$$\Rightarrow \frac{f'}{f} \approx \frac{\mu_1}{\mu_0} \quad (3.15)$$

Therefore we might approximate the real camera by scaling the focal length f by a factor of $\frac{\mu_1}{\mu_0}$, and also shifting the camera center so that the imaginary ray is captured at the same image location. On the other hand, Treibitz *et al.* showed that these imaginary rays do not intersect at a single point [54], so the camera center shift will not be the same for all rays. For this reason, and because of the small

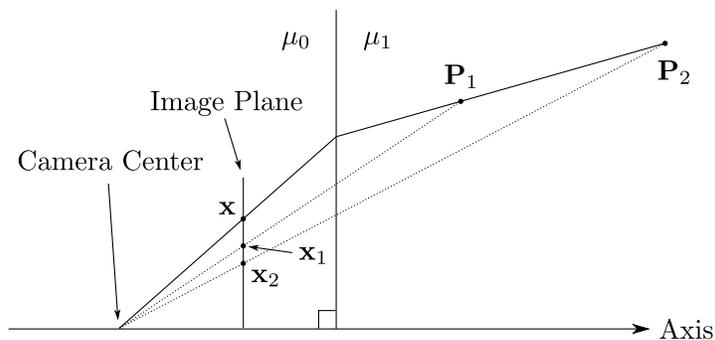


Figure 3.6: Failure of radial distortion to approximate refraction.

angle approximation and the assumption that the image and refraction planes are parallel, focal length adjustment is not a physically correct model of refraction.

The second component of radial distortion is ordinarily used to correct for lens imperfections that result in the image not being exactly a perspective projection. Simple distortion models correct for warping that is symmetric about a central point, and depends only on the radial distance from that point. Other models may have tangential warp parameters that account for imperfect alignment of lens elements. All such models, however, operate in image space and rely on the assumption of a single-viewpoint camera (*i.e.* all rays pass through a single point at the camera center). The radial distortion model used by Bundler [48], which was reported to give reasonable results for underwater 3D reconstruction [30], is:

$$\hat{\mathbf{x}} = (1 + k_1 \|\mathbf{x}\|^2 + k_2 \|\mathbf{x}\|^4) \mathbf{x}, \quad (3.16)$$

where \mathbf{x} is in normalized image coordinates (after perspective division), and k_1 and k_2 are the distortion parameters.

Regardless of the complexity of the radial distortion model, it cannot be an exact model of refraction, and the reason is illustrated in Figure 3.6. Given any

camera ray not parallel to the refraction axis, we can take two points P_1 and P_2 on the refracted ray, at different depths in the scene. For the radial distortion model to accommodate for refraction, it must warp the measured image point x to a position consistent with a perspective projection of the actual scene point. However, x could be the image of either P_1 or P_2 , which map to different image points x_1 and x_2 under perspective projection, and there is no way to determine if x should be warped to x_1 or x_2 based only on its image coordinates. Therefore, no distortion model can eliminate refraction to give a perspective image using information in the image alone.

It is worth noting that Figure 3.6 is highly exaggerated. When we apply focal length adjustment and radial distortion together, the shift in camera center means that the perspective projected rays are much closer to being parallel to the refracted rays. The radial distortion model can then adjust the rays to keep them close to the refracted rays over the desired depth range. Nevertheless, since this imaging model does not have a single viewpoint, the perspective and refracted rays eventually diverge which leads to erroneous results.

3.3.4 Dispersion Analysis

Our underwater camera calibration method exploits the dispersion of light to achieve high accuracy. We first analyze dispersion in the context of the flat refraction imaging model to show that the effect is in fact measurable with commonly available camera equipment.

The geometric construction that we use is shown in Figure 3.7 (left). A scene point at $(z, p + q)$ emits light at two wavelengths a and b . The refractive indices of the materials on the left and right are $\mu_{0,a}$ and $\mu_{1,a}$ respectively for wavelength

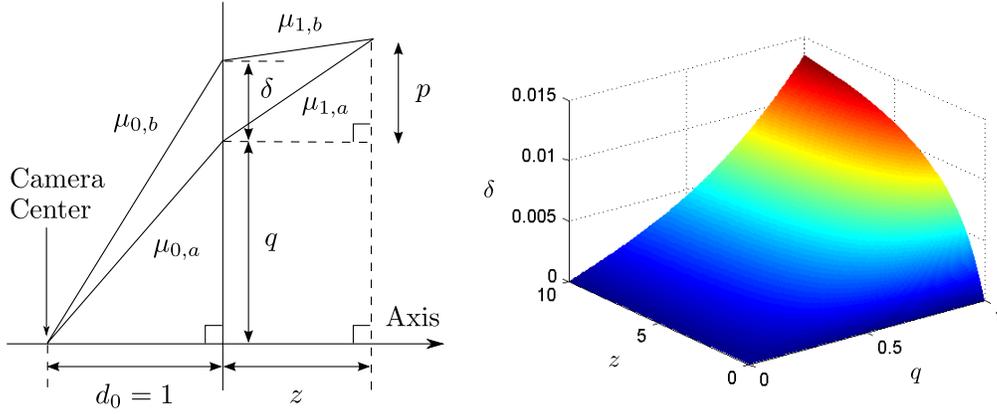


Figure 3.7: Dispersion of light in the flat refraction imaging model, for a single refraction. (Left) Geometric construction. (Right) Amount of dispersion for various settings of q and z , for $\mu_{0,a} = \mu_{0,b} = 1$, $\mu_{1,a} = 1.332$, $\mu_{1,b} = 1.343$.

a , and $\mu_{0,b}$ and $\mu_{1,b}$ for wavelength b . In this instance based on refraction from air to water, we assume $\mu_{1,b} > \mu_{0,b}$ and $\mu_{1,a} > \mu_{0,a}$. Suppose $\mu_{0,b}/\mu_{1,b} < \mu_{0,a}/\mu_{1,a}$ so that the refrax for wavelength b is $(0, q + \delta)$, $\delta > 0$, further from the axis than the refrax for wavelength a at $(0, q)$. Note that since the two refractive index ratios are different, we cannot have $\delta = 0$, for the refracted rays would diverge and never meet at a single scene point. Without loss of generality, we choose a scale such that $d_0 = 1$. Snell's law now gives:

$$\mu_{0,a} \frac{q}{\sqrt{q^2 + 1}} = \mu_{1,a} \frac{p}{\sqrt{p^2 + z^2}}, \quad (3.17)$$

$$\mu_{0,b} \frac{q + \delta}{\sqrt{(q + \delta)^2 + 1}} = \mu_{1,b} \frac{p - \delta}{\sqrt{(p - \delta)^2 + z^2}}. \quad (3.18)$$

Squaring and rearranging (3.18) gives

$$(\mu_{0,b}^2 - \mu_{1,b}^2)q_\delta^2 p_\delta^2 + \mu_{0,b}^2 q_\delta^2 z^2 - \mu_{1,b}^2 p_\delta^2 = 0 \quad (3.19)$$

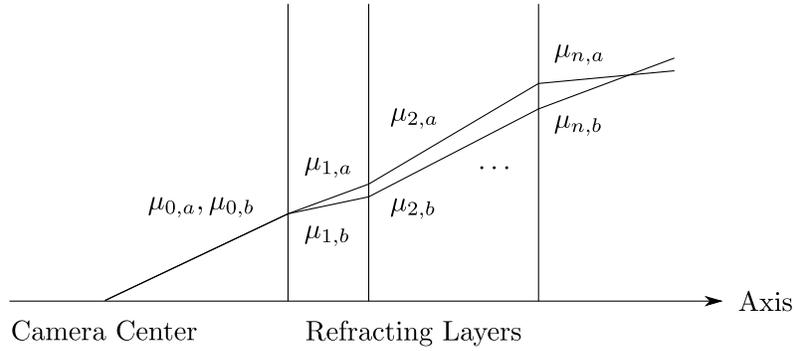


Figure 3.8: Dispersion of light in a multi-layer flat refraction imaging model.

where $q_\delta = q + \delta$ and $p_\delta = p - \delta$. Solving (3.17) for p and substituting into (3.19) gives a quartic equation in δ in terms of q , z , and the refractive indices.

According to Daimon *et al.*, the refractive index of distilled water at 19°C is 1.332 for 656nm light and 1.343 for 404nm light. Using these refractive indices, in Figure 3.7 (right) we plot δ over a range of values for q and z . For a typical camera with a resolution of 1024 pixels over a 60° field of view, $\delta = 0.005$ corresponds to about 4.4 pixels. We would observe this amount of dispersion toward the edge of the image ($q = 0.58$).

One last issue that we wish to address is dispersion in multilayer flat refraction systems. For two refractions from air to glass to water, we found that the amount of dispersion is quite similar to the single refraction case. Consider the general case with n refracting layers and two different wavelengths a and b . It is difficult to make a statement encompassing all possible optical materials, so we will limit ourselves to common materials in which the following assumptions hold:

1. The refractive indices vary with wavelength, such that $\mu_{i,a} \neq \mu_{i,b}$ for some layer i (see, *e.g.*, [41]); and
2. The Principle of Reversibility applies (Section 3.1).

Suppose, for the sake of argument, that a scene point not on the axis of refraction is observed under the two wavelengths, and exhibits no dispersion. That means the rays observed by the camera are superimposed, as illustrated in Figure 3.8. First note that the refractive indices in Snell's law telescope, so that the angle θ_i of the ray in layer i can be written in terms of only two refractive indices and the angle θ_0 at the camera:

$$\sin(\theta_i) = \frac{\mu_{i-1,a}}{\mu_{i,a}} \sin(\theta_{i-1}) \quad (3.20)$$

$$= \frac{\mu_{i-1,a}}{\mu_{i,a}} \cdot \frac{\mu_{i-2,a}}{\mu_{i-1,a}} \cdot \dots \cdot \frac{\mu_{0,a}}{\mu_{1,a}} \sin(\theta_0) \quad (3.21)$$

$$= \frac{\mu_{0,a}}{\mu_{i,a}} \sin(\theta_0) \quad (3.22)$$

and similarly for wavelength b . If the last refractive indices $\mu_{n,a}$ and $\mu_{n,b}$ are not equal, and we exclude the unlikely case that $\frac{\mu_{0,a}}{\mu_{n,a}} = \frac{\mu_{0,b}}{\mu_{n,b}}$, then the rays in the scene are non-parallel and intersect in at most one point, if at all. (If the refractive index dependency on wavelength is a decreasing function for all layers, as in typical optical materials over the visible spectrum, then there is no intersection point in the scene.) If there is no intersection point, this contradicts our assumption of a single scene point being observed; on the other hand, a single intersection point means that if we take the light path for wavelength a , any other point along the scene ray would require a different light path for wavelength b . This means that the rays at the camera will be superimposed for at most one scene point, but will diverge for any other point, and hence exhibit dispersion.

A similar argument can be made when $\mu_{n,a} = \mu_{n,b}$ and $\mu_{0,a} = \mu_{0,b}$, but now we consider that the light paths diverge in some intermediate refracting layer.

Excluding pathological cases where the light paths re-converge (such as finely tuned achromatic lens designs, which are also not flat refraction models), the rays in the scene will be parallel but will not coincide. The lack of an intersection point is once again a contradiction.

3.3.5 Geometric Constraints

The flat refraction imaging model possesses some mathematical properties that are useful for calibration. In this subsection we present three constraints, which are equations that the model parameters and scene points must satisfy in order to be physically correct.

Coplanarity Constraint

The coplanarity constraint was derived by Agrawal *et al.* [2]. We have described the specifics in Section 2.3, but reproduce the equation here for reference:

$$\text{Coplanarity : } (R\mathbf{P} + \mathbf{t})^\top (\mathbf{v}_0 \times \mathbf{A}) . \quad (3.23)$$

The scene point denoted by \mathbf{P} is transformed into camera coordinates by rotation R and translation \mathbf{t} , and it must lie on the plane containing the ray from the camera \mathbf{v}_0 and the axis of refraction \mathbf{A} .

To see why this is the case, recall that Snell’s law implies that when a light ray is refracted, the rays on both sides of the interface as well as the surface normal lie on the same plane. Since all refraction interfaces have the same normal in our imaging model, applying this property inductively shows that the entire light path lies on a single “plane of refraction” regardless of the number of refracting layers.

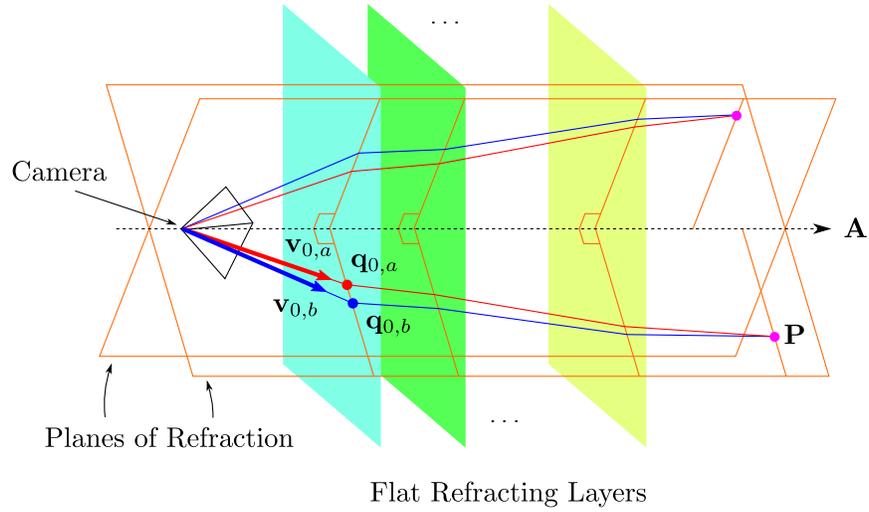


Figure 3.9: Light paths for scene points observed under two different wavelengths, yielding the dispersion and triangulation constraints.

It remains only to note that \mathbf{A} is (anti-)parallel to the interface normal.

Dispersion Constraint

We derive a new constraint that relates multiple wavelengths of light through dispersion. Observe that the argument for a plane of refraction containing the path of a refracted ray does not depend on the refractive indices. Indeed, the plane does not change even if the refractive indices are varied. Suppose that a single scene point emits two different wavelengths, then the coplanarity constraint implies that the light paths for both wavelengths must lie on the plane formed by the point and the axis of refraction. Let $\mathbf{v}_{0,a}$ and $\mathbf{v}_{0,b}$ be the initial ray directions from the camera for wavelengths a and b , and assuming that they correspond to the same scene point, we can write:

$$\text{Dispersion: } (\mathbf{v}_{0,a} \times \mathbf{v}_{0,b})^\top \mathbf{A} = 0. \quad (3.24)$$

Notice that this equation does not depend on the scene point, but that the normal of the plane of refraction is immediately given by $\mathbf{v}_{0,a} \times \mathbf{v}_{0,b}$, as long as the refractive indices vary with wavelength such that $\mathbf{v}_{0,a} \neq \mathbf{v}_{0,b}$ (see Subsection 3.3.4). Therefore, each observation of a scene point under two different wavelengths yields a linear constraint on the single unknown \mathbf{A} in the equation. In Subsection 4.1.1 we describe our method to deal with measurement noise and construct an overdetermined linear system.

Triangulation Constraint

In addition to coplanarity relationships, dispersion also gives a constraint on light path intersections. When a scene point is observed under different wavelengths of light, the light paths may be different but they all intersect at that point. This is similar to triangulation as described in Section 2.1, except that the multiple views of a point arise not by moving the camera center but from the different wavelengths. To formally state this as a constraint, we write:

$$\text{Triangulation : } \begin{cases} \mathbf{q}_{0,a} = F(\mathbf{P}, R, \mathbf{t}, \mathbf{A}, d_0, \dots, d_{n-1}, \mu_{0,a}, \dots, \mu_{n,a}) \\ \mathbf{q}_{0,b} = F(\mathbf{P}, R, \mathbf{t}, \mathbf{A}, d_0, \dots, d_{n-1}, \mu_{0,b}, \dots, \mu_{n,b}) \end{cases} \quad (3.25)$$

where F is a function that projects a point \mathbf{P} , using the given refraction model parameters, onto refrax points $\mathbf{q}_{0,a}$ and $\mathbf{q}_{0,b}$ on the refraction interface closest to the camera. All of the model and point parameters are the same in both projections except for the refractive indices, which depend on the wavelengths a and b . Since forward projection is nonlinear (see Subsection 3.3.2), this is a nonlinear constraint.

Chapter 4

Underwater Camera Calibration

Having introduced and developed some geometrical properties of the flat refraction imaging model, we now discuss how to apply them in a practical calibration method for underwater cameras. The importance of accurately estimating the imaging model parameters will become clear in our 3D reconstruction experiments in Chapter 5. Indeed, Gedge *et al.* found that an unreliable calibration was likely the main reason that their multi-view reconstructions did not yield good results [19].

In Chapter 2 we described several previous works on the calibration problem, including methods based on nonlinear optimization [19, 28, 44], hybrid convex and evolutionary optimization [31], and solving linear systems followed by nonlinear refinement [2, 27]. Our work follows the last approach and builds upon the significant contributions of Agrawal *et al.* [2], which we also use as a point of reference in evaluating our results.

4.1 Calibration Method

Recall from Section 2.3 that the calibration method introduced by Agrawal *et al.* [2] consists of three main steps. In the first step, the axis of refraction \mathbf{A} is found, together with the rotation R and translation perpendicular to the axis $\mathbf{t}_\perp = \mathbf{t} - (\mathbf{t}^\top \mathbf{A}) \cdot \mathbf{A}$ of the calibration object (such as a checkerboard pattern). In the second step, the layer thicknesses d_i and the calibration object translation $\mathbf{t}_\parallel = (\mathbf{t}^\top \mathbf{A}) \cdot \mathbf{A}$ parallel to the refraction axis are computed. (Optionally, the refractive indices μ_i can be computed as well.) In the last step, all of the model and calibration object parameters are refined using a nonlinear optimization that minimizes the root mean square (RMS) reprojection error, which is defined as:

$$\text{RMS}(\boldsymbol{\theta}) = \sqrt{\frac{1}{N} \sum_{j=1}^N \|\hat{\mathbf{x}}_j - \mathbf{x}_j\|^2} \quad (4.1)$$

where $\boldsymbol{\theta}$ represents the parameters, $\hat{\mathbf{x}}_j$ is the reprojection of point j of the calibration object onto the image using the refraction model, and \mathbf{x}_j is the corresponding measured image point.

The main novelty in our method is that we exploit the phenomenon of dispersion by capturing images using disparate wavelengths of light. An overview of the new method is as follows: the first step of the above procedure is split into two parts to compute the refraction axis and the calibration object parameters separately. Next, the remaining parameters are computed while incorporating dispersion constraints. Finally, nonlinear optimization is used to refine the estimated parameters, also using dispersion.

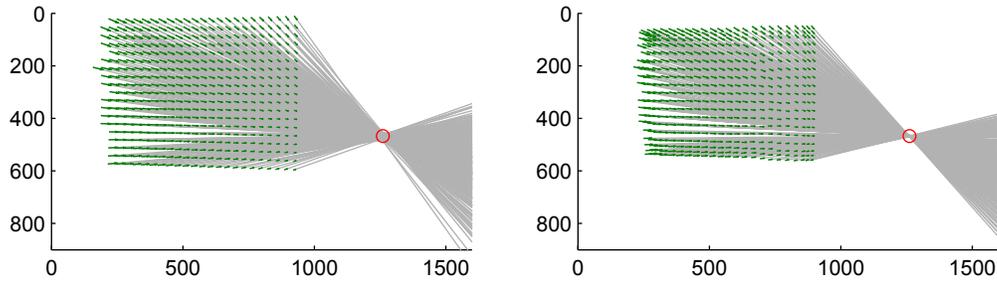


Figure 4.1: Effect of applying a local neighbourhood average over dispersion point pairs for estimating the refraction axis. Each point pair is shown as a small green arrow and extrapolated into a gray line. The projection point of the refraction axis, as estimated using the dispersion constraint, is marked with a red circle. (Left) Without averaging. (Right) With averaging. Notice how the lines focus more tightly at a single point. A side-effect is that the green arrows at the edges are drawn slightly toward the center.

4.1.1 Refraction Axis

The dispersion constraint (3.24) can be used directly to form a system of linear equations, given two or more scene points observed under two different wavelengths. In our work we construct an overdetermined system and solve it using the method of least squares. The residual being minimized has a geometric meaning, being the sine of the angles between the refraction axis and each plane of refraction. Also note that the estimation of the refraction axis \mathbf{A} is *independent* of any scene point parameters, and the points used in this step do not need to be part of a calibration object. By contrast, the method of Agrawal *et al.* simultaneously solves for \mathbf{A} and the calibration object pose [2], even though these parameters are not inherently related.

One caveat to using the dispersion constraint, as the analysis of Subsection 3.3.4 shows, is that the amount of dispersion is small, typically on the order of a few pixels. In our method we improve the signal-to-noise ratio by averaging over neighbouring points. A scene point is captured as a pair of image points \mathbf{x}

and $(\mathbf{x} + \mathbf{w})$, corresponding to rays $\mathbf{v}_{0,a}$ and $\mathbf{v}_{0,b}$ for wavelengths a and b . For convenience of notation, let image points be in \mathbb{R}^3 with zero third coordinate. We wish to compute a new point pair $\{\mathbf{y}, (\mathbf{y} + \bar{\mathbf{w}})\}$ based on the set of measured point pairs $\{\mathbf{x}_i, (\mathbf{x}_i + \mathbf{w}_i)\}_{i=1:k}$ in the local neighbourhood.

First we note that by definition, the axis of refraction \mathbf{A} passes through the camera center, so that any point along \mathbf{A} projects to the same image point \mathbf{u} . Consider any point pair $\{\mathbf{x}_j, (\mathbf{x}_j + \mathbf{w}_j)\}$ and the corresponding ray directions $\mathbf{v}_{0,a}^j, \mathbf{v}_{0,b}^j$. Since perspective projection transforms lines to lines, we can construct a line by joining any point along \mathbf{v}_a^j to any point along \mathbf{A} , and the projection will be a line in image space passing through \mathbf{x}_j and \mathbf{u} . Now we know that $\mathbf{v}_{0,a}^j, \mathbf{v}_{0,b}^j$, and \mathbf{A} lie on the same plane of refraction. Therefore, in the absence of noise, the line we constructed intersects the ray $\mathbf{v}_{0,b}^j$ so that its image passes through $(\mathbf{x}_j + \mathbf{w}_j)$ as well.

Each pair of points $\{\mathbf{x}_i, (\mathbf{x}_i + \mathbf{w}_i)\}$ defines a line in the image with direction \mathbf{w}_i . The preceding discussion shows that when the points are uncorrupted by noise, all of these lines intersect at \mathbf{u} . Let $\bar{\mathbf{w}} = \frac{1}{k} \sum_{i=1}^k \mathbf{w}_i$ be the average direction, then we have that

$$(\mathbf{u} - \mathbf{x}_i) \times \mathbf{w}_i = 0 \quad \forall i \Rightarrow \mathbf{u} \times \bar{\mathbf{w}} = \frac{1}{k} \sum_{i=1}^k \mathbf{x}_i \times \mathbf{w}_i, \quad (4.2)$$

where the implication follows by summing over i . Now we constrain the line $\{\mathbf{y}, (\mathbf{y} + \bar{\mathbf{w}})\}$ to also pass through \mathbf{u} :

$$(\mathbf{u} - \mathbf{y}) \times \bar{\mathbf{w}} = 0 \Rightarrow \mathbf{u} \times \bar{\mathbf{w}} = \mathbf{y} \times \bar{\mathbf{w}}. \quad (4.3)$$

Substituting (4.3) into (4.2) eliminates \mathbf{u} and gives a linear equation for \mathbf{y} . There

are an infinite number of solutions, so we find one that is “close” to the original point pair by choosing \mathbf{y} on the line perpendicular to $\bar{\mathbf{w}}$ and passing through the centroid of the point neighborhood. The image points $\{\mathbf{y}, (\mathbf{y} + \bar{\mathbf{w}})\}$ are then back-projected into rays and used in the dispersion constraint.

Figure 4.1 displays an example of the effect of applying the averaging technique to dispersion data from a real image. In practice we found that good results were obtained by setting the neighbourhood radius to about 6% of the image width, typically encompassing around 20 point pairs.

4.1.2 Calibration Object Pose

Part of the calibration object pose parameters, namely the rotation R and translation \mathbf{t}_\perp perpendicular to the axis, are needed in the next step to estimate refraction layer thicknesses. Whereas in Agrawal *et al.*'s method these are computed together with \mathbf{A} by applying the coplanarity constraint (2.8), we show that the equations can be simplified considerably when \mathbf{A} is known. The constraint is first rewritten as follows [2]:

$$(R\mathbf{P} + \mathbf{t})^\top (\mathbf{A} \times \mathbf{v}_0) = 0 \quad (4.4)$$

$$\Leftrightarrow \underbrace{\begin{bmatrix} \mathbf{P}^\top \otimes \mathbf{v}_0^\top & \mathbf{v}_0^\top \end{bmatrix}}_B \begin{bmatrix} E_{(1:9)} \\ \mathbf{s} \end{bmatrix} = 0 \quad (4.5)$$

where \otimes denotes the Kronecker product, $E = [\mathbf{A}]_\times R$ and $\mathbf{s} = \mathbf{A} \times \mathbf{t}$, and subscripts in parentheses denote matrix elements taken column-wise. Since the refraction axis is known, we can rotate the coordinate system such that \mathbf{A} is aligned with the positive z-axis. Let r_1, \dots, r_9 be the entries of R , then with such a transfor-

mation we have

$$E = \begin{bmatrix} -r_2 & -r_5 & -r_8 \\ r_1 & r_4 & r_7 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{s} = \begin{bmatrix} -t_2 & t_1 & 0 \end{bmatrix}^\top. \quad (4.6)$$

Furthermore, for a planar calibration object such as a checkerboard where $\mathbf{P}_{(3)} = 0$, columns 7-9 of B are zero (Eqn. (4.5)). Therefore we can drop the last row and the last column of E to obtain the simplified constraint:

$$\begin{bmatrix} \mathbf{P}_{(1:2)}^\top \otimes \mathbf{v}_{(1:2)}^\top & \mathbf{v}_{(1:2)}^\top \end{bmatrix} \begin{bmatrix} E_{(1,2,4,5)} \\ \mathbf{s}_{(1:2)} \end{bmatrix} = 0. \quad (4.7)$$

Stacking five or more equations and solving the resulting linear system yields four entries of R and the translation perpendicular to the axis, up to an unknown scale factor. We recover the scale factor and the remaining entries of R by solving the quadratic constraint that it is an orthogonal matrix.

In general there are four possible solutions. The sign of the scale factor corresponds to one solution in front of the camera and one behind. Two further solutions are obtained by negating the signs of r_8 and r_7 , corresponding to a reflection across the plane parallel to the refraction layers and passing through the object origin. The correct solution is found after estimating the refraction layer thicknesses by choosing the one with the minimum reprojection error.

4.1.3 Refraction Layer Thicknesses

To obtain the thicknesses d_0, \dots, d_{n-1} of the refraction layers, we use the equation (2.11) developed by Agrawal *et al.* to construct a linear system. The difference is that we incorporate the triangulation constraint (3.25) implicitly by adding one equation for each observed wavelength for each scene point. For example, using the same notation as in Section 2.3 but adding subscripts for two wavelengths a and b , we have the following set of equations for each point \mathbf{p} :

$$\begin{bmatrix} \mathbf{v}_{n,a} \times \begin{bmatrix} \frac{\mathbf{v}_{0,a}}{c_{0,a}} & \dots & \frac{\mathbf{v}_{n-1,a}}{c_{n-1,a}} & \mathbf{z}_1 \end{bmatrix} \\ \mathbf{v}_{n,b} \times \begin{bmatrix} \frac{\mathbf{v}_{0,b}}{c_{0,b}} & \dots & \frac{\mathbf{v}_{n-1,b}}{c_{n-1,b}} & \mathbf{z}_1 \end{bmatrix} \end{bmatrix} \begin{bmatrix} d_0 \\ \vdots \\ d_{n-1} \\ \alpha \end{bmatrix} = - \begin{bmatrix} \mathbf{v}_{n,a} \times \mathbf{p} \\ \mathbf{v}_{n,b} \times \mathbf{p} \end{bmatrix} \quad (4.8)$$

where $c_{i,\lambda} = \mathbf{v}_{i,\lambda}^\top \mathbf{z}_1$. Note that this is really only two equations since all the vectors lie on the plane of refraction and have zero third coordinate in that coordinate system. After constructing and solving the system of equations, α corresponds to the translation \mathbf{t}_{\parallel} parallel to \mathbf{A} , which is simply t_3 in the refraction-axis-aligned coordinate system of Subsection 4.1.2.

4.1.4 Nonlinear refinement

The last step in our calibration method is to apply nonlinear optimization over all the parameters \mathbf{A} , d_i , R , and \mathbf{t} with respect to the reprojection error (4.1). We use the MATLAB function `lsqnonlin` to do this. Because we reproject each scene point using all of the observed wavelengths, the optimization implicitly tries to satisfy the triangulation constraint (3.25).

Computing the reprojection error requires a forward projection for each scene point. Rather than solving for the roots of polynomial equations as done by Agrawal *et al.* [2], our approach is to perform a one-dimensional optimization using the equations for back projection. The main advantage is that this approach can accommodate any number of refraction layers, even as the polynomial equations increase significantly in complexity (4th degree for one refraction, 4th or 12th degree for two refractions, and currently unknown degree for three or more refractions). Although we did not perform benchmark tests, we did not notice any considerable performance loss compared to the polynomial solving approach for two refractions, and accuracy is limited only by machine precision.

The details of our forward projection implementation are as follows. Recall from Subsection 4.1.1 that when a scene point is imaged under two different wavelengths of light, the two resultant image points lie on a line passing through the image point of the refraction axis. This holds even for a hypothetical wavelength that does not refract at all, *i.e.* the line also passes through the image of the scene point under standard perspective projection. Thus, given a scene point and the axis of refraction, we simply construct this line in the image and search along it for the point whose back projection (Subsection 3.3.1) intersects the scene point.

The error function that we use is the signed distance ϵ between the scene point and the back-projected ray, measured on the depth plane of the scene point. If x is the image position of the scene point under standard perspective projection, it can be shown that either $[x, x + \epsilon]$ or $[x - \epsilon, x]$ brackets the correct solution, depending on the sign of ϵ . Thus we have a very simple root-finding problem with a strictly monotonic function, which we solve using Brent's method as implemented in the MATLAB function `fzero`. Further optimizations may be possible, but we ob-

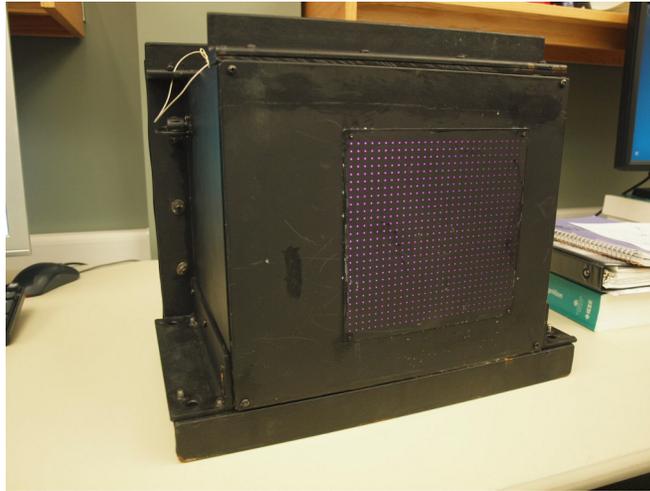


Figure 4.2: The calibration device that we constructed to emit two wavelengths of light in a known spatial pattern.

served that this method is already fast, taking only around 7 iterations on average.

4.2 Calibration Device

4.2.1 Design and Construction

Thus far we have presented a theoretical calibration framework based on imaging points in a scene at selective wavelengths of light. A potential way to achieve this in practice is to use special scene illumination and/or spectral filters placed in front of the camera, together with a reflective calibration object such as a checkerboard pattern. We found it more convenient, however, to build a specialized calibration device that emits its own light at the desired wavelengths.

Figure 4.2 shows the assembled calibration device. It consists of a watertight acrylic box with LED lighting elements inside. The box is painted all black, except for a rectangular area on one side from which the light is emitted. Over

the light emitting area we mounted a precision CNC-drilled plate with a 27×29 grid pattern of 0.65mm diameter holes spaced 6mm apart. Each hole in the plate thus restricts the emitted light to a narrow point, yet combining all of the light wavelengths produced inside the box.

The lighting elements consist of two types of LEDs, one emitting light at 660nm^1 and the other at 405nm^2 . Both were tested using a spectrometer to confirm the dominant wavelength specifications. We chose these wavelengths to be as far apart as possible to maximize the dispersion effect, while remaining visible to typical cameras equipped with a CCD or CMOS sensor and a Bayer-pattern color filter array (CFA). Longer wavelengths in the infrared region suffer from high attenuation in water, while ultraviolet rays are potentially hazardous.

One issue we encountered is that achieving uniform lighting is not trivial. Lighting uniformity is important to avoid over- or under-exposure, which can interfere with our point localization method described in Subsection 4.2.2. The LEDs inside the box are arranged in a grid pattern, with 122 in total split evenly between the two types. Our solution was to use a series of diffuser films placed in front of the LED grid, but this resulted in the box becoming quite large. As a result, a significant amount of ballast was also needed to make it sink in water. We anticipate that more sophisticated construction techniques will reduce the size and weight of the calibration device.

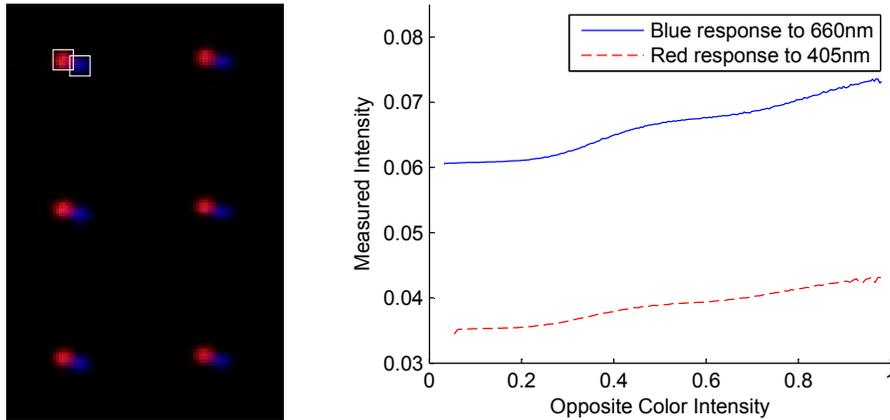


Figure 4.3: (Left) Close-up of an image of six illuminated points on the calibration device, showing the dispersion caused by an air-glass-water interface. Best viewed in color; one red-blue pair is highlighted for visibility. (Right) Sensitivity of red and blue pixels to 660nm and 405nm light. See text for details.

4.2.2 Point Detection and Localization

For simplicity, our calibration device emits two wavelengths of light at the same time. This means we have to demultiplex the signals, which is made possible by the CFA mounted on most color cameras. (An alternative would be to time-multiplex the signal, but errors might be introduced if the calibration device or camera moves.) We simply capture images in a raw format, before demosaicing is applied, and isolate the two wavelengths by considering the red and blue pixels separately.

To determine if there is significant leakage of 405nm light onto red pixels and 660nm onto blue pixels, we conducted tests by photographing each type of LED individually in a dark room, and correlating the measurements of each pixel with neighbours of the opposite color. Figure 4.3 (right) displays the result of these tests. For 660nm light, as the intensity measured by red pixels increases from

¹Lumex SSL-LX5093SRC/E

²Bivar UV5TZ-405-30

near-zero to saturation, the neighbouring blue pixels register an increase of less than 1.5%. Similarly, red pixels register less than 1% response to 405nm light. These numbers are low enough that we consider the crosstalk between the two color channels to be negligible.

The captured images are processed in three steps: points on the calibration pattern are first detected, then the sub-pixel center location of each point is found, and lastly the overall grid pattern is inferred. We detect points with a combination of thresholding and morphological dilation, which is simple and works most of the time (false positives occur occasionally, and are manually removed).

To refine the accuracy of the detected point locations, we approximate the point spread function as a bivariate Gaussian and fit this distribution to the measured pixel values. An initial estimate is obtained by extracting a square window around each point. Let \mathbf{x}_i be the coordinate vector of the i^{th} pixel in the window and $I(\mathbf{x}_i)$ be its intensity, then we construct a covariance matrix according to:

$$C = \frac{\sum_i I(\mathbf{x}_i) \cdot \mathbf{x}_i \mathbf{x}_i^\top}{\sum_i I(\mathbf{x}_i)}. \quad (4.9)$$

The eigenvectors of C give the orientation of the Gaussian distribution, and the eigenvalues give the variance in the corresponding directions. We use the MATLAB function `fminsearch` to perform the final optimization.

The last step of inferring the grid pattern is needed because we found that the intensity of the points varied significantly, especially when the calibration device was set at an angle. To prevent overexposure of the brightest points, some points around the periphery would be too dim to measure, resulting in an incomplete grid. We implemented a method to detect and scan along the horizontal and vertical grid

axes, thereby recovering the object coordinates needed for calibration (Subsection 4.1.2).

4.2.3 Chromatic Aberration Correction

In lens-based optical systems, dispersion occurs within the lens elements themselves as they refract light. This effect is known as Chromatic Aberration (CA) and results in visible color fringes around sharp edges in the image. It is important to compensate for CA in order to isolate the dispersive effect of the refractive planes in front of the camera.

For this purpose, we use the image distortion model introduced by Brown [7]. We first capture images of our calibration object in air, and obtain two sets of camera intrinsic parameters using the red and the blue channels separately. Then we apply a nonlinear optimization to adjust the image distortion parameters of the blue channel so that the blue points are aligned with the red points. The choice of which color channel to warp is arbitrary as long as we do so consistently with the underwater images.

Our tests suggest that this image warping CA correction approach is effective. For the camera used in our experimental results, which has a resolution of 4368×2912 pixels, the average distance between aligned red and blue points was 0.05px.

4.3 Experimental Results

We evaluated our calibration method through experiments using both procedurally generated synthetic data and real data captured with a camera. We used a Canon

5D camera with a resolution of 4368×2912 pixels and a 30mm fixed focal length lens. The intrinsic parameters were obtained using the MATLAB Camera Calibration Toolbox [4] and used to generate the synthetic data. For the real data, we applied the chromatic aberration correction method described in Subsection 4.2.3. Our results are compared Agrawal *et al.*'s method using the publicly-available source code accompanying their paper³.

4.3.1 Synthetic Data

Our synthetic data experiments consisted of three different configurations for the refraction model as listed in Table 4.1. All refractive indices are assumed known and are listed in table 4.2.

	Refractions	d_0	Est.?	d_1	Est.?
I	Air→Water	0.06	Yes	–	–
II	Air→Acrylic→Water	0.06	Yes	0.0056	No
III	Air→Acrylic→Water	0.06	Yes	0.03	Yes

Table 4.1: Configurations for synthetic data experiments, showing the values for parameters d_0 and d_1 and whether they were estimated during calibration or assumed known. The angle between the refraction axis and the camera's optical axis was set to 4.47 degrees for all configurations.

Wavelength	Water	Acrylic
660nm	1.33151	1.488
405nm	1.34318	1.516
Unspecified/589nm	1.33344	1.491

Table 4.2: Refractive indices used in all experiments. Sources are [14] for water and [32] for acrylic.

The calibration pattern was a 27×29 planar grid of points emitting both 405nm and 660nm light. Since the use of two wavelengths in our method could be con-

³<http://www.umiacs.umd.edu/aagrwal/cvpr12/FlatRefraction.html>

strued as doubling the number of points, to provide a fair comparison we used a 39×40 grid with Agrawal *et al.*'s method. Moreover, all the points in the 39×40 grid used 405nm light to remove any advantage due to stronger refractions. For a more direct comparison with our real data results, we also included a dataset for a 34×35 grid to determine the impact of using a smaller number of points.

All feature points were perturbed by random Gaussian noise with standard deviation ranging from 0 to 1 pixel. For each noise level we generated 100 trials with the calibration pattern placed 0.44 units in front of the camera and rotated randomly by up to 20 degrees.

Figures 4.4 and 4.5 summarize the results. We include results before the non-linear refinement step to compare the effectiveness of the dispersion and triangulation constraints. For Agrawal *et al.*'s method the corresponding estimates are computed by the 8-point algorithm and the linear system for recovering layer thicknesses [2].

Our calibration method appears to give more accurate results overall for every refraction model configuration. For configuration I we see that the initial estimates for both d_0 and \mathbf{A} are noticeably better using our method, indicating that the dispersion and triangulation constraints are indeed effective. Agrawal *et al.*'s method is helped by the nonlinear refinement step, which narrows the gap considerably, but we see that the implicit triangulation constraint in our method still yields improved accuracy.

Turning to the results for configuration III, the difference between the two methods is particularly striking. The layer thickness error plots for Agrawal *et al.*'s method rise sharply as the noise increases and the final estimates are far from the ground truth, whereas our method is able to obtain reasonable estimates

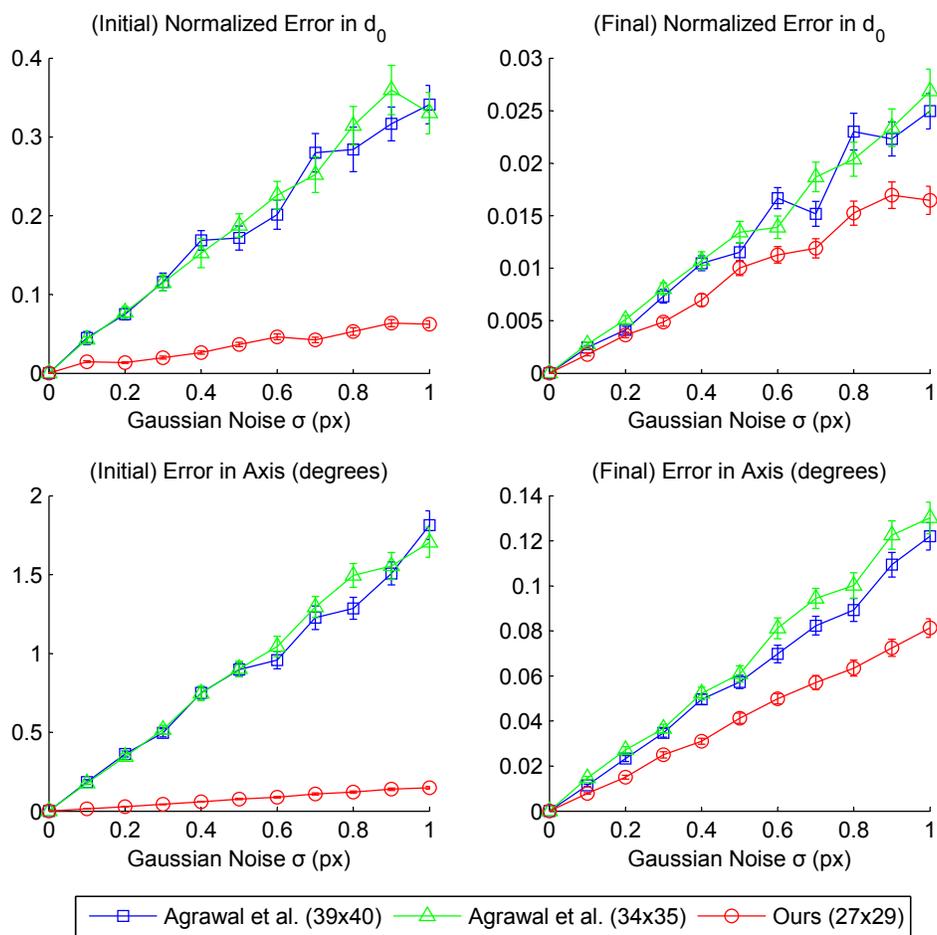


Figure 4.4: Synthetic data results for configuration I. Error bars represent standard error. Results for configuration II are very similar and are omitted. (Left column) Before nonlinear refinement. (Right column) After nonlinear refinement.

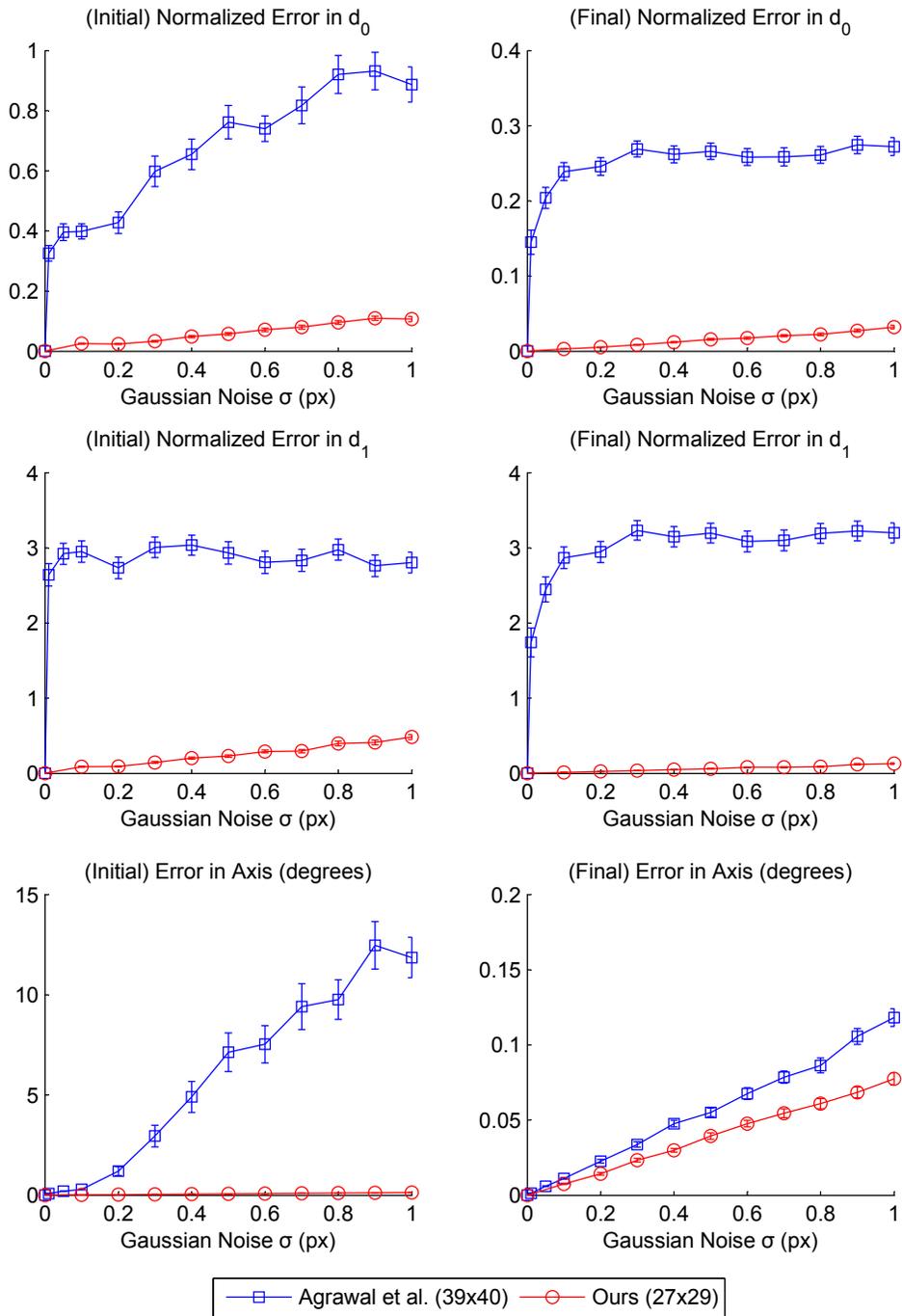


Figure 4.5: Synthetic data results for configuration III. Error bars represent standard error. Plots for Agrawal *et al.*'s method contain two additional data points at $\sigma = 0.01, 0.05$. (Left column) Initial estimates before nonlinear refinement. (Right column) After nonlinear refinement.

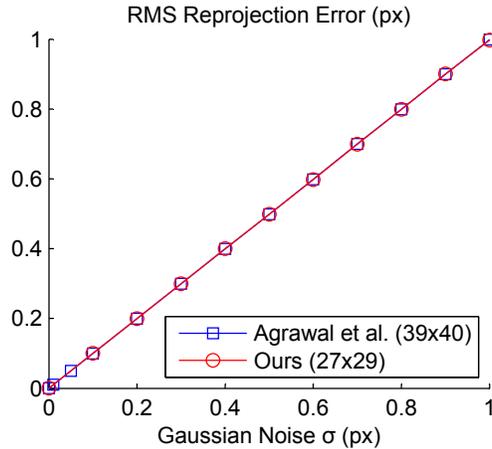


Figure 4.6: Reprojection error after nonlinear refinement for synthetic data configuration III. Error bars represent standard error. Plots for Agrawal *et al.*'s method contain two additional data points at $\sigma = 0.01, 0.05$.

for all three parameters. Intuitively, this configuration is challenging because the distortion caused by an additional refraction is slight compared to the difference between refraction and no refraction. Indeed, some authors including Agrawal *et al.* have suggested that multi-layer refractions are well approximated by a single refraction [2, 51]. In practice it is often the case that some layer thicknesses can be measured directly (*e.g.* the housing for an underwater camera), but when this is not possible, it seems clear that our method should be favored.

A possible contention is that our implementation of Agrawal *et al.*'s method for configuration III is incorrect, since only configuration I is included in the original source code. We believe that the implementation is correct because the error goes to zero in the absence of noise, the refraction axis estimates appear reasonable, and the reprojection error is being minimized properly (Figure 4.6). Unfortunately, Agrawal *et al.*'s work [2] does not give layer thickness estimation results for this configuration for us to corroborate our findings.

In light of our results, we believe that the additional constraints provided by



Figure 4.7: Experimental setup, showing the calibration device and a checkerboard inside the tank, a checkerboard affixed to the tank surface, and the camera mounted on a SliderDolly translation rail.

dispersion have a significant impact on calibration accuracy beyond simply doubling the number of feature points.

4.3.2 Real Data

We carried out an experiment by placing our calibration device inside a filled water tank with transparent acrylic sides. The camera was mounted outside the tank, facing inwards. The setup can be seen in Figure 4.7.

Obtaining Ground Truth

In order to approximate the intended use case of underwater imaging, we positioned the camera so that the lens barrel was within 0.1m of the surface of the tank. With such a setup, we could not obtain the ground truth refraction parameters by placing checkerboard patterns on the tank surface, since it was too close for the camera to focus.

Our solution was to mount the camera on a SliderDolly translation rail, which allowed us to move the camera precisely in a straight line. We also attached a ruler with a needle to accurately measure the camera’s motion along the rail. By moving the camera backward a certain distance, we could then capture an image of a checkerboard pattern on the tank surface and determine its pose with respect to the camera. The final ground truth values were found from 48 such pose estimates, averaging the rotations⁴ and fitting a linear scale of the measured distances to extrapolate the original camera position.

Calibration Experiment

We collected two sets of data, one using our novel calibration device which has a 27×29 grid of points, and a second using a 34×35 checkerboard pattern. The calibration objects were placed approximately 0.45m behind the front surface of the water tank and moved around slightly within the camera’s field of view. We used an air→acrylic→water refraction model in all calibrations, with the tank wall thickness assumed known at a measured value of 5.6mm. We remark that the checkerboard in this experiment has somewhat fewer points than the 39×40 grid used in our synthetic data experiments. This was simply to keep the squares of the checkerboard from being too small, and the results in Figure 4.4 show that the impact of using a slightly different number of points is minimal.

Table 4.3 summarizes the calibration results that we obtained. Additionally, Figure 4.8 gives a visual representation of the refraction axis estimation results, plotting the projection point of each estimated axis. (Since the refraction axis passes through the camera center, its image is a single point.) There are two

⁴We used the SVD method described by Curtis *et al.* [13].

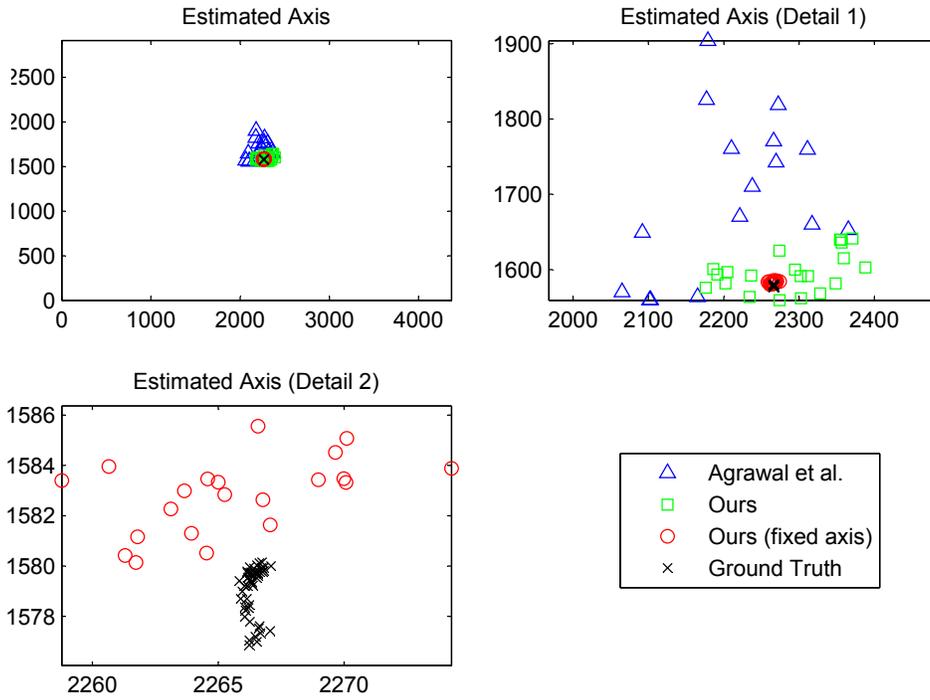


Figure 4.8: Refraction axis estimation results. Each marker represents the projection point of the axis. The vertical and horizontal axes are in pixels and are shown with the same aspect ratio as the image. (Top Left) The entire image. (Top Right) Detail near the center of the image. (Bottom Left) Detail showing the distribution of ground truth estimates and the estimates from our method.

Method	Estimated d_0 (mm)	σ	Error in \mathbf{A} (degrees)	σ
Ground Truth	45.91	0.07 [†]	0.011 [‡]	0.006
Agrawal <i>et al.</i>	43.34	39.22	2.173	0.775
Ours	46.09	11.77	0.866	0.393
Ours (fixed axis)	52.31	14.08	0.065	0.022

Table 4.3: Estimated calibration parameters using real data. [†]Standard deviation of linear scale fitting error (Subsection 4.3.2). [‡]Error of each measurement point with respect to the average.

variations of our method, one of which is labeled “fixed axis.” This label means that the estimated refraction axis \mathbf{A} from the first calibration step, described in Subsection 4.1.1, was taken as the final value and not allowed to vary during the nonlinear refinement step (Subsection 4.1.4). The reason we did this is seen in Figure 4.8 (Bottom Left), where the refraction axis estimates have very low spread and are within a few pixels of the ground truth. This level of accuracy is unfortunately degraded when nonlinear refinement is allowed to vary \mathbf{A} , but on the other hand the estimate for d_0 improves, as shown in Table 4.3.

Interestingly, we found that the estimated d_0 was particularly sensitive to variations in the refractive index difference $\lambda_{2,405\text{nm}} - \lambda_{2,660\text{nm}}$ for water, but much less sensitive to variations in both index values that do not change this difference. This is in line with the theory since the triangulation constraint is based on the difference in refraction angle. In our experiments, we found that varying only one index value by 0.0005 affected the estimated d_0 by about 4mm, whereas varying both values correspondingly by this amount had an effect of only 0.3mm. Further investigation is needed to generalize these relationships, but for our purposes the refractive index of water is known to a high enough precision that we can achieve good results.

Overall we see that our calibration method gives better accuracy than Agrawal *et al.*'s method, which agrees with the results of our synthetic data experiments. The “fixed axis” variant highlights a discrepancy in that the nonlinear refinement step did not improve all parameter estimates as expected, but actually increased the error in the refraction axis. While we do not have a definite explanation for this behavior, one possibility is suggested and briefly analyzed in Subsection 5.3.3, when we apply our calibration procedure to 3D reconstruction with real images.

Chapter 5

Underwater 3D Reconstruction

In this chapter we explore the problem of 3D reconstruction in the presence of refraction. Our goal is to achieve high-quality reconstruction results in underwater environments. To that end, we first describe the necessary adaptations that we made to existing software. Then, through a set of simulated experiments with a multi-camera array, we aim to answer the following questions:

1. Under what conditions, and in what ways, is there a significant difference between the reconstruction results obtained using a physically-correct refraction model compared to a perspective approximation?
2. What is the impact of calibration accuracy on the quality of the reconstruction?

After establishing the feasibility and identifying some advantages of using a physically-correct refraction model, we conduct an experiment using real data to validate the proposed methods. Finally, we explore a potential application using images of the seabed captured by an underwater observation platform.

5.1 Reconstruction Method

As discussed in Chapter 1, the software that we have adapted for underwater 3D reconstruction are PMVS by Furukawa and Ponce [18], and Bundler by Snavely *et al.* [48]. These two open-source implementations come together to form a workflow for a dense, patch-based 3D reconstruction.

5.1.1 Reference Implementation

Kang *et al.* showed in an experimental study that the combination of PMVS and Bundler can produce satisfactory results for underwater reconstruction, given certain mild assumptions [30]. To follow their terminology, we will refer to the direct application of these algorithms as “Rdist+Fadj.” The name refers to the fact that Bundler optimizes both the focal length and radial distortion parameters of each camera, thereby approximating refraction through the camera intrinsic parameters. (Note that the scene structure is optimized simultaneously.)

For our experiments, we use an initial focal length approximation as described in section 3.3. The radial distortion model used by Bundler has two parameters as shown in (3.16). In addition to the optimized focal length and radial distortion parameters, Bundler also outputs a rotation and translation for each camera. These parameters are passed into PMVS, along with the undistorted images, to compute the final dense reconstruction.

5.1.2 Refraction-Corrected Implementation

We made a number of modifications to both Bundler and PMVS to account for a physically-correct model of refraction, as described in Section 3.3. The most

substantial changes are outlined here, and the code is made available at www.cs.ualberta.ca/~yang/vision.htm. We will refer to the 3D reconstruction method using this refraction model as “FlatRefract” (in reference to the planar nature of the refraction interfaces).

Bundler modifications

We modified the main bundle adjustment algorithm to add the refraction parameters n and d_i . This involves replacing the perspective projection function with a new refractive forward projection function (Subsection 4.1.4) to map scene points to image points. The initial triangulation of scene points also needed to be changed. For convenience in this particular codebase, we used a “virtual focal length” method very similar to the one described by Jordt-Sedlazeck and Koch [28]. It is essentially a pixel-wise focal length adjustment where each image point is back projected into a scene ray, and the ray is then intersected with the refraction axis to find the equivalent perspective projection. This process is illustrated in Figure 3.5 for a single refraction.

PMVS modifications

Modifications were made to PMVS in the following areas:

1. Initial feature point matching,
2. Point triangulation,
3. Patch projection and sampling, and
4. Patch neighbor radius.

The first step in the PMVS algorithm is to detect and match feature points across images. Since the camera pose and intrinsic parameters are already known, the search for correspondences is restricted to a small margin around each epipolar line (see Section 2.1). As shown by Gedge *et al.*, this is incorrect in the presence of refraction and can lead to erroneous or missed correspondences [20]. We follow their approach in sampling along the refractive epipolar curves, with two changes: firstly, the sampling density is automatically adjusted based on the image distance between sample points. Secondly, the depth limits of the sampling volume are computed from the minimum allowed angle between triangulation rays (this is the parameter that ensures a reasonable baseline to point depth ratio). Thus the need for manually specified parameters is removed.

After each pair of corresponding feature points is obtained, the scene point is obtained using triangulation to create a patch. We implemented the back projection method described in Subsection 3.3.1 together with the midpoint triangulation method from Section 2.1. This is followed by a nonlinear optimization to minimize the reprojection error of the triangulated point, using refractive forward projection.

The main optimization carried out by PMVS is to minimize the photometric discrepancy of each patch across multiple views. We chose to implement a rigorous sampling strategy where every point on a patch is forward projected, accounting for refraction, onto each view where the patch is visible. (In the original code, only the center point and two orthogonal directions are projected, and the remaining points are sampled along these directions in image space; see Figure 2.5.) While a straightforward implementation would be rather slow, our implementation takes advantage of the forward projection technique from Section 3.3,

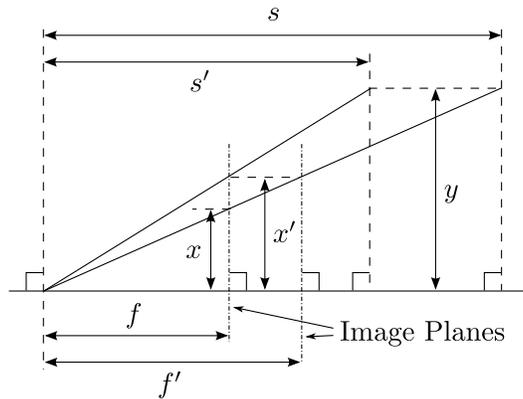


Figure 5.1: Relationship between focal length and object distance.

in which the projection of one point can be re-used as the initial guess for neighboring points.

Lastly, a small but significant change concerns the computation of the patch neighbor radius. This radius controls the distance between neighboring patches, and is directly proportional to the distance from the camera. In the presence of refraction, objects in the scene can appear closer or further away depending on the refractive indices. As we noted in Section 3.3, such a magnification effect can be approximated by adjusting the camera focal length. For a pinhole perspective camera without refraction, this is equivalent to changing the distance from the camera to the object.

Suppose the object, which has a displacement y away from the optical axis, is located at distance s from the camera and is imaged at point x . If the focal length is scaled by a certain factor from f to f' such that the image now appears at x' , then the same effect can be achieved with a new distance s' , related to s by the

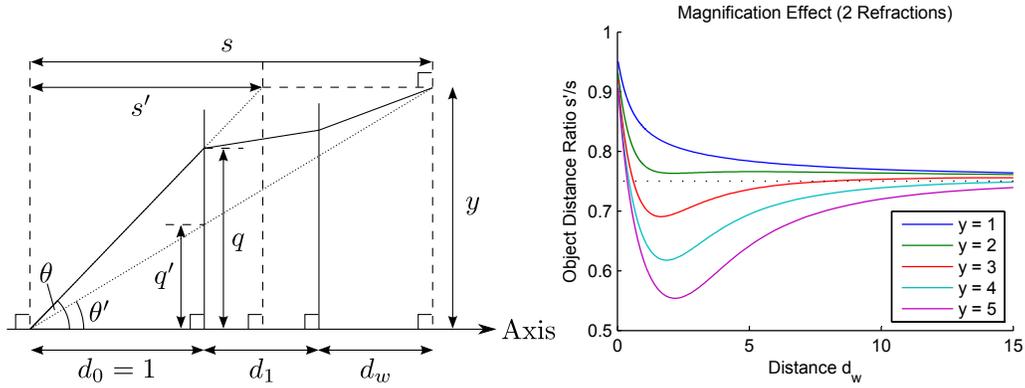


Figure 5.2: (Left) Geometry of light paths with and without refraction. (Right) Apparent shift in object position for an air-glass-water refraction model with $d_1 = 0.1$. The dotted line is the refractive index ratio of air to water. The corresponding graph for a single refraction is very similar.

same scale factor. With reference to Figure 5.1,

$$\frac{fy}{s'} = x' = \frac{f'y}{s} \Rightarrow \frac{s'}{s} = \frac{f}{f'}. \quad (5.1)$$

Therefore, we could scale the patch neighbor radius by a constant factor, based on the focal length adjustment strategy.

Instead, we choose to follow a more rigorous approach and compute the true scale factor for each reconstructed patch. For the underwater reconstruction problem that we are concerned with, the objects always appear closer. This means that the patch neighborhood radius will be smaller when performing a 3D reconstruction without refraction correction. In order to achieve the same patch density with a refraction-corrected reconstruction, it is necessary to adjust the radius according to the distance along the observed ray to the patch's apparent position.

Figure 5.2 shows a typical air-glass-water refraction model and compares the apparent object distance s' to the actual object distance s . Given that θ and θ' are

respectively the angles of the rays with and without refraction, q and q' are the corresponding projected points on the first interface from the camera, and y is the object's displacement from the refraction axis, we can write:

$$s \tan(\theta') = y = s' \tan(\theta) \Rightarrow \frac{s'}{s} = \frac{\tan(\theta')}{\tan(\theta)} = \frac{q'}{q} \quad (5.2)$$

The right side of Figure 5.2 illustrates how the ratio varies with y and the object distance in water d_w .

In the limit as $d_w \rightarrow \infty$, we observed that $s'/s \rightarrow \mu_0/\mu_2$, where μ_0 and μ_2 are respectively the refractive indices of air and water. This is the same ratio that we derived in Subsection 3.3.3, suggesting that for distant objects the focal length scaling may be a reasonable approximation. However, we did not further investigate the limit convergence properties since the relationship between q and q' involves a quartic equation without a convenient closed-form solution.

5.2 Camera Array Calibration

5.2.1 Extrinsic Calibration

The calibration method presented in Chapter 4 only deals with obtaining the refraction and camera parameters of a single view. For 3D reconstruction from multiple views, we also need the relative pose of each view, as mentioned in Subsection 2.1.3. We obtain these parameters by capturing images of a standard checkerboard pattern, extracting the grid points on the pattern, and then running Bundler using these points as input. Note that the camera intrinsic parameters are applied first (see 5.2.2), so that the bundle adjustment manipulates only the

rotation and translation parameters of each view.

We use a checkerboard pattern instead of a general scene because the grid intersection points can be found accurately¹ and matched perfectly across views. Furthermore, points from multiple placements of the checkerboard can be combined into a single, dense point cloud. After the camera extrinsic parameters are found, we apply a scaling factor so that the sparse reconstruction output of Bundler matches the physically measured dimensions of the checkerboard. This allows us to perform 3D reconstructions at the actual scale and to take measurements in real world units.

5.2.2 Intrinsic Calibration

By default, Bundler assumes that the principal point (intersection of the optical axis with the image plane) is the center of the image, and includes the focal length as an optimization parameter. While these assumptions are reasonable for uncalibrated images, in our work we can easily obtain accurate intrinsic parameters using the MATLAB Camera Calibration Toolbox [4]. We therefore implemented an image pre-processing step to correct for radial distortion and shift the principal point to the image center.

In effect, we compensate for lens distortions and misalignment separately, so that the Rdist+Fadj method only has to deal with distortion due to refraction. This pre-processing is also helpful for the FlatRefract method: we can assume a known focal length with no lens distortion during the bundle adjustment phase, reducing the complexity and increasing the stability of the optimization.

¹We use OpenCV's findChessboardCorners function [6].

5.3 Experimental Results

5.3.1 Evaluation Methodology

Our evaluation methodology is similar to the one described by Seitz *et al.* and used in the Middlebury Multi-View Stereo benchmark [46]. Specifically, we measured the *accuracy* and *completeness* of the reconstructed model R relative to the ground truth G . We obtain R simply by considering the center point of each patch reconstructed by PMVS. For synthetic data we sample the ground truth geometry very densely (400 points/cm²) to create G , whereas for real data it is obtained from a multi-view 3D reconstruction in air (see Section 5.3.3).

The first step is to align the two point clouds using an iterative closest point (ICP) algorithm². We manually inspect the results to ensure that proper alignment is achieved. Each point $\mathbf{x} \in G$ is a correspondence of a point $\mathbf{y} \in R$ if $\|\mathbf{x} - \mathbf{y}\| < \rho_C$, where $\|\cdot\|$ denotes Euclidean distance and ρ_C is a threshold chosen to be 1mm in our experiments. The completeness of R is then defined as the proportion of points in G having correspondences in R :

$$\text{Completeness}(R, G) = \frac{|\{\mathbf{x} \in G : \|\mathbf{x} - \mathbf{y}\| < \rho_C, \mathbf{y} \in R\}|}{|G|} \quad (5.3)$$

On the other hand, the accuracy of the reconstructed model is measured in two different ways. For synthetic data we have the benefit of the ground truth geometry, and therefore we compute the perpendicular distance of each point in R to the nearest surface. For real data we measure the smallest distance from each point in R to any of its correspondences in G . The distance measured for each point rep-

²We used the open-source CloudCompare software [16].

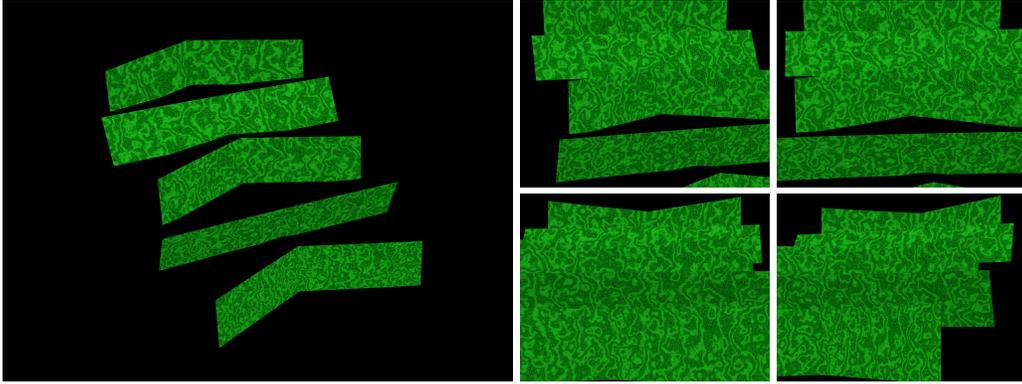


Figure 5.3: Renderings of the scene used in our synthetic data experiments. (Left) An overhead view, with the cameras are positioned near the bottom right of the image and point toward the top left. The green planes recede away from the cameras and rise upwards, and the furthest planes have enlarged textures to facilitate feature matching. (Right) Four of the actual camera views.

resents its “error” with respect to the ground truth model. One way to compute an accuracy score for the reconstructed model is to simply average the point errors:

$$\text{Accuracy}(R, G) = \frac{1}{|R'|} \sum_{\mathbf{y} \in R'} \min_{\{\mathbf{x} \in G: \|\mathbf{x} - \mathbf{y}\| < \rho_A\}} \|\mathbf{x} - \mathbf{y}\|, \quad (5.4)$$

where $R' = \{\mathbf{y} \in R : \|\mathbf{x} - \mathbf{y}\| < \rho_A, \mathbf{x} \in G\}$ and $\rho_A = 2\text{mm}$. For sections with more detailed analysis, however, we characterize the accuracy through error histograms, allowing us to estimate the number of points within each range of error values.

5.3.2 Synthetic Data Results

In this subsection we present 3D reconstruction results from synthetically generated images. Our objective is to study the impact of using a physically-correct refraction model, while minimizing the influence of unrelated sources of error

such as lens imperfections, calibration errors, measurement noise, and so on.

We designed a test scene shown in Figure 5.3 consisting of several textured planes, slanted at different angles and placed at varying depths. This scene covers most of the scene volume captured by our camera array configuration (see below), ensures a unique solution for point cloud alignment, allows measurement of anisotropic scaling or shearing in any direction, and facilitates unambiguous measurement of the distance from each reconstructed point to the ground truth surface.

We simulated a two-layer flat refraction model based on the setup in our real data experiments (Subsection 5.3.3), with a 5.6mm layer of acrylic in front of each camera followed by the underwater scene. The refractive indices that we used are 1.491 for acrylic and 1.333 for water. The cameras are arranged in a grid pattern of two rows and four columns, with a vertical/horizontal spacing of 0.2/0.15m. The scene itself has a depth range of about 0.4m and the center is located about 0.6m away from the camera array. All views were rendered using the POV-Ray ray tracing program [40], simulating cameras with a resolution of 1032×776 pixels and a focal length of 1805 pixels (these parameters are derived from our real data experiments).

Using the camera and scene setup described above, we have nine different test case combinations of the following variables:

1. Number of views: all eight views, middle two views from the top row only, or four views by taking the middle two of each row.
2. Refraction interfaces setup: a single set of interfaces shared by all cameras (as if looking through the same side of a water tank), or a separate set of

interfaces for each camera. In the latter case, the axis of refraction is either kept parallel to the optical axis of each camera, or tilted by 3.6 degrees.

Note that in the shared refraction interfaces setup, the cameras are angled inwards to center the scene in each image. This introduces a refraction axis tilt between 8 and 25 degrees depending on the camera position.

Comparison of Imaging Models

We first compare the 3D reconstruction results of the FlatRefract and Rdist+Fadj methods. Since we are only interested in the impact of the imaging model, we leave out the effects of calibration by initializing the reconstruction with the ground truth camera and refraction parameters. The Rdist+Fadj method was initialized with the ground truth camera poses and an adjusted focal length (Section 3.3). We also used perfectly matched, noise-free feature points as input to the bundle adjustment phase. These feature points were generated from a sparse, regular sampling of the ground truth geometry, giving about 700 points in each view.

For both FlatRefract and Rdist+Fadj, we investigated three different configurations of the bundle adjustment procedure:

1. All camera parameters (including any refraction parameters) are adjusted. This is denoted as “full adj.”
2. All parameters except camera translation are adjusted. This is denoted as “fixed t.”
3. All parameters except camera translation and rotation are adjusted. This is denoted as “fixed R,t.”

Additionally, we included results without using bundle adjustment, in which the FlatRefract method uses ground truth camera and refraction parameters. (There is no equivalent for Rdist+Fadj, since this imaging model is an approximation.)

Accuracy. Figures 5.4, 5.5, and 5.6 display our experimental results for reconstruction accuracy. For each imaging model and bundle adjustment configuration, we plot a histogram of reconstructed point errors as a line graph, with 50 histogram bins over the range from 0mm to 2mm. In separate bar graphs we also visualize the scale factor adjustment that was applied during the point cloud alignment process (Subsection 5.3.1). A scale factor of 1 means that the reconstructed model is at the correct scale, and anything else means that it is either larger or smaller than the ground truth model. Some examples of the 3D reconstructions are shown in Figures 5.7, 5.8, and 5.9, together with the ground truth geometry for a visual comparison.

From this data we can make a number of interesting observations. First of all, the FlatRefract method maintains high accuracy in all test cases, with the error distributions heavily biased toward 0 and dropping off rapidly by 0.5mm. The scale factor errors are also very low, being at most 0.03%. (Note that the results are not perfect because the PMVS reconstruction phase involves detecting and matching new feature points, and optimizing patches by projecting onto image pixels.) As one might expect, accuracy improves when adding more views, and the differences between the bundle adjustment configurations are negligible since noise-free feature points were used.

Secondly, the performance of the Rdist+Fadj reconstruction method varies widely depending on the setup of the refractive interfaces. When the axes of re-

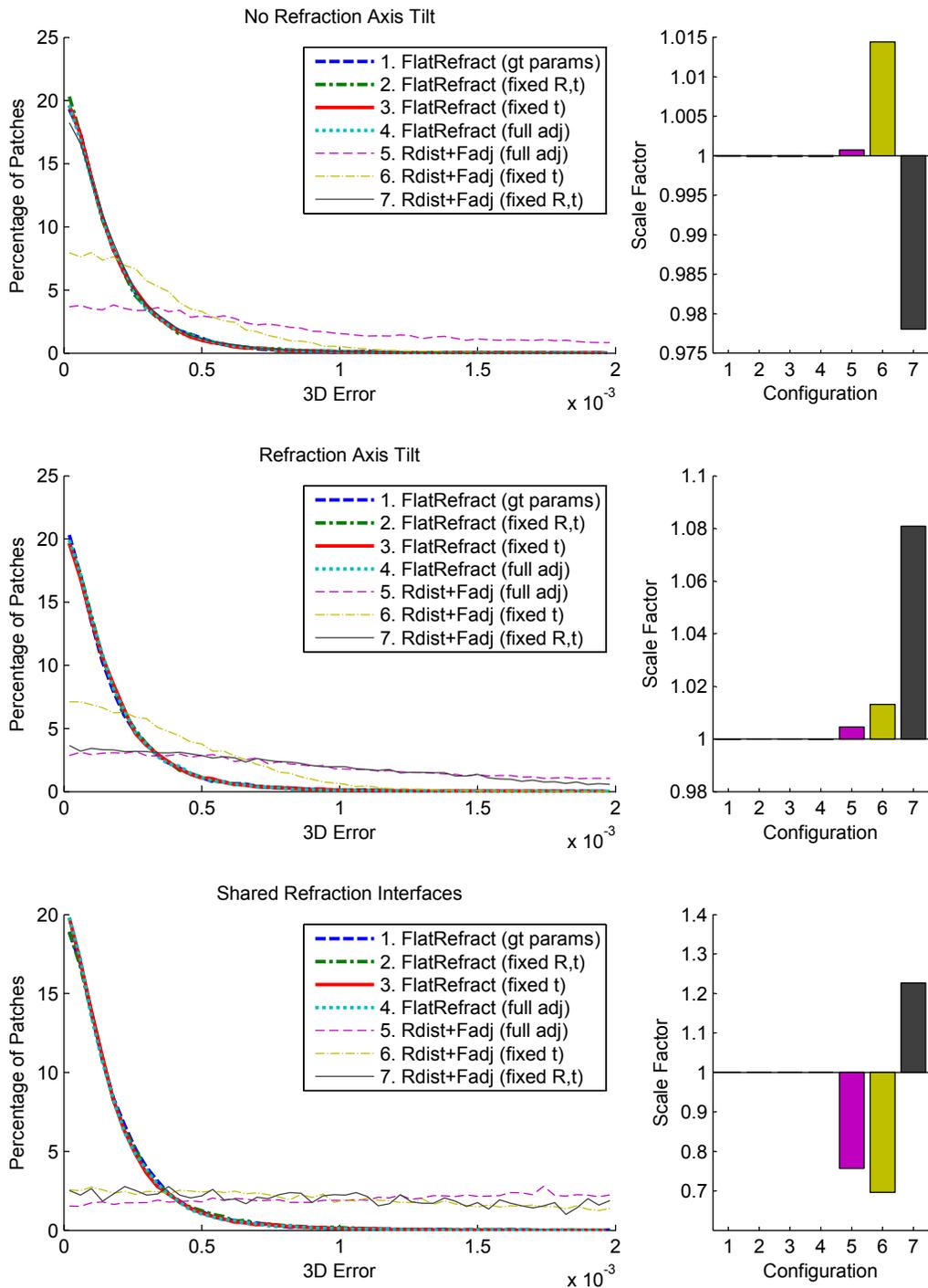


Figure 5.4: Error distribution and scale factor plots for two views.

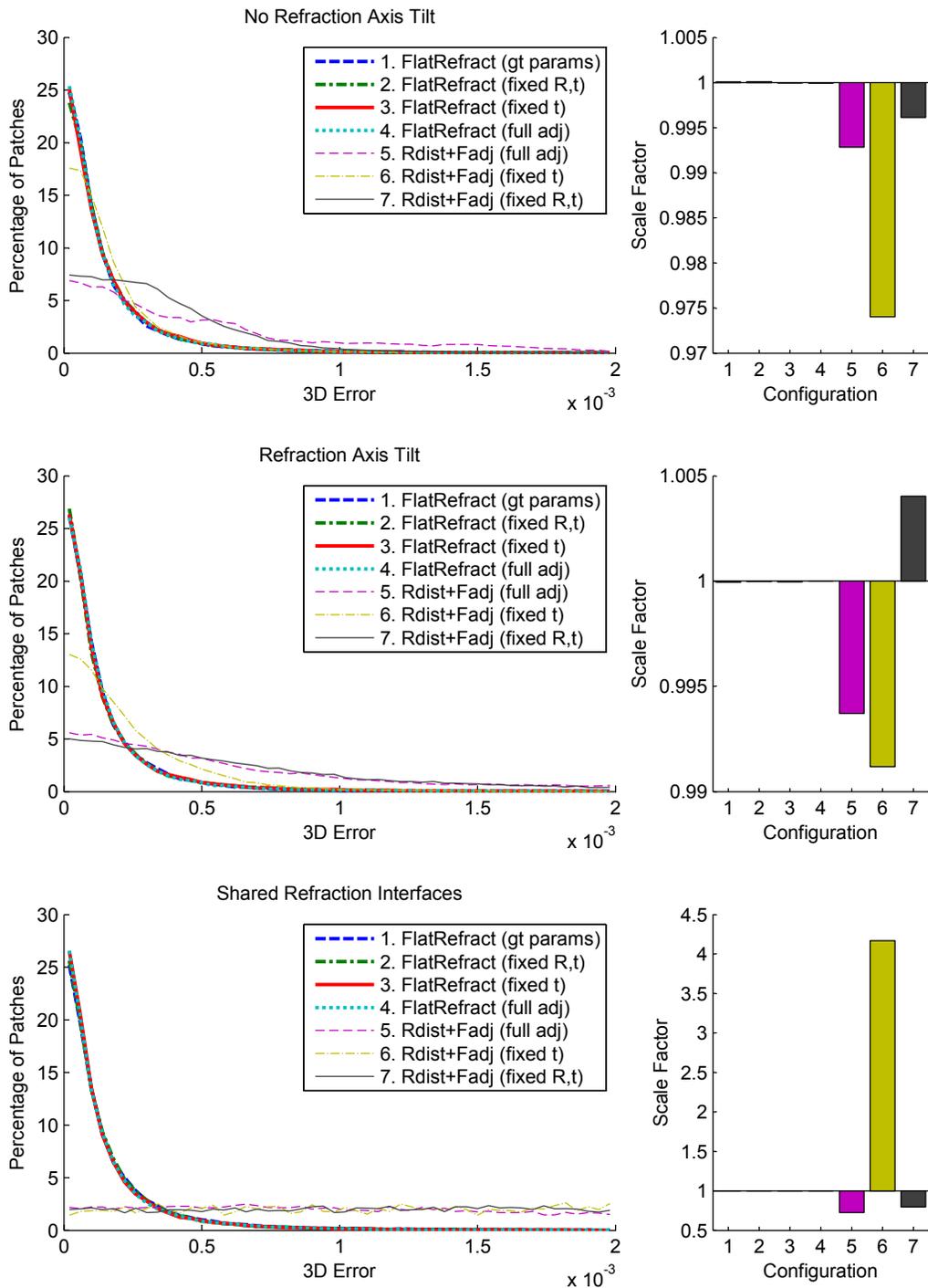


Figure 5.5: Error distribution and scale factor plots for four views.

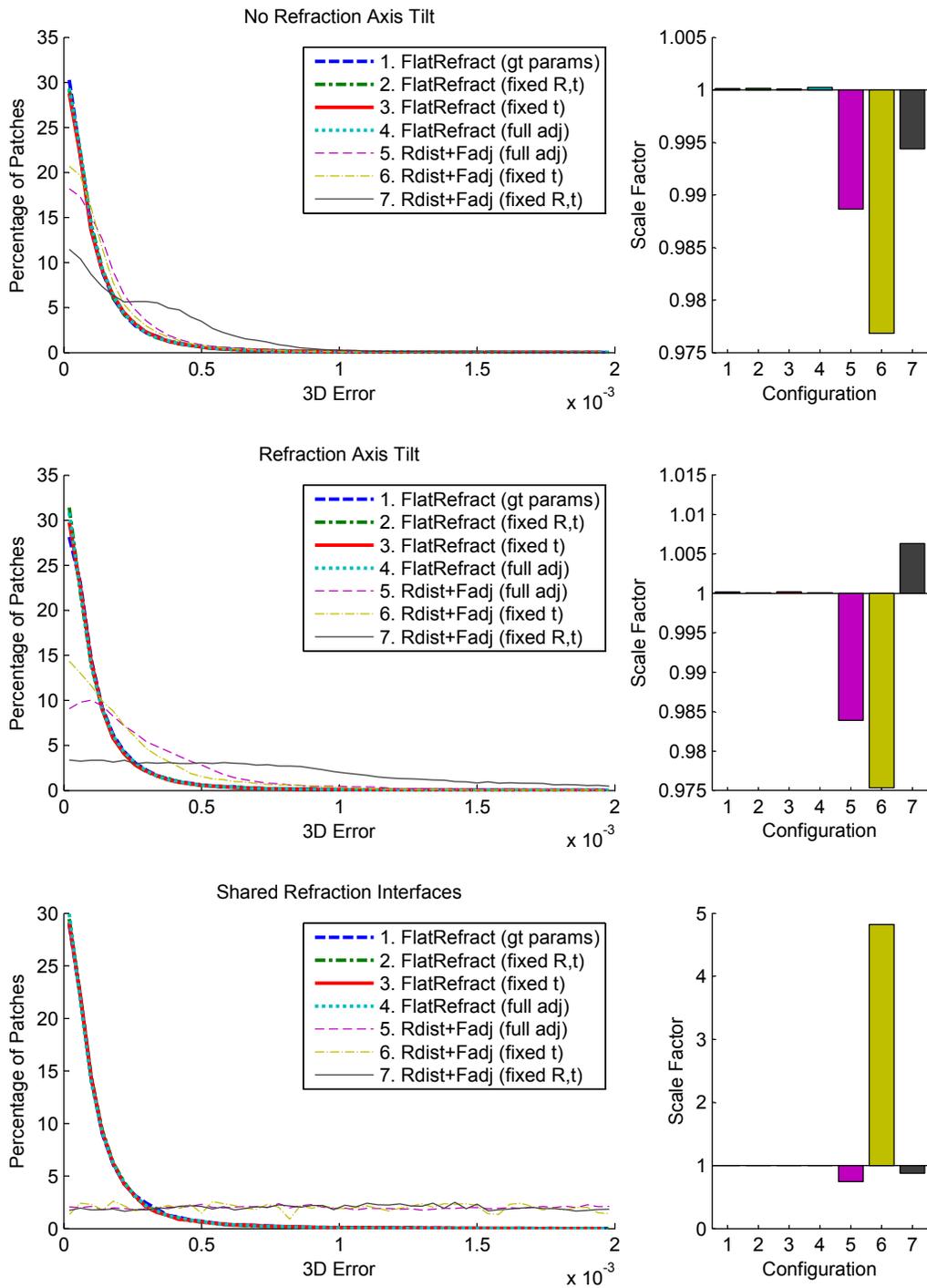


Figure 5.6: Error distribution and scale factor plots for eight views.

fraction are not tilted with respect to the optical axes, Rdist+Fadj can sometimes yield results on par with FlatRefract, as shown in Figure 5.4 (Top, “fixed R,t”) and visualized in 5.7 (Bottom Right). This particular test case and configuration seems to be an exception, however, as all the other test cases show lower accuracy for Rdist+Fadj. Furthermore, when a modest tilt is introduced in the refraction axes (see above), the accuracy decreases both in absolute terms and relative to the refraction-corrected method. This yields some visible discrepancies from the ground truth data as shown in Figure 5.8. When using a setup with shared refraction interfaces, both the error histograms and the visualizations in Figure 5.9 demonstrate that Rdist+Fadj consistently fails to produce a reasonable reconstruction result, regardless of the number of views.

Lastly, the 3D reconstructions using Rdist+Fadj show significant and varying changes to the overall scale of the scene. Previous studies have ignored scene scaling by focusing on metric reconstruction, where scale is ambiguous [30, 31], but we believe that in many applications it is important to obtain 3D data in real world units. From this perspective, Rdist+Fadj has the undesirable effect of scaling the scene by 2-2.5% in several cases (and much worse when using shared interfaces). Moreover, the amount of scaling is inconsistent as the number of views is varied, as we see a markedly different pattern for two views compared to four or eight views. Since the two-view reconstructions largely omit the closest set of angled planes (see Fig. 5.7, Bottom) because no views are taken from the lower row, this suggests that the scale factor also depends on which part of the scene is visible. This is in contrast to the FlatRefract method, where the scale factor is consistently very close to 1.

If we compare the three bundle adjustment configurations for Rdist+Fadj, we

see that none of them is clearly superior in all test cases. The “fixed t ” configuration gives better accuracy in most cases, but also tends to have the highest scale factor error. On the other hand, “full adj” generally gives a scale factor close to 1, but has the worst accuracy for two and four views. It appears that there may be some trade-off between point cloud accuracy and scale factor error.

Completeness. Applying the completeness measure (5.3) to the 3D reconstruction results yields Figure 5.10. Observe that the completeness scores in the separate interfaces test cases closely mirror the accuracy plots discussed above. This means that in these test cases, there are no major holes or gaps in any of the reconstruction results (corroborated by manual inspection), and incompleteness is mainly due to loss of accuracy.

Overall, we see once again that the FlatRefract method achieves the highest completeness score, except where it is matched by Rdist+Fadj (fixed R,t) for two views, and Rdist+Fadj gives consistently poor results for the shared refraction interfaces setup. Note that the completeness scores do not reach 1.0 because some parts of the scene are occluded or fall outside of the cameras’ field of view.

Impact of Feature Detection Errors

In the previous subsection, error-free feature points were used as input to the 3D reconstruction process. We now assess the impact of errors in feature points by using Bundler’s built-in feature detection and matching method, which is based on SIFT [36]. Other than the input feature points, the experiment was structured with exactly the same test cases and bundle adjustment configurations. The results for eight views are shown in Figures 5.11 and 5.12. Further results for two and

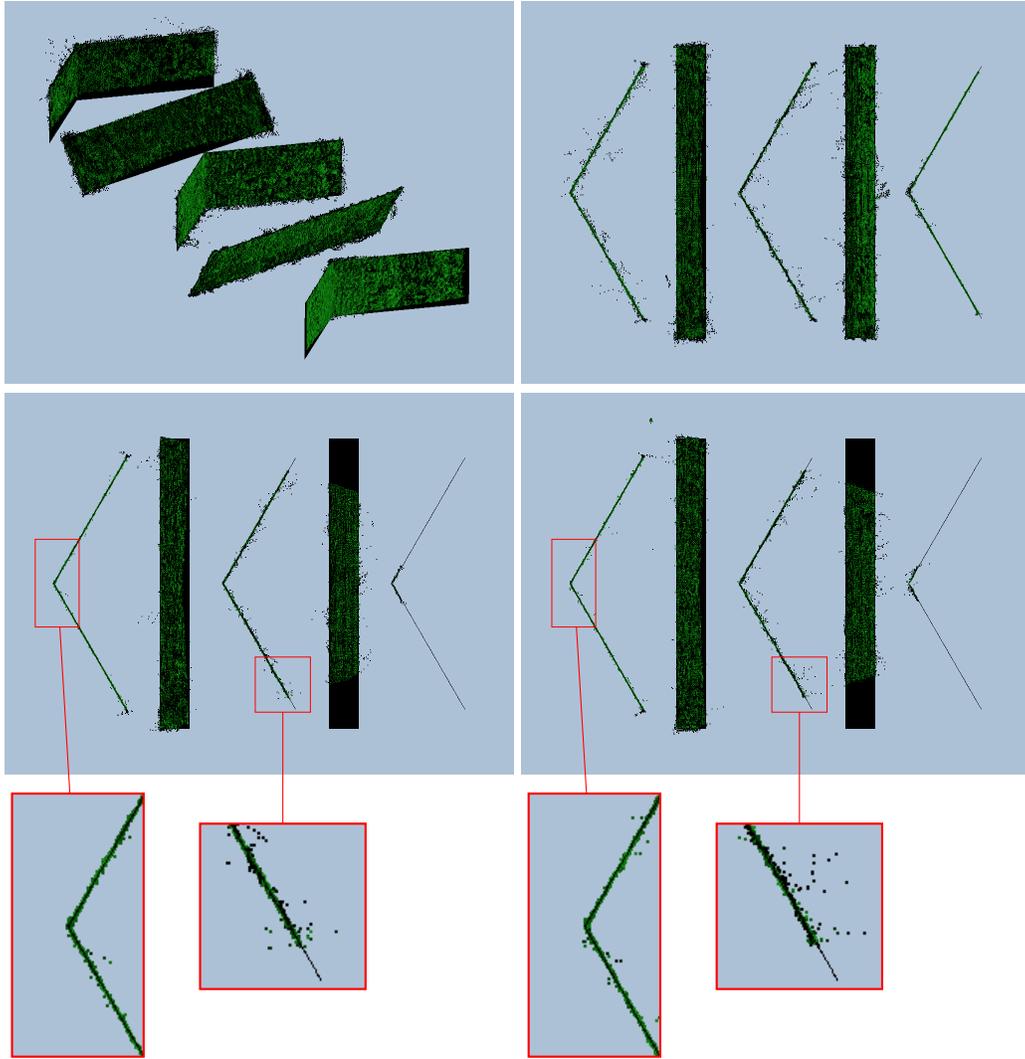


Figure 5.7: Reconstructed 3D point clouds (green) overlaid on the ground truth geometry (black). (Top Row) Reconstruction result for FlatRefract (full bundle adjustment), eight views, and non-tilted refraction axes. The same scene is shown from two different angles. (Bottom Left) Result for FlatRefract (fixed R,t), two views, and no axis tilt. (Bottom Right) Result for Rdist+Fadj (fixed R,t), two views, and no axis tilt. Magnified views are provided for visual comparison with Figure 5.8.

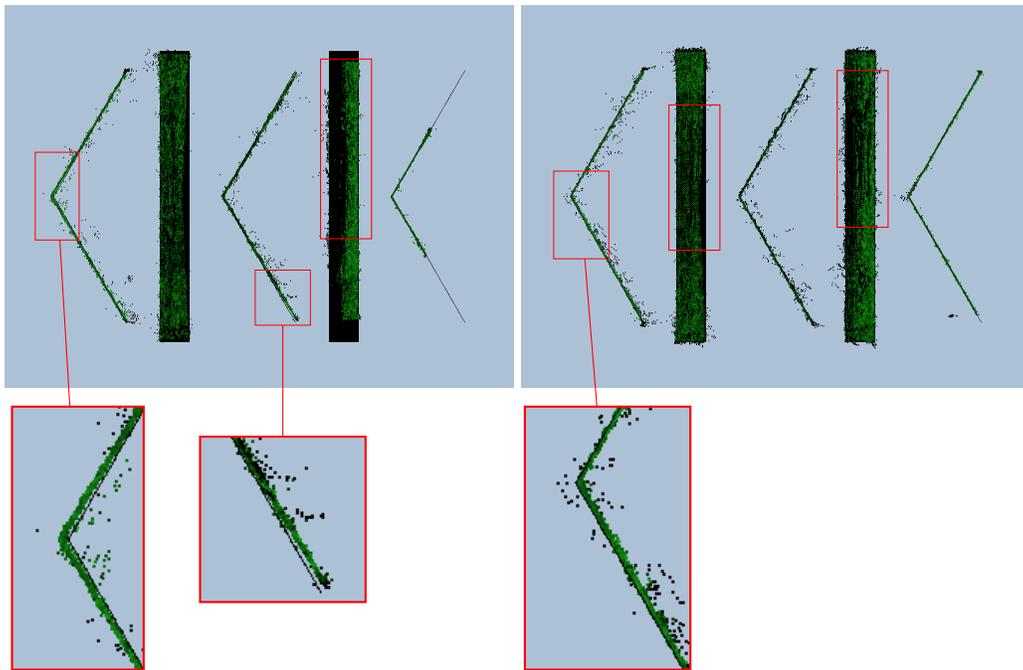


Figure 5.8: Reconstructed 3D point clouds, with highlighted regions showing deviations from the ground truth geometry. (Left) Result for Rdist+Fadj (full adj), four views, and tilted refraction axes. Gaps between the reconstructed points and the ground truth model are visible in the magnified regions, and the remaining highlighted region shows where the points intersect and disappear behind the ground truth plane. (Right) Result for Rdist+Fadj (full adj), eight views, and tilted refraction axes. The magnified region shows where points are not centered on the ground truth plane, and in the other two regions the density of points varies, indicating that they form slightly curved surfaces intersecting the ground truth model.

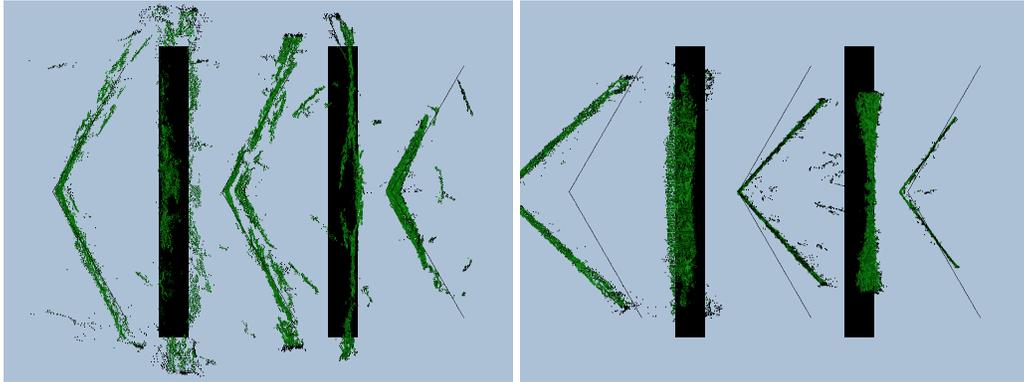


Figure 5.9: Reconstructed 3D point clouds, showing examples where Rdist+Fadj performs poorly. (Left) Result for Rdist+Fadj (fixed R,t), eight views, and shared refraction interfaces. (Right) Result for Rdist+Fadj (full adj), eight views, and shared refraction interfaces.

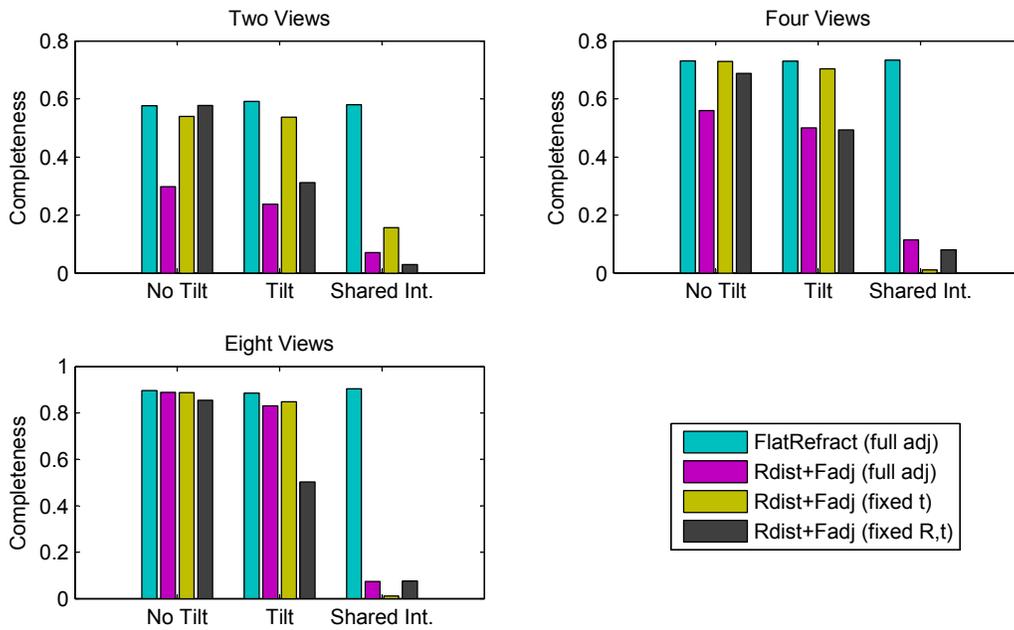


Figure 5.10: Completeness measure for 3D reconstruction results. To avoid clutter, only the “full adj” configuration is shown for the FlatRefract method; the results for the other configurations are very similar. “Tilt” refers to the tilt of the refraction axis with respect to the optical axis, and “Shared Int.” refers to the shared refraction interfaces setup.

four views are shown in Appendix A.

Compared with the previous accuracy results, we see that the Rdist+Fadj results are generally about the same, with a few exceptions: the “fixed t” configuration improves significantly for two views for the separate interfaces test cases, and yet is considerably poorer for eight views with no refraction axis tilt; also, the “full adj” configuration improves slightly for four and eight views with tilted refraction axis. On the other hand, with the FlatRefract method there are only instances where the results are poorer, since all of the bundle adjustment configurations already matched ground truth configuration previously. Of particular interest is that the “fixed R,t” configuration performed much worse for two views, but not for four or eight views; similarly, the “full adj” configuration suffers in four and eight views but not for two views. There is no configuration for the FlatRefract method that always performs worse when using feature points with errors, and neither is there any that always performs well (except for the ground truth configuration).

Given the evidence, it is difficult to draw definite conclusions about the impact of feature point errors, other than to note that they can sometimes cause measurably worse results for both the Rdist+Fadj and FlatRefract methods. There appears to be complex interactions between the parametrization of the bundle adjustment, the camera setup, and the scene itself (recall that the two-camera setup does not capture the front-most part of the scene), and a more detailed study is beyond the scope of this thesis. We are also not sure why Rdist+Fadj improves in some instances, but it may have to do with differences in where Bundler-detected feature points are concentrated, compared to the uniform sampling that we used. Nevertheless, we point out that the scale factor errors and completeness scores (Figure 5.12) are broadly very similar to the previous results, indicating that feature point

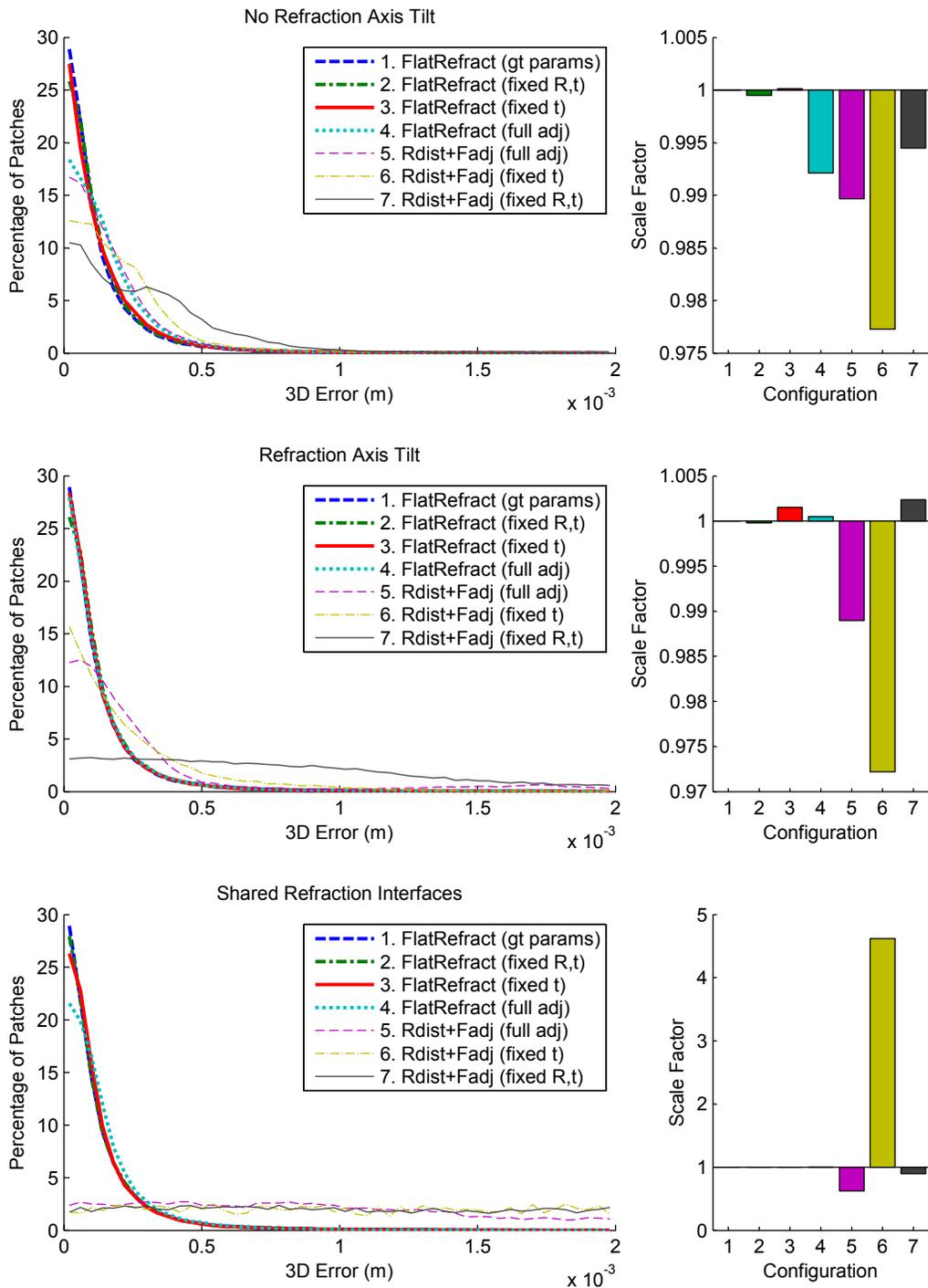


Figure 5.11: Error distribution and scale factor plots for eight views.

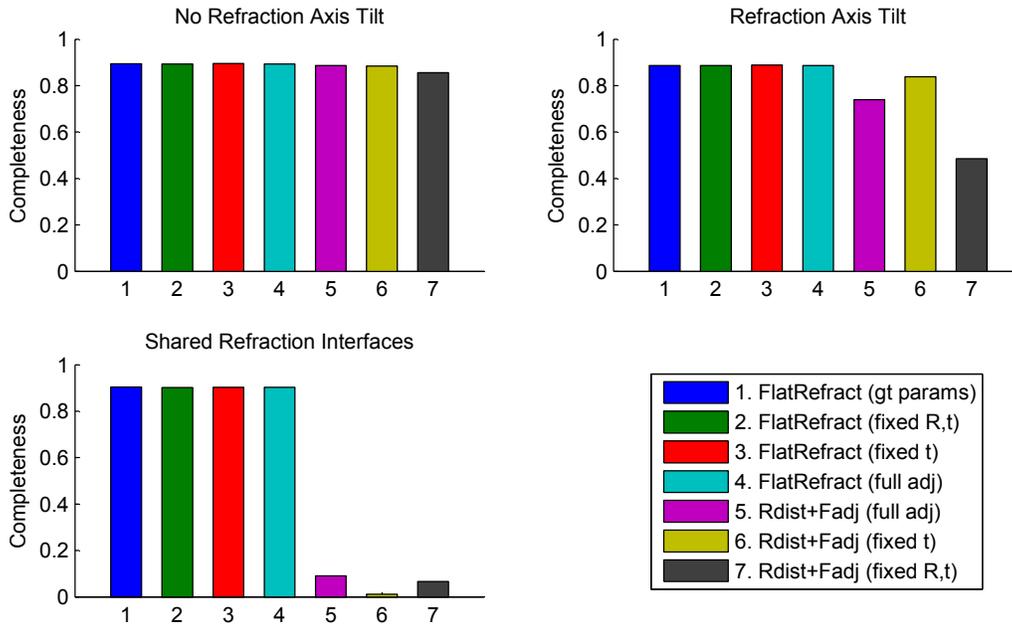


Figure 5.12: Completeness measure for eight view 3D reconstruction results using feature points detected by Bundler.

errors do not drastically alter the reconstructed 3D model characteristics.

Impact of Calibration Errors

Some amount of error in the calibrated camera parameters are inevitable in real applications. Specifically with respect to the refraction model parameters, we wish to understand the effect of such errors on 3D reconstruction and the extent to which they can be corrected during the bundle adjustment phase.

We repeated the synthetic data calibration experiment from Chapter 4 (Table 4.1 row II, 1px Gaussian noise), but using the lower resolution of the cameras in our array, and obtained a representative sample of errors in the camera-to-glass distance d_0 and angular errors in the refraction axis. These error values are multiplied by a factor ranging from 1.0 to 18.0 and used to corrupt the calibration parameters input to the 3D reconstruction stage. Table 5.1 shows a subset of the

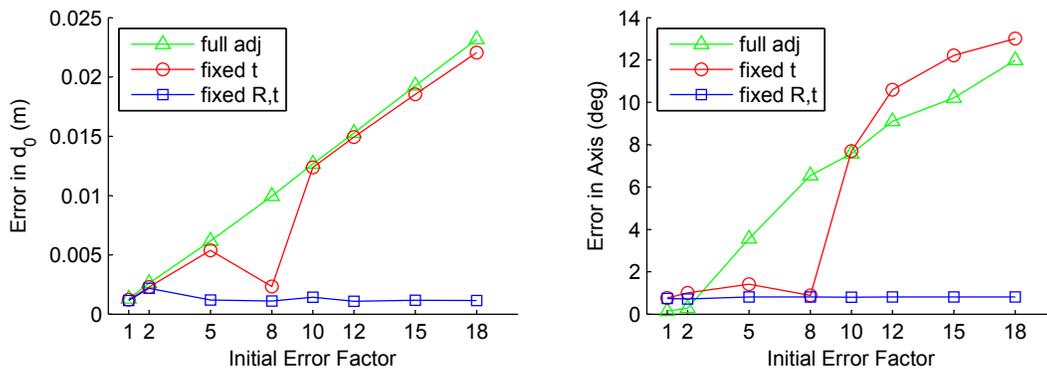


Figure 5.13: Bundle adjustment results for two views.

error values. Since the refraction model parameters do not apply to Rdist+Fadj, we look at only the FlatRefract method. We also focus on the shared refraction interfaces setup as it corresponds to the setup in our real data experiments.

Multiplication Factor	1	5	10	15
Mean Initial d_0 Error (m)	0.0029	0.0147	0.0294	0.0441
Mean Initial Axis Error (deg)	0.1914	0.9572	1.9144	2.8716

Table 5.1: Error values added to the calibration data. The base values in the first column are multiplied by constant factors, some of which are shown for reference.

Figures 5.13, 5.14, and 5.15 show the errors remaining in d_0 and the refraction axis, averaged over all cameras, after the bundle adjustment phase of the 3D reconstructions. The feature points used were detected from the images by Bundler. As the error in the initial calibration data is increased from 1 to 18 times the expected amount from our calibration method (see above), at some point the bundle adjustment may fail to recover the correct solution such that the residual errors increase markedly. For the “fixed t” configuration this appears to happen between 8-10 for two views, and around 12-15 for four and eight views. In terms of refraction axis error this is in the range of 2-3 degrees, which can potentially occur in practice as we saw in Table 4.3 for Agrawal *et al.*’s calibration method.

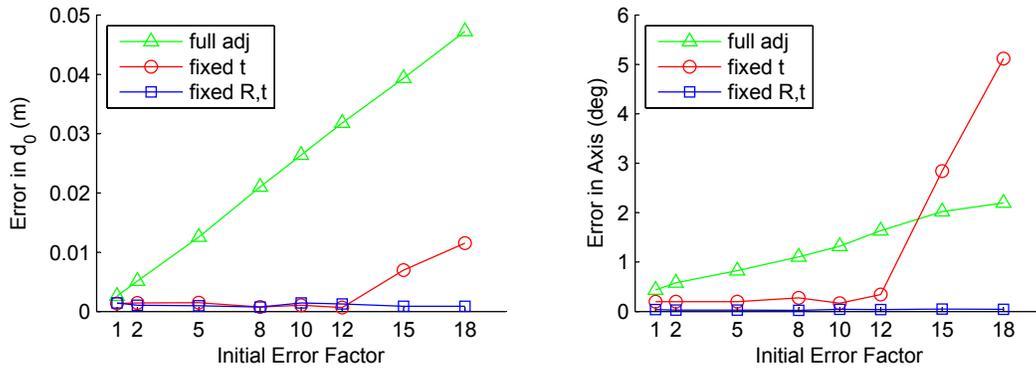


Figure 5.14: Bundle adjustment results for four views.

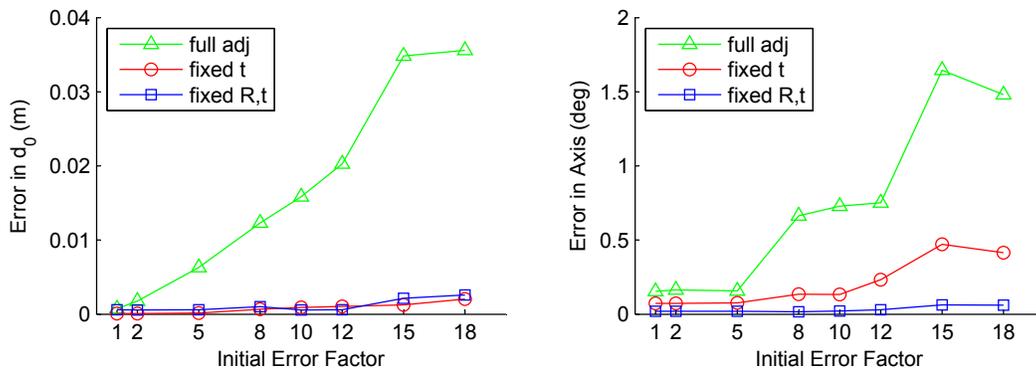


Figure 5.15: Bundle adjustment results for eight views.

Remarkably, the “fixed R,t” bundle adjustment configuration seemed to be quite insensitive to the starting conditions, in some cases performing better when the initial error was higher. On the other hand, the “full adj” configuration yielded consistently poor results. Even if we ignore the error in d_0 , which might be caused by a change in the overall scene scale (since camera positions are not fixed in this configuration), the large refraction axis errors suggest that straightforward bundle adjustment is insufficient to optimize camera extrinsic and refraction model parameters simultaneously. A similar method developed by Jordt-Sedlazeck and Koch seems to imply otherwise, but they do not provide a quantitative evaluation in either synthetic or real experiments to show that such an optimization was successfully achieved [28].

Comparing the bundle adjustment results with respect to the number of views, we see that lower errors are achieved with more views. In fact, for two views in the “fixed R,t” configuration the refraction axis error is about 0.8 degrees even when the initial error was lower. This improves in four and eight views where the refraction axis and d_0 errors remain below 0.063 degrees and 0.0026m respectively. We note, however, that this bundle adjustment configuration is given perfect data for the camera poses, which may not be a reasonable assumption in practice.

We analyze the 3D reconstruction results in Figures 5.16, 5.17, and 5.18. The plots for accuracy are similar in overall shape to the refraction axis error plots, affirming that the refraction model parameters have a direct influence on reconstruction quality. As for scale factor errors (the percentage difference between 1.0 and the reconstructed model scale), the “full adj” configuration predictably does not maintain the scale of the scene. In contrast, the errors for the other two configurations are generally below 0.3% (sometimes dipping to 0.02%), with the

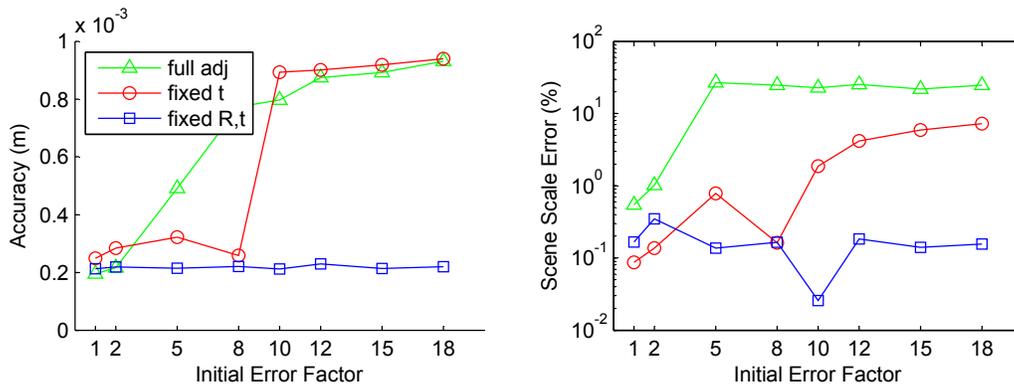


Figure 5.16: Reconstruction accuracy results for two views.

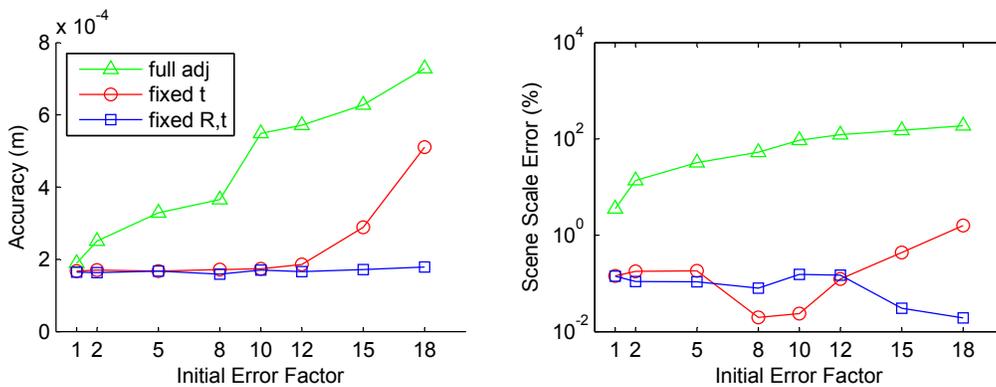


Figure 5.17: Reconstruction accuracy results for four views.

exception of “fixed t” with two views. Once again “fixed R,t” is seen to be the most robust bundle adjustment configuration. Plots for reconstruction completeness closely mirror the accuracy plots, and are included in Figure A.5.

5.3.3 Real Data Results

For our real data experiments, we set up a camera array and an underwater scene in a water tank as shown in Figure 5.19. Similar to the simulated camera array in our synthetic data experiments, the cameras are in two rows of four and spaced roughly 0.15m apart horizontally and 0.2m vertically. The array is placed parallel

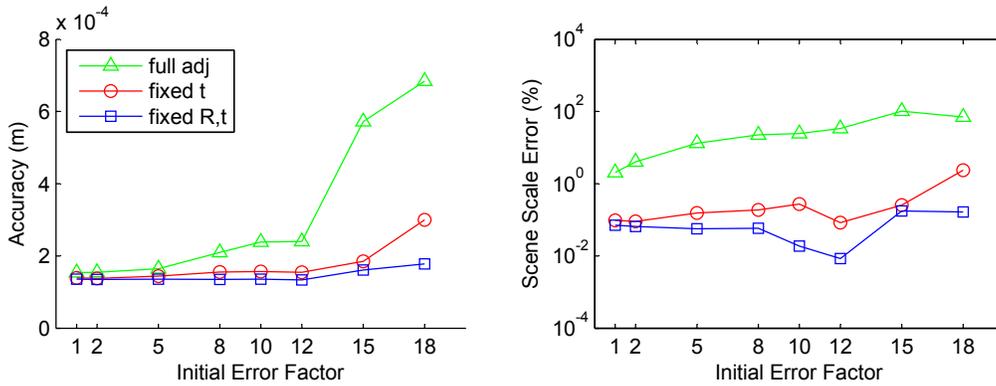


Figure 5.18: Reconstruction accuracy results for eight views.



Figure 5.19: Experimental setup for our real data experiments. (Left) The scene set up inside an acrylic water tank. (Right) The camera array.

to the flat surface of the tank so that the lens of each camera is within about 0.02m of the surface, and the cameras are angled inwards to capture the scene which is approximately 0.56m away. Each camera has a resolution of 1032×776 pixels and a horizontal field of view of about 32 degrees.

The scene itself consists of two aquarium objects attached together on a rigid base, positioned to provide a depth range of about 0.22m in each camera's field of view. To obtain a ground truth model, we removed the objects from the water and captured 100 images from different angles using a turntable, as shown in Figure 5.20. The model was generated by Bundler and PMVS using the standard



Figure 5.20: (Top Left) Camera setup to capture ground truth 3D model. (Top Right, Bottom) Various rendered views of the reconstructed ground truth model.

perspective projection model and known camera intrinsic parameters (see Subsection 5.2.2). Additionally, we placed a checkerboard pattern at the back of the model. The grid intersection points of the pattern were extracted separately and used to scale the model into real world units, similar to the process described in Subsection 5.2.1.

Overall Procedure

We first present a summary of the procedure we used in our real experiments, tying together all of the calibration and reconstruction steps discussed previously.

1. Calibrate the intrinsic parameters for each camera (Subsection 5.2.2).
2. Calibrate the extrinsic parameters for the camera array as a whole (Subsection 5.2.1).
3. Compute chromatic aberration correction image warp parameters for each camera (Subsection 4.2.3).
4. Calibrate the refraction model parameters for each camera (Chapter 4).
5. Capture images of the underwater scene.
6. Apply either the FlatRefract refraction-corrected reconstruction method (Subsection 5.1.1), or the Rdist+Fadj perspective approximation method (Subsection 5.1.2).

Figure 5.21 shows examples of the underwater images that we captured and used for our 3D reconstructions.

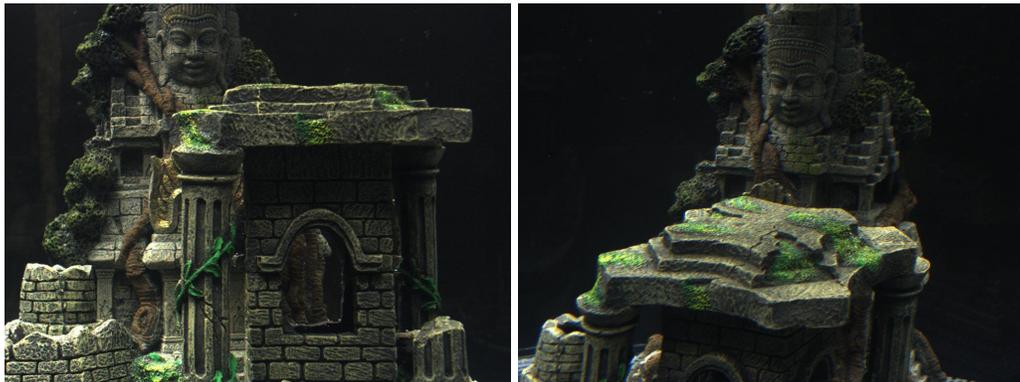


Figure 5.21: Examples of the images used in our real data 3D reconstruction experiments.

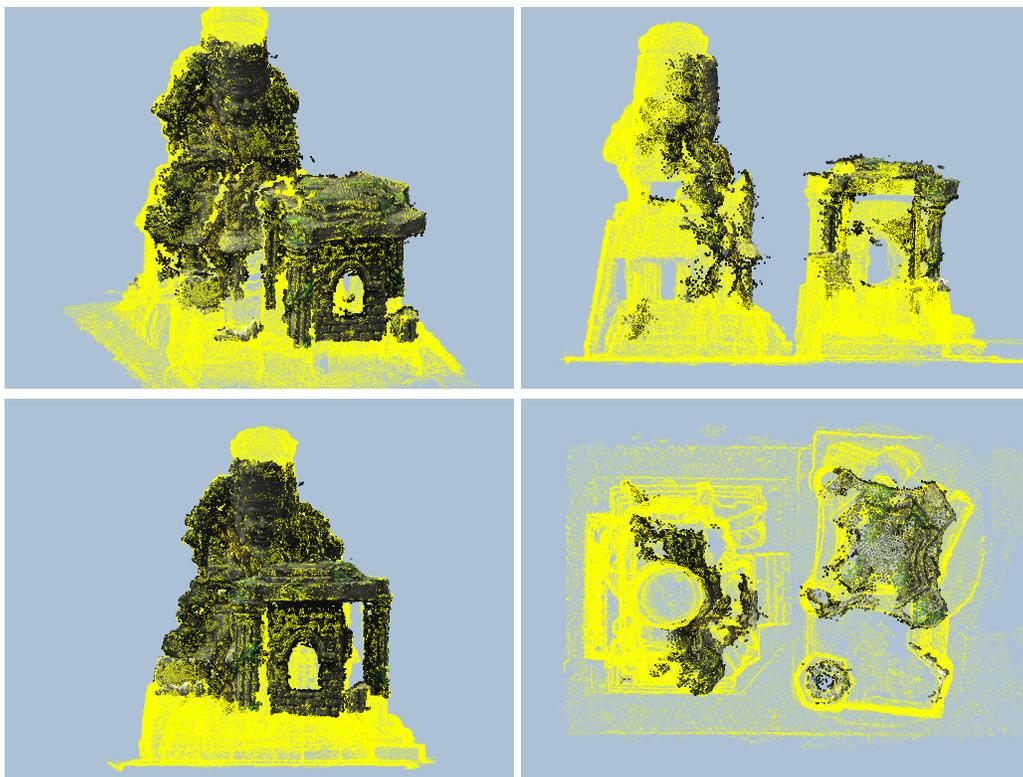


Figure 5.22: 3D reconstruction results for the FlatRefract method using eight views (fixed R,t), overlaid on yellow points of the ground truth model.

Reconstruction Results

Our calibration and 3D reconstruction method gave convincing results, as shown in Figure 5.22, using the full eight-camera array and the “fixed R,t” bundle adjustment configuration. The data shown in Figure 5.24 indicate that high accuracy was achieved, with a mean error of 0.53mm, and a scale factor error of 0.76%. In contrast, all of the 3D models reconstructed with the Rdist+Fadj method are noticeably distorted and have at least 10% scale error. Figure 5.23 shows a typical result using eight views, which appears horizontally compressed, while Figure A.6 shows the most visually acceptable model we obtained, which comes from using two views in the “fixed t” configuration. We believe that increased distortion occurs with eight views because the cameras at the left and right ends of the array are included, and they are more steeply tilted with respect to the refraction interfaces.

We also performed the 3D reconstruction using two and four views by taking subsets of the underwater images, in the same manner as in our synthetic data experiments. Compared to using all eight views, the accuracy of the FlatRefract method decreases, although it is still clearly better than Rdist+Fadj (see Figure 5.24). However, the scale factor error of around 3.7% is unexpectedly high. Our analysis on the impact of calibration errors in Subsection 5.3.2 showed that bundle adjustment of the refraction parameters can be problematic with too few views, and this appears to be the case here. (Although the “fixed R,t” configuration did not have significant scale errors in simulation, that experiment was done with perfect camera pose parameters, which is not the case with real data.)

Additional 3D reconstruction results using different bundle adjustment configurations are included in Appendix A.

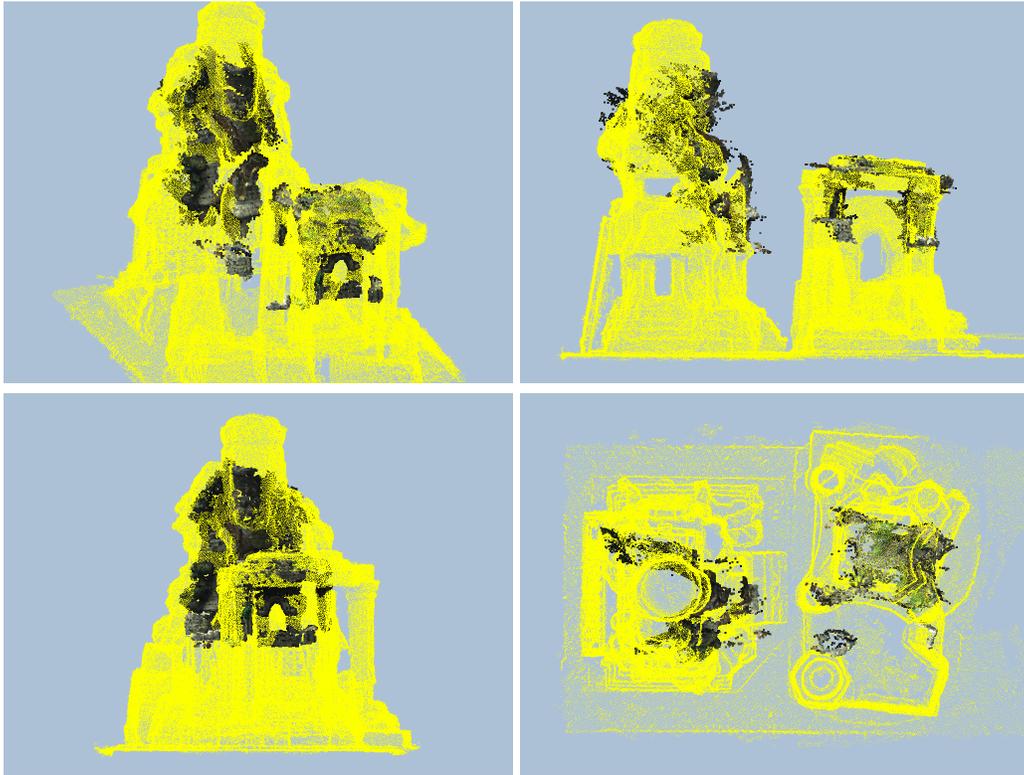


Figure 5.23: 3D reconstruction results for Rdist+Fadj using eight views (full adj), overlaid on yellow points of the ground truth model.

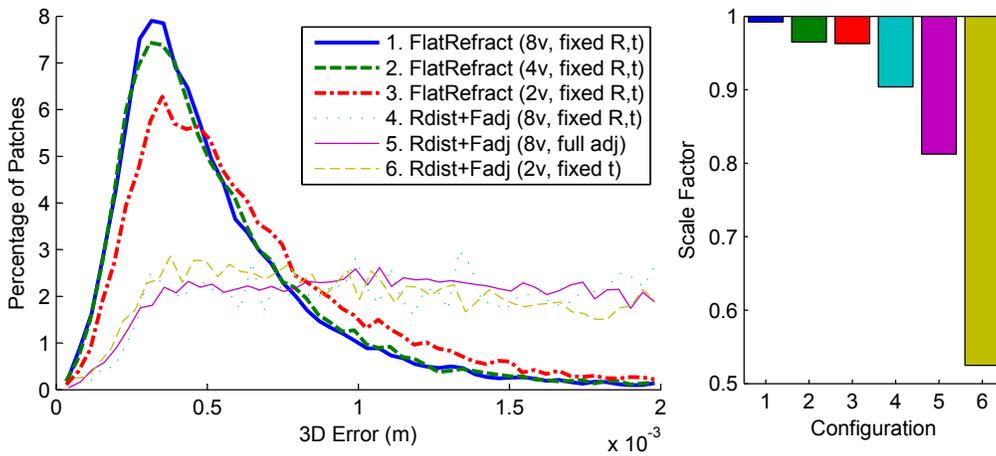


Figure 5.24: Accuracy of real data 3D reconstructions.

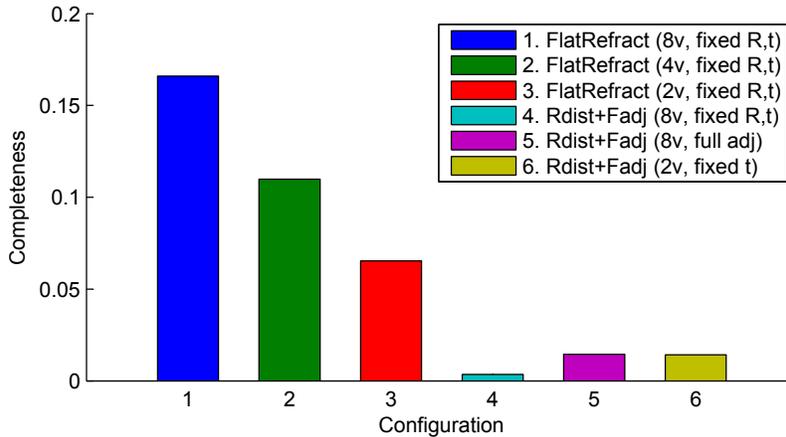


Figure 5.25: Completeness of real data 3D reconstructions.

Some noteworthy differences exist between the results in Figure 5.24 and our synthetic data experimental results. First, the vertical scale of the accuracy plot is significantly lower, and the error is more broadly spread out towards 1mm and above. Several potential factors could have contributed to this increase in error, including lens distortions and focusing issues, as well as the fact that the ground truth model is itself not perfect. We also noticed that the images contained some color-fringing and blurring caused by dispersion, which our method does not correct for, and which may interfere with the photo-consistency measure used by PMVS. (We initially assumed that a small amount of dispersion under normal lighting conditions would not present a problem. In hindsight, it may be advantageous to perform color channel warps using the refraction model parameters, analogous to the process described in Subsection 4.2.3.) Another possible factor is that the wall of our water tank is not perfectly flat (see 5.3.3, “Calibration”).

A second difference from our synthetic data results is that there is a steep decline in the number of points with error below 0.3mm. We believe this is largely due to the way that the error is measured, since the ground truth in this case is

a point cloud rather than a continuous surface (see 5.3.1), meaning that a point exactly on the surface will still not have zero error unless it coincides with a point on the ground truth model.

Third, the 0.76% error in scale factor for eight views is about an order of magnitude greater than in our synthetic data experiments. It is likely that at least part of this difference comes from physical measurement errors during the calibration and reconstruction process, including measurement of the checkerboard patterns used for intrinsic and extrinsic calibration and for rescaling of the ground truth model (Subsection 5.3.3).

Calibration

The calibrated refraction parameters visualized in Figure 5.26 and listed in Table A.1. While we did not have the ground truth available, the visualization suggests that a good calibration of the refraction axes was achieved, because the refraction interfaces are nearly parallel. On the other hand, the refraction interfaces do not line up as they should, being part of a single tank wall, meaning that distances d_0 are not very accurate. Fortunately, during the bundle adjustment phase of our 3D reconstruction procedure, much of this error was corrected to give the result shown in Figure 5.27.

Some difficulties in calibrating d_0 were noted in our previous experiments in Subsection 4.3.2. We investigated a number of possible reasons including lens distortion and chromatic aberration, but could not find an adequate explanation. Subsequently we discovered that the flat surface of our acrylic-walled tank was in fact not perfectly flat, with an outward deflection in the centre measuring about 3mm relative to the left and right edges, and 1.5mm relative to the top and bottom.

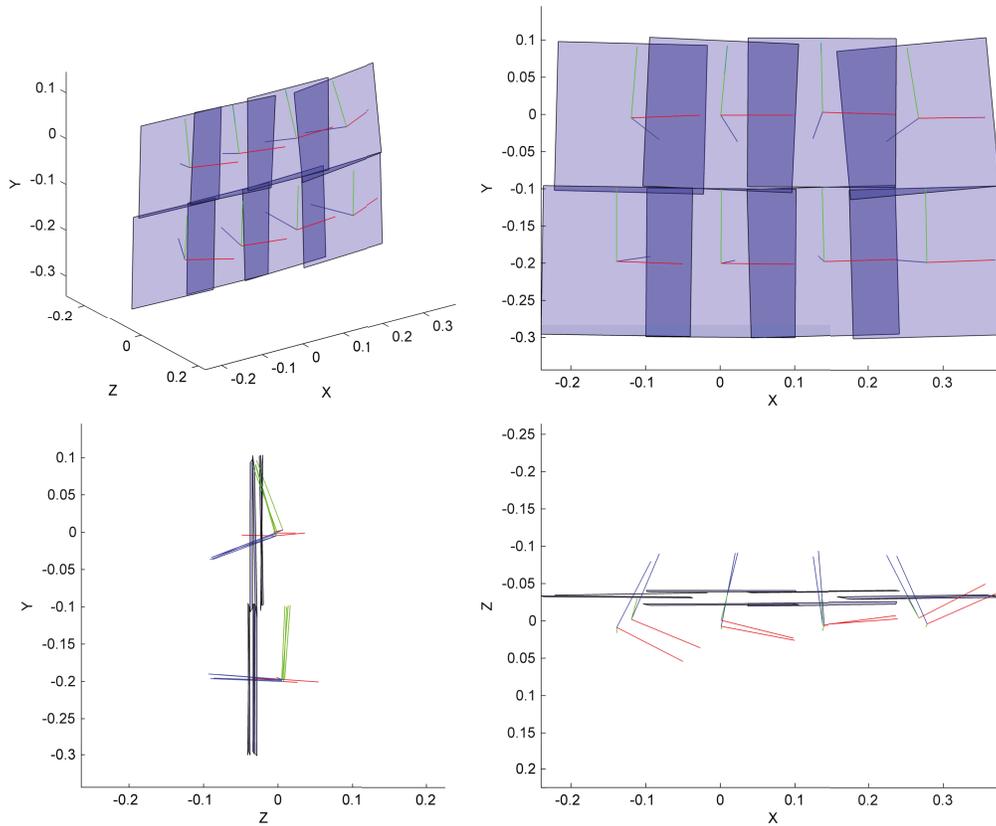


Figure 5.26: Camera array calibration result. Each camera is shown as a set of XYZ axes (red, green, blue), and translucent blue squares represent the refraction interface closest to it.

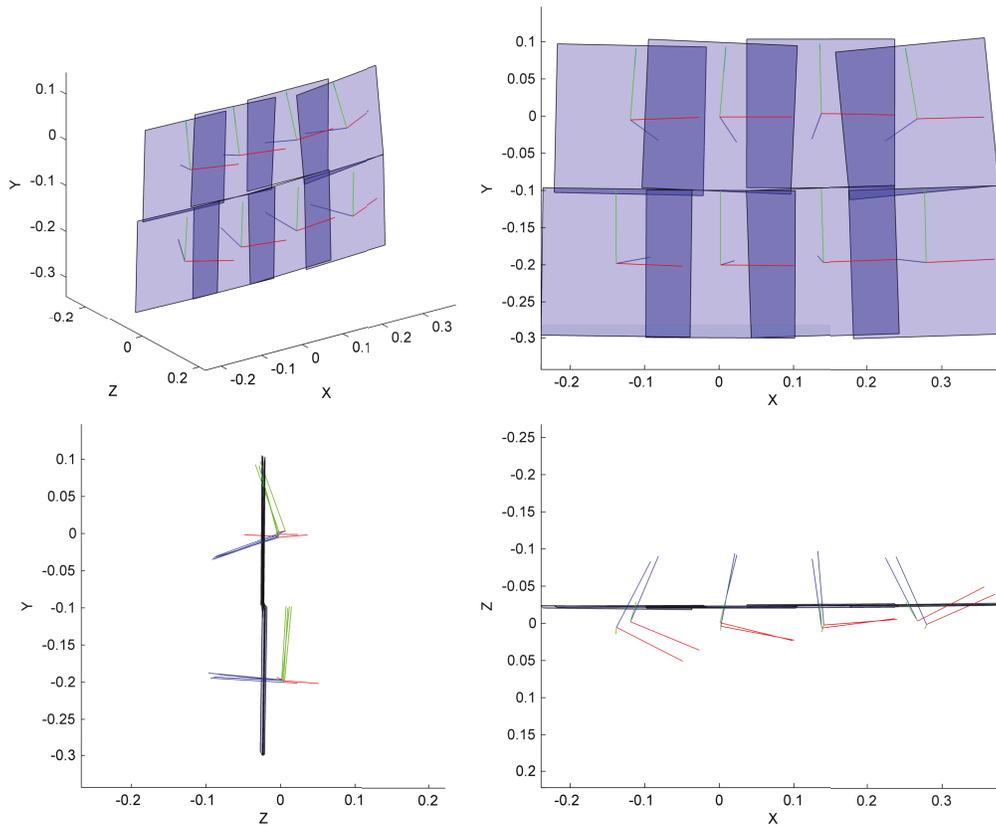


Figure 5.27: Camera array calibration result, after bundle adjustment. Each camera is shown as a set of XYZ axes (red, green, blue), and translucent blue squares represent the refraction interface closest to it.

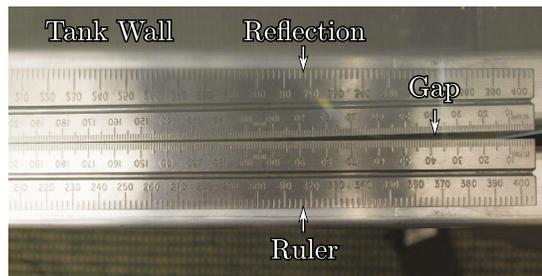


Figure 5.28: A straight edge placed against the wall of our water tank. The tank wall comes out of the page at the top of the image (the camera is looking down at the ground). The gap between the ruler and its reflection reveals that the wall surface is not completely flat.

This can be seen by placing a straight edge against the tank wall, as in Figure 5.28.

We conducted several simulated calibrations with the refraction interface modelled as a quadratic surface (forward projection was done by extending the method in Subsection 4.1.4 to use a two-dimensional optimization). When the refraction interface was tilted by 20 degrees, we observed errors in d_0 ranging from 6mm to 9mm depending on the orientation of the calibration pattern; when the refraction interface was not tilted, the error was lower at around 1mm. Although we did not collect enough results to reach a conclusion, further investigation may be merited. We believe that in most underwater applications, curvature of the refraction interfaces will not be an issue, since the ports of camera housings are usually small and made of glass that will not bend under pressure.

5.3.4 Application to Seabed Images

Having experimentally verified our calibration and 3D reconstruction methods in the laboratory, we now apply them to data acquired by the underwater observation platform mentioned in Chapter 1. This platform, operated by Ocean Networks Canada, is situated in an area known as Folger Passage, just off the west coast of Vancouver Island in British Columbia, Canada. It is mounted at a depth of 23m on a reef containing many organisms such as sponges, barnacles, and algae, whose growth and physical changes over time are of interest to marine biologists³.

Temp. (°C)	Salinity (psu)	Wavelength (nm)	Est. Refractive Index [42]
9.385	29.828	598	1.339

Table 5.2: Seawater conditions at time of capture of the undersea dataset, and its estimated refractive index for a typical light wavelength.

³For more information, please refer to: <http://www.oceannetworks.ca/installations/observatories/northeast-pacific/folger-passage>

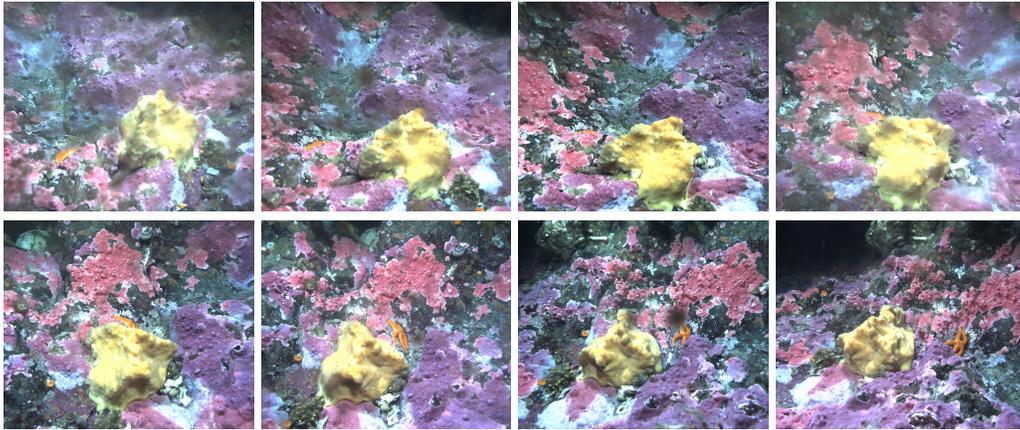


Figure 5.29: Undersea dataset from the Folger Passage observatory operated by Ocean Networks Canada. This dataset was captured on January 1, 2013 at 19:00:04.

Figure 5.29 shows the images in the dataset that we used, captured by the camera array in Figure 1.1. The cameras, which are Point Grey Research 5MP color Grasshoppers, have a resolution of 2448×2048 pixels and a 54 degree horizontal field of view, and are mounted in individual waterproof housings with flat glass ports. Eight of these cameras are rigidly mounted in a semicircle with radius 0.3m and angled downward by 30 degrees. Two LED lights illuminate the scene, since not much sunlight reaches the depth at which the platform is located.

We had the opportunity to capture a set of calibration data while the observation platform was being serviced on land. The calibration results shown in Figure A.11 were obtained by applying the same calibration steps described in Subsection 5.3.3. (Calibrated parameter values are given in Table A.2) We assumed a glass thickness of 6mm with refractive indices of 1.51421 and 1.5302 for 660nm and 405nm respectively. The refractive indices for water were assumed to be the same as in our laboratory experiments, since the cameras were immersed in fresh water when capturing the calibration data. However, for the 3D reconstruction

phase, the refractive index of sea water is needed. Although a direct measurement is not available, the observation platform includes instruments to measure water temperature and salinity. We use the empirical equation derived by Quan and Fry [42] to estimate the seawater refractive index, as presented in Table 5.2.

The reconstructed point clouds are shown in Figure 5.30, providing a visual comparison between the results from the Rdist+Fadj and FlatRefract methods. The most evident differences are found on the periphery of the reconstruction volume, with the FlatRefract method reconstructing additional regions along the right edge and on the top left. On the other hand, the Rdist+Fadj “fixed t” reconstruction contains a larger group of points in the bottom-left corner. As for main subject of the scene, the yellow sponge in the middle, we cannot say that any of the three results is definitely better than the others.

Although a ground truth model is not available for a quantitative evaluation, we can still compare the point clouds with each other. The FlatRefract reconstructed model contains 375,303 points, which is 4.7% more than than Rdist+Fadj “fixed t” with 358,374 points, and 7.9% more than Rdist+Fadj “full adj” with 347,752 points. By aligning the models using ICP, we found that the Rdist+Fadj “fixed t” model was 3.2% smaller than the refraction-corrected model, and the Rdist+Fadj “full adj” model was 1.7% larger. The disparity in scale can be seen in Figure 5.31. Considering our previous synthetic and real data results, it is likely that the Rdist+Fadj models are incorrectly scaled, while the FlatRefract model is probably close to actual scale.

Additionally, the ICP aligned point cloud models reveal some distortion in the scene. While difficult to show in static images, we observed that compared to the FlatRefract model, the left side of the Rdist+Fadj “full adj” model ap-

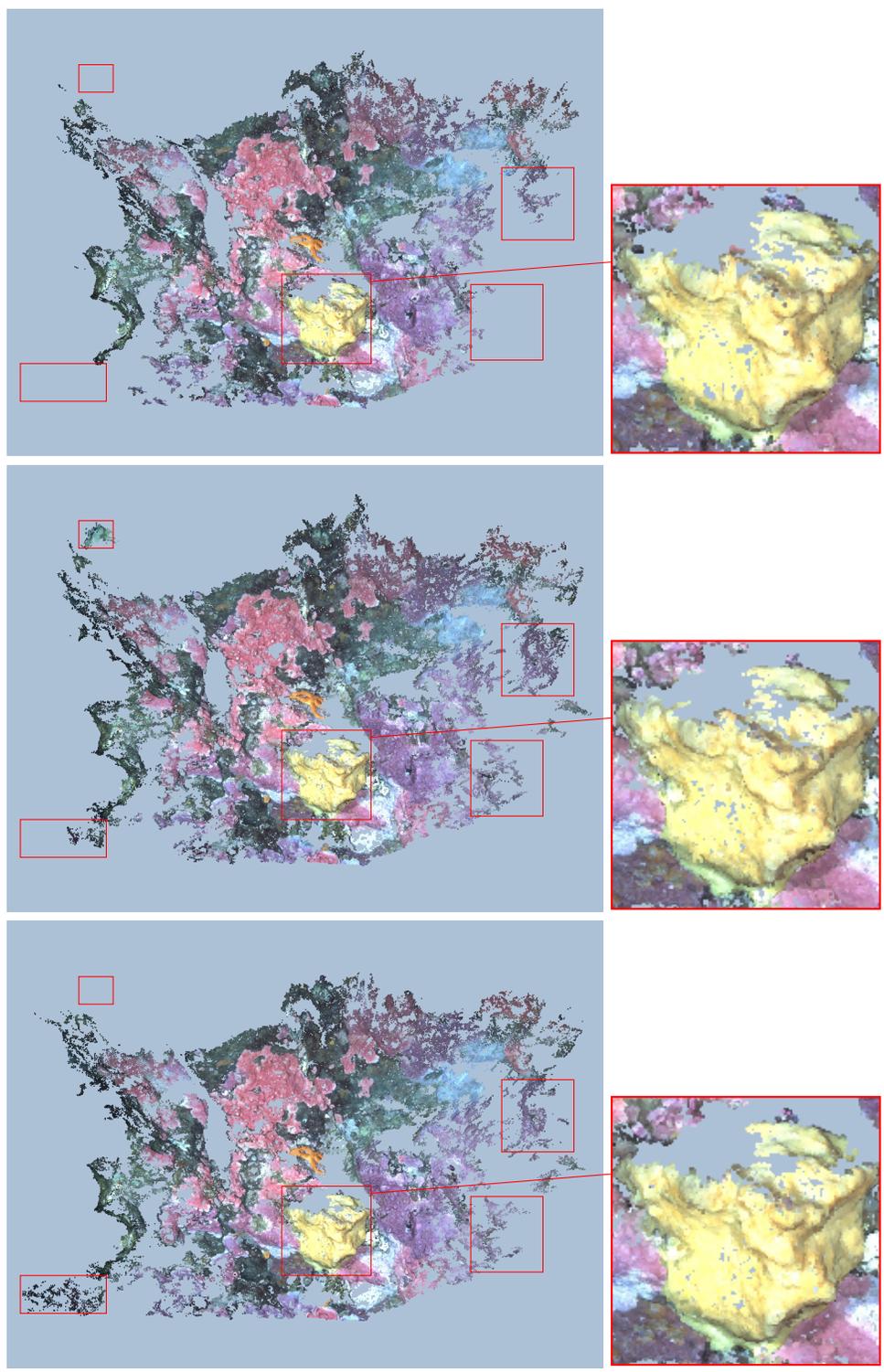


Figure 5.30: Qualitative comparison of the 3D reconstruction results using: (Top) Rdist+Fadj "full adj," (Middle) FlatRefract "fixed R,t," and (Bottom) Rdist+Fadj "fixed t." Regions of interest are highlighted.

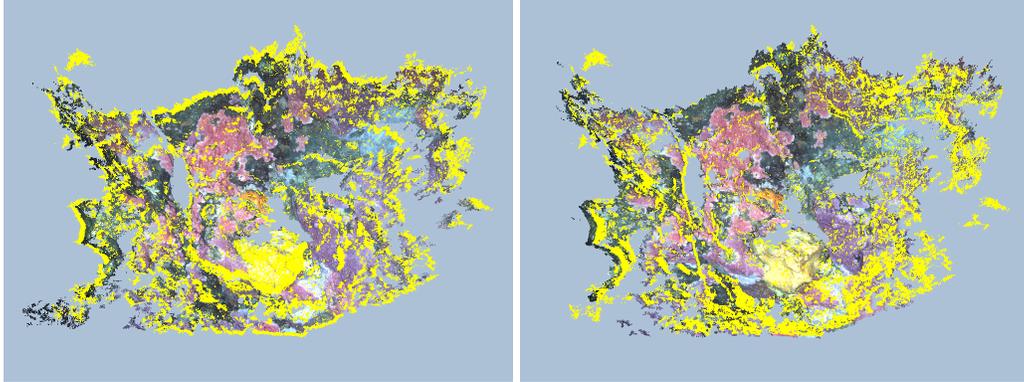


Figure 5.31: The 3D reconstruction from the FlatRefract method in yellow, overlaid with the reconstructions from: (Left) Rdist+Fadj “fixed t,” and (Right) Rdist+Fadj “full adj.” The difference in scale can be easily seen in both cases.

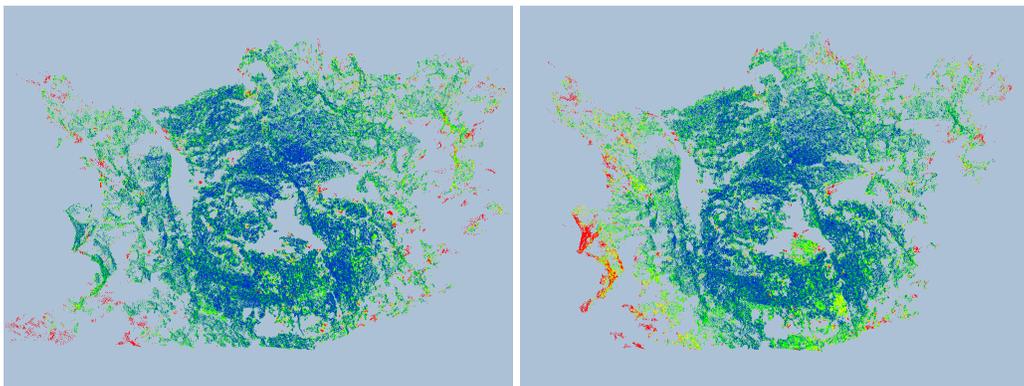


Figure 5.32: Point cloud distance between the FlatRefract reconstruction and: (Left) Rdist+Fadj “fixed t”, (Right) Rdist+Fadj “full adj.” These point cloud distances were computed after ICP alignment. The colors blue, green, yellow, and red correspond to point distances of approximately $\leq 0.5\text{mm}$, 1.1mm , 2.1mm , and $\geq 3.2\text{mm}$ respectively.

Configuration	Mean Dist (mm)	σ
Fadj+Rdist (full adj)	0.780	1.564
Fadj+Rdist (fixed t)	0.837	1.227

Table 5.3: Point cloud distance statistics after ICP alignment. Distances are measured against the point cloud reconstructed with the FlatRefract method.

pears stretched outwards, while the back of the yellow sponge is enlarged. The Rdist+Fadj “fixed t” model is more similar to the FlatRefract model, but differences are still visible in the periphery areas which are shifted inward at the upper left and right and outward at the lower left. We have attempted to illustrate these differences in Figure 5.32, which shows the closest distance from each point of the Rdist+Fadj models to the FlatRefract model⁴. Table 5.3 gives the summary statistics. Although this does not tell us which of the models being compared deviates more from the ground truth, we would suspect Rdist+Fadj based on the theory and our controlled experiments.

In summary, we have demonstrated that our new calibration and 3D reconstruction methods can be successfully used in a real underwater application, with favorable results. Several more views of the reconstructed 3D model are available in Appendix A.

⁴Point cloud distances were computed using the open-source CloudCompare software [16]

Chapter 6

Conclusions and Future Work

We began this thesis with an overview of recent developments in refractive geometry and underwater 3D reconstruction. This research area has been gaining popularity, with the increasing availability of underwater images through initiatives such as Ocean Networks Canada. A number of different authors have attempted to tackle the problems of calibration, 3D reconstruction, or both. We noted a trend toward the use of a flat refraction model as a representation of the physical imaging process. In Chapter 3, we gave a synthesis of some characteristics of this imaging model, including forward and back projection, and certain geometric constraints.

Building upon several insights about the flat refraction model, we were inspired to develop a novel calibration method. Our idea was to measure the dispersion of light, a side-effect of refraction that is often ignored, yet always present to some degree with flat port underwater camera housings. In Chapters 3 and 4 we demonstrated that the dispersion effect is not only measurable with a consumer-grade camera, but that it also leads to new geometric constraints on the refraction

model. By combining these constraints with an existing calibration method, we were able to obtain a new method involving a simpler set of equations. Experiments using simulated data showed that our method consistently achieves better accuracy for all calibration parameters.

We designed and created a new calibration device for use with our method. It emits a precise pattern of light in two distinct wavelengths, chosen to maximize the dispersion effect. We developed the necessary image processing techniques to decouple and measure the two wavelengths emitted simultaneously. Using this device and a large water tank, we performed a calibration experiment with real images. Quantitative evaluation against ground truth data confirmed that our method yielded better accuracy compared to previous work.

In Chapter 5 we set out to apply our calibration method in 3D reconstruction. We described a series of modifications to existing reconstruction algorithms, thoroughly integrating the flat refraction model in such a way that a fair comparison could be performed with the perspective camera model. We then detailed a procedure to calibrate the intrinsic, extrinsic, and refraction parameters of a camera array using our calibration method together with existing software.

Through a series of experiments with simulated data, we gained a better understanding of how the flat refraction model compares with the standard perspective model. Whereas previous works had been somewhat contradictory as to whether the perspective model works in underwater 3D reconstruction, we found that a clear explanation should take the scale factor of the scene into account. Reconstructions with the perspective model may change the scale factor error by up to several percent, and this could be construed as a reconstruction error; on the other hand, if the geometry is scaled to fit the ground truth, then the deviations

can be small in some situations. We determined that such deviations increase as the refraction interfaces are tilted with respect to the cameras, and that when the cameras are angled to view a scene behind a single set of refraction interfaces, the perspective model fails to give acceptable results. In contrast, the flat refraction model gave good results in all cases.

Next, we studied the impact of calibration errors and feature detection errors on 3D reconstruction with the flat refraction model. Both of these factors sometimes caused a discernible reduction in the reconstruction quality, and this was mainly due to the bundle adjustment optimization being unreliable. In particular, we discovered that optimizing camera pose and refraction parameters simultaneously often yielded poor results, even if scale factor changes were ignored. Nevertheless, with sufficiently many views, known or partially known camera pose parameters, and a reasonably accurate initial calibration, the bundle adjustment worked well as expected.

Lastly, we performed 3D reconstruction experiments using real images captured both in the laboratory and on the ocean floor. Our laboratory results included a comparison with ground truth data, which indicated that our refractive calibration and reconstruction methods yielded a highly accurate reconstruction. The same was not true of the perspective camera model, which did not give acceptable results.

The reconstruction results with ocean floor images was less clear-cut, due to a different camera array configuration in which the perspective model performed better. We compared the reconstructed point clouds visually and numerically to reveal some structural differences. Although ground truth data was not available in this case, indirect evidence suggested that the reconstruction using the refraction

model was likely more accurate and closer to the actual scale. This served to demonstrate a successful application of our methods in a real usage scenario.

6.1 Future Work

One of the main difficulties that we experienced in performing 3D reconstructions with real images was with bundle adjustment. Errors arising in the calibration of camera pose and refraction parameters meant that this was a necessary step, but our implementation based on Bundler was not always reliable. It would be useful to investigate new error functions besides reprojection error, perhaps add soft constraints to various parameters, and tune the optimization options. There might also be room for improvement in the feature detection, matching, and outlier rejection processes, as the existing algorithms do not account for refraction. In our experience, such enhancements would be especially beneficial when using only two views.

If a reliable bundle adjustment method could be found, it would be interesting to extend our work to support multiple image sets with a moving camera array. Currently our method requires a calibrated camera pose for each image, so for an eight-view reconstruction we used eight cameras mounted rigidly together. A more flexible approach would be to use, for example, a rig with only two cameras to take multiple images or videos of the scene from different angles, and combine them in a single 3D reconstruction. This would provide increased scene coverage, while still reconstructing the scene at the actual scale because of the fully-calibrated camera rig.

Our calibration method, while offering accurate results, has some obvious lim-

itations. Most notably, our calibration device is rather bulky and only suitable for use in controlled environments. It may be possible to create a device similar to a checkerboard pattern in size by using different optical components such as light pipes and fiber optics. Conversely, a powered calibration device might not be needed at all if it could be replaced by a lighting unit, which illuminates a general scene with two distinct wavelengths.

A second limitation of our method is that precise refractive index values are needed for each wavelength, and the values must be accurate at least relative to each other. Further investigation is needed to determine how accurate the values need to be in absolute terms. One option is to calibrate the refraction parameters in a controlled environment, since underwater cameras usually do not move within their housings when deployed.

Lastly, our calibration procedure for an entire camera array is quite involved, requiring four sets of images for the camera intrinsic and extrinsic parameters, the refraction model parameters, and chromatic aberration correction. It should be possible to calibrate both the camera extrinsic (pose) parameters and refraction model parameters with a single set of images using the calibration device. This would save time when working with large camera arrays.

Other potential improvements for our calibration and 3D reconstruction methods include: adding GPU acceleration for refractive forward projection, making the detection of the calibration pattern points more robust, and correcting for dispersion in underwater images using the calibrated refraction model. The last of these may be useful for underwater photography in general, and not just for 3D reconstruction.

Finally, we would like to mention two research ideas related to dispersion that

we did not have time to explore. The first is that 3D reconstruction using a single view might be possible by illuminating a scene in two different wavelengths, and imaging it through a refractive interface. The effect is similar to the work of Chen *et al.* [11], but using only a single image and without requiring that the refractive medium be removed.

The second idea relates to the work of Kutulakos and Steger [33] on reconstructing refractive surfaces. We hypothesize that by using two or three different wavelengths, additional constraints are imposed that allow more refractive surfaces to be recovered, or the same number of surfaces but with fewer views. We are not sure if this can be made practical, but it seems to be at least of theoretical interest.

Bibliography

- [1] A. Agrawal and S. Ramalingam. Single Image Calibration of Multi-axial Imaging Systems. *CVPR*, 2013.
- [2] A. Agrawal, S. Ramalingam, Y. Taguchi, and V. Chari. A theory of multi-layer flat refractive geometry. In *CVPR*, 2012.
- [3] Ocean Networks Canada Data Archive. <http://www.oceannetworks.ca>. Ocean Networks Canada, University of Victoria, Canada. Accessed: 07-04-2014.
- [4] J.-Y. Bouguet. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/. Accessed: 19-08-2013.
- [5] J. K. Bourne. Robots of the gulf spill: Fishlike subs, smart torpedoes. <http://news.nationalgeographic.com/news/2010/10/101026-bp-gulf-oil-spill-robots-science-nsf/>. Accessed: 03-09-2013.
- [6] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

- [7] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37:855–866, 1971.
- [8] J. Cameron. Ghostwalking in titanic. <http://ngm.nationalgeographic.com/2012/04/titanic/cameron-text>. Accessed: 03-09-2013.
- [9] Y.-J. Chang and T. Chen. Multi-view 3d reconstruction for scenes under the refractive plane with known vertical direction. In *ICCV*, 2011.
- [10] V. Chari and P. Sturm. Multi-view geometry of the refractive plane. In *BMVC*, 2009.
- [11] Z. Chen, K.-Y. K. Wong, Y. Matsushita, X. Zhu, and M. Liu. Self-calibrating depth from refraction. In *ICCV*, 2011.
- [12] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *TPAMI*, 18(11):1098–1104, 1996.
- [13] W. D. Curtis, A. L. Janin, and K. Zikan. A note on averaging rotations. In *Virtual Reality Annual International Symposium*, pages 377–385, Sep 1993.
- [14] M. Daimon and A. Masumura. Measurement of the refractive index of distilled water from the near-infrared region to the ultraviolet region. *Appl. Opt.*, 46(18):3811–3820, 2007.
- [15] B. de Greve. Reflections and refractions in ray tracing. http://www.bramz.net/data/writings/reflection_transmission.pdf. Accessed: 11-02-2014.

- [16] EDF R&D, Telecom ParisTech (2013). Cloudcompare (version 2.5) [GPL software]. <http://www.danielgm.net/cc/>. Accessed: 7-11-2013.
- [17] P. Favaro and S. Soatto. A geometric approach to shape from defocus. *TPAMI*, 27(3):406–417, March 2005.
- [18] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *TPAMI*, 32(8):1362–1376, 2010.
- [19] J. Gedge. Underwater stereo matching and its calibration. Master’s thesis, University of Alberta, 2011.
- [20] J. Gedge, M. Gong, and Y.-H. Yang. Refractive epipolar geometry for underwater stereo matching. In *CRV*, pages 146–152, 2011.
- [21] G. Glaeser and H. P. Schröcker. Reflections on refractions. *Journal for Geometry and Graphics*, 4(1):1–18, 2000.
- [22] C. Harris and M. Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.
- [23] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, November 1997.
- [24] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [25] N. Hasler, B. Rosenhahn, T. Thormahlen, M. Wand, J. Gall, and H. P. Seidel. Markerless motion capture with unsynchronized moving cameras. In *CVPR*, 2009.

- [26] A. Jordt-Sedlazeck, D. Jung, and R. Koch. Refractive plane sweep for underwater images. In *Pattern Recognition*, volume 8142 of *Lecture Notes in Computer Science*, pages 333–342. Springer Berlin Heidelberg, 2013.
- [27] A. Jordt-Sedlazeck and R. Koch. Refractive calibration of underwater cameras. In *ECCV (5)*, volume 7576 of *Lecture Notes in Computer Science*, pages 846–859. Springer, 2012.
- [28] A. Jordt-Sedlazeck and R. Koch. Refractive structure-from-motion on underwater images. In *ICCV*, 2013.
- [29] F. Kahl and R. Hartley. Multiple-View Geometry Under the L_∞ -Norm. *TPAMI*, 30(9):1603–1617, 2008.
- [30] L. Kang, L. Wu, and Y.-H. Yang. Experimental study of the influence of refraction on underwater three-dimensional reconstruction using the syp camera model. *Appl. Opt.*, 51(31):7591–7603, Nov 2012.
- [31] L. Kang, L. Wu, and Y.-H. Yang. Two-view underwater structure and motion for cameras under flat refractive interfaces. *ECCV*, 2012.
- [32] S. N. Kasarova, N. G. Sultanova, C. D. Ivanov, and I. D. Nikolov. Analysis of the dispersion of optical plastic materials. *Optical Materials*, 29(11):1481–1490, 2007.
- [33] K. N. Kutulakos and E. Steger. A theory of refractive and specular 3d shape by light-path triangulation. In *ICCV*, 2005.

- [34] R. Li, H. Li, W. Zou, R. G. Smith, and T. A. Curran. Quantitative photogrammetric analysis of digital underwater video imagery. *Oceanic Engineering*, 22(2):364–375, 1997.
- [35] M. I. A. Lourakis and A. A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30, 2009.
- [36] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, November 2004.
- [37] N. J. W. Morris and K. N. Kutulakos. Dynamic refraction stereo. In *ICCV*, 2005.
- [38] N. J. W. Morris and K. N. Kutulakos. Reconstructing the surface of inhomogeneous transparent scenes by scatter-trace photography. In *ICCV*, 2007.
- [39] D. Nistér. An efficient solution to the five-point relative pose problem. *TPAMI*, 26(6):756–77, June 2004.
- [40] Persistence of Vision Pty. Ltd. (2004). Persistence of vision raytracer (version 3.6). <http://www.povray.org/download/>. Accessed: 30-09-2013.
- [41] M. Polyanskiy. Refractiveindex.info refractive index database. <http://refractiveindex.info/>. Accessed: 10-02-2014.
- [42] X. Quan and E. S. Fry. Empirical equation for the index of refraction of seawater. *Appl. Opt.*, 34(18):3477–3480, Jun 1995.
- [43] D. Scharstein. Multi-view stereo. <http://vision.middlebury.edu/mview/>. Accessed: 13-08-2013.

- [44] A. Sedlazeck and R. Koch. Calibration of housing parameters for underwater stereo-camera rigs. In *BMVC*, 2011.
- [45] A. Sedlazeck and R. Koch. Simulating deep sea underwater images using physical models for light attenuation, scattering, and refraction. In *VMV*, pages 49–56. Eurographics Association, 2011.
- [46] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006.
- [47] M. Shortis, E. Harvey, and J. Seager. A review of the status and trends in underwater videometric measurement. *SPIE Conf. 6491, Videometrics IX*, 2007.
- [48] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH*, 2006.
- [49] K. V. Stefanik, J. C. Gassaway, K. Kochersberger, and A. L. Abbott. Uav-based stereo vision for rapid aerial terrain mapping. *GIScience & Remote Sensing*, 48(1):24–49, 2011.
- [50] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *ECCV*, 1996.
- [51] G. Telem and S. Filin. Photogrammetric modeling of underwater environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(5):433 – 444, 2010.

- [52] E. Tola, V. Lepetit, and P. Fua. DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. *TPAMI*, 32(5):815–830, May 2010.
- [53] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 9(2):137–154, 1992.
- [54] T. Treibitz, Y. Y. Schechner, and H. Singh. Flat refractive geometry. In *CVPR*, 2008.
- [55] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape-from-shading: a survey. *TPAMI*, 21(8):690–706, Aug 1999.
- [56] M. Zuliani. RANSAC for Dummies. Technical report, UCSB, January 2012.

Appendix A

Supplementary Experimental Data

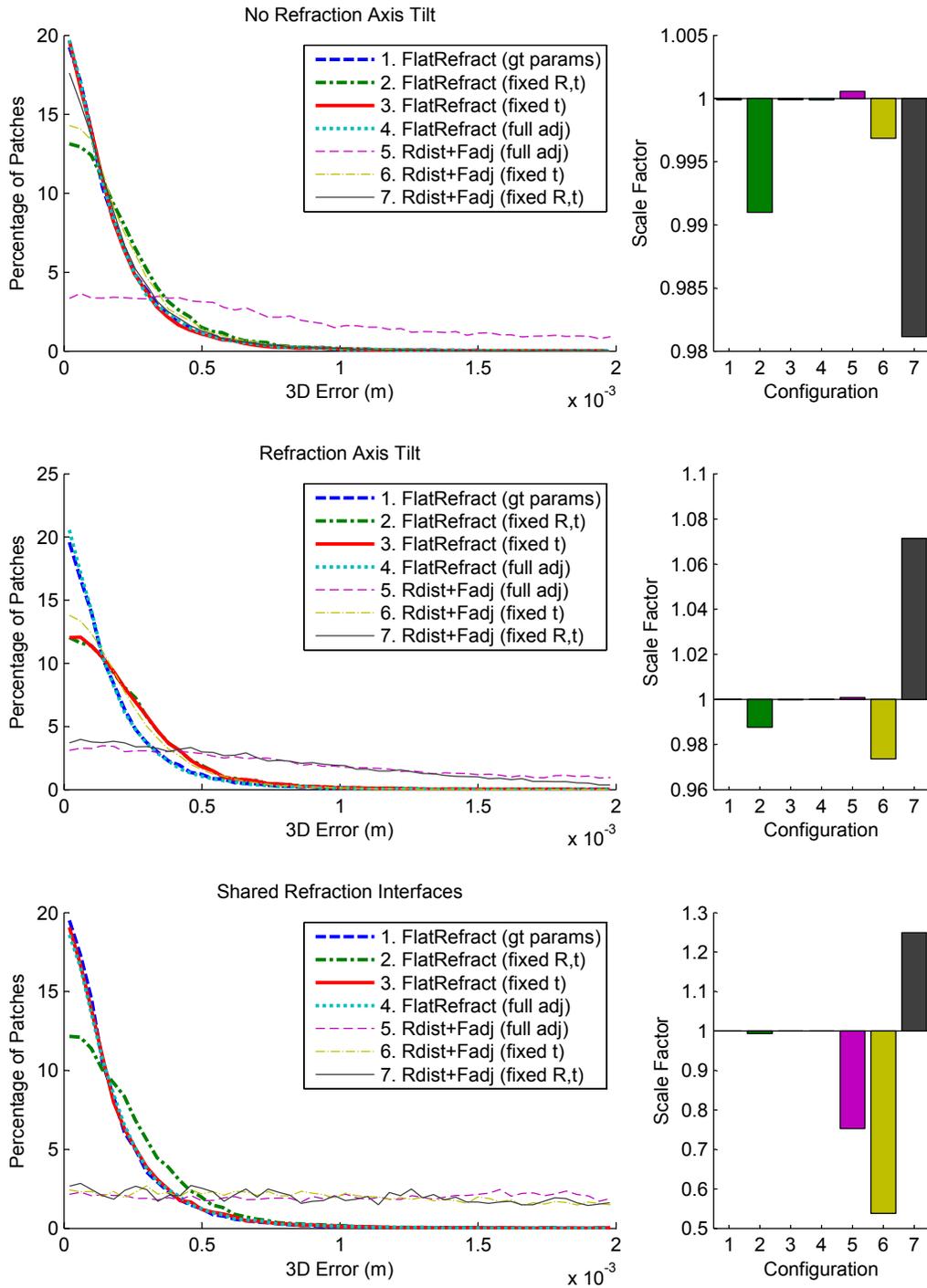


Figure A.1: Error distribution and scale factor plots for two views, using feature points detected by Bundler. See Subsection 5.3.2.

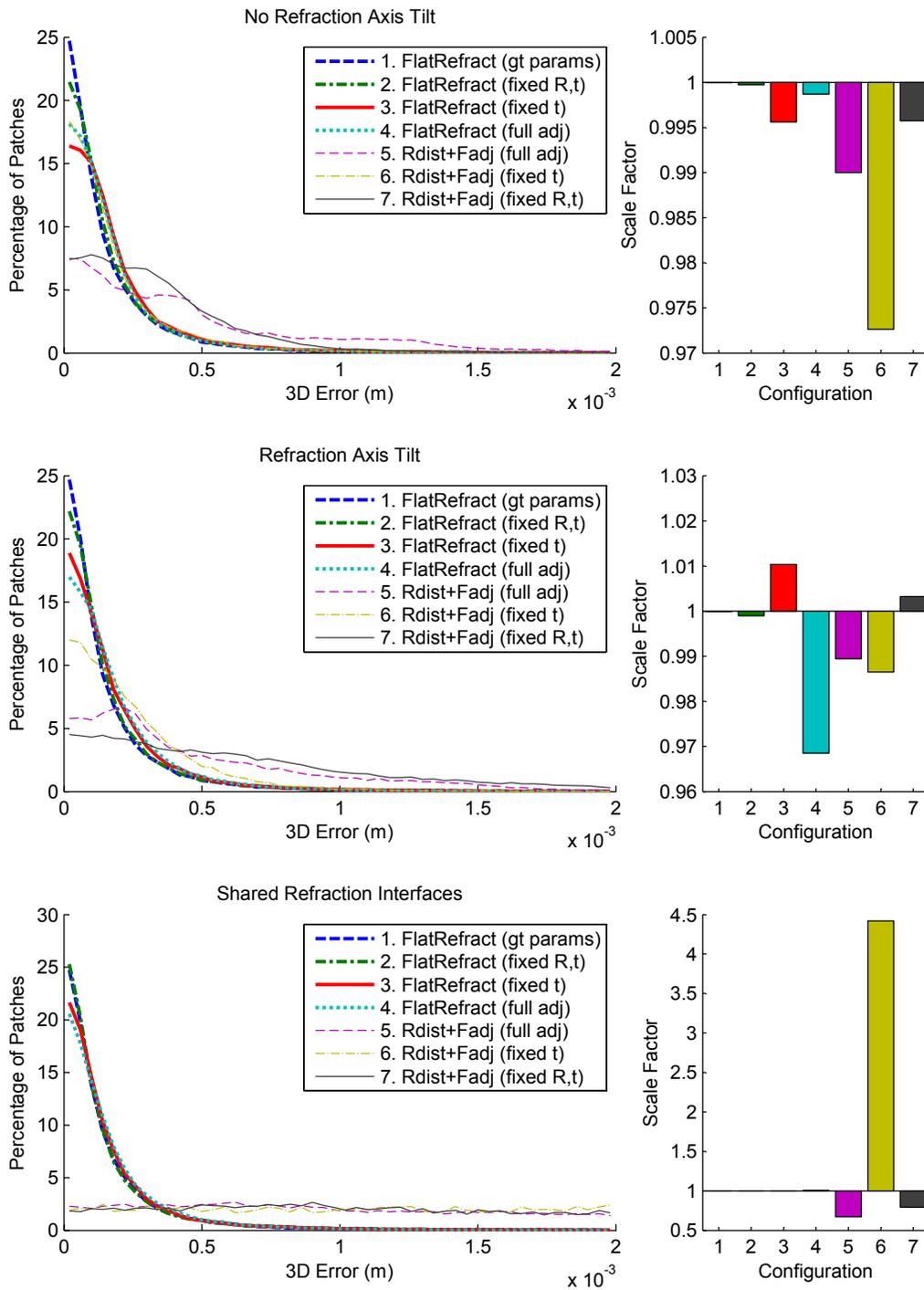


Figure A.2: Error distribution and scale factor plots for four views, using feature points detected by Bundler. See Subsection 5.3.2.

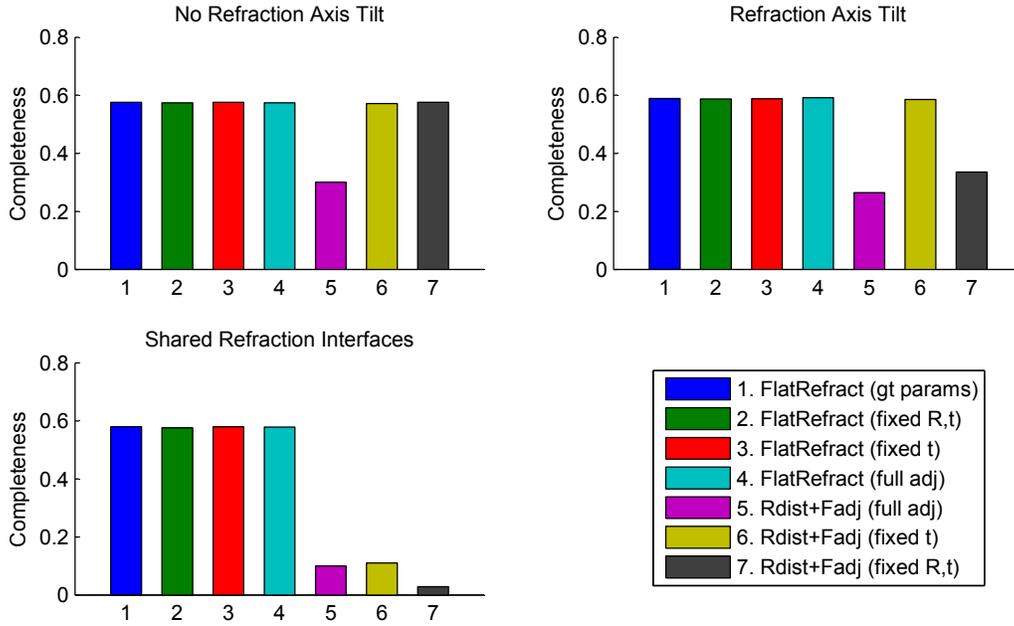


Figure A.3: Completeness measure for two views, using feature points detected by Bundler. See Subsection 5.3.2.

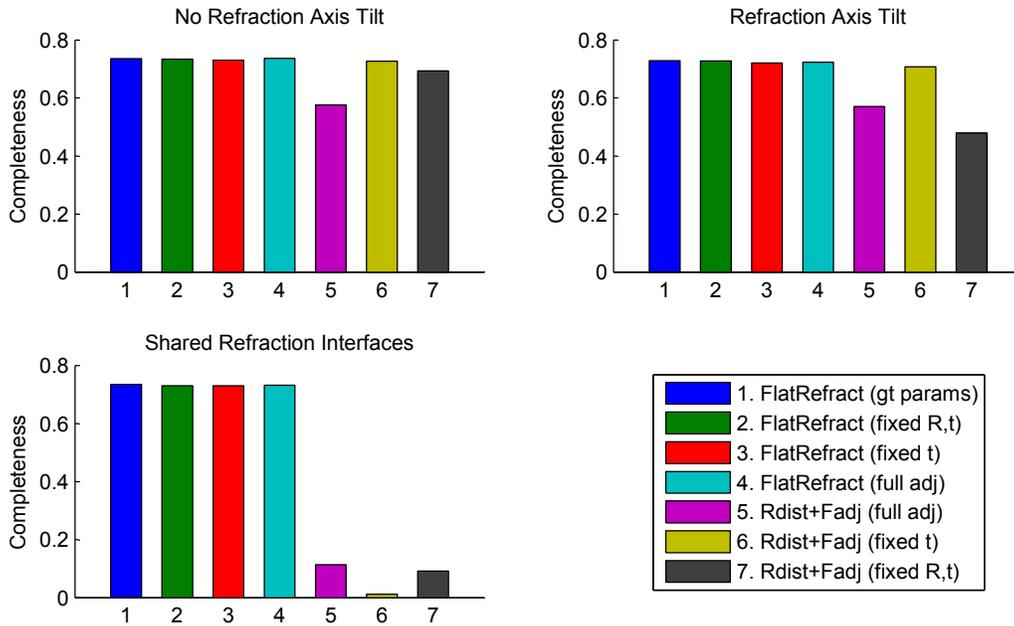


Figure A.4: Completeness measure for four views, using feature points detected by Bundler. See Subsection 5.3.2.

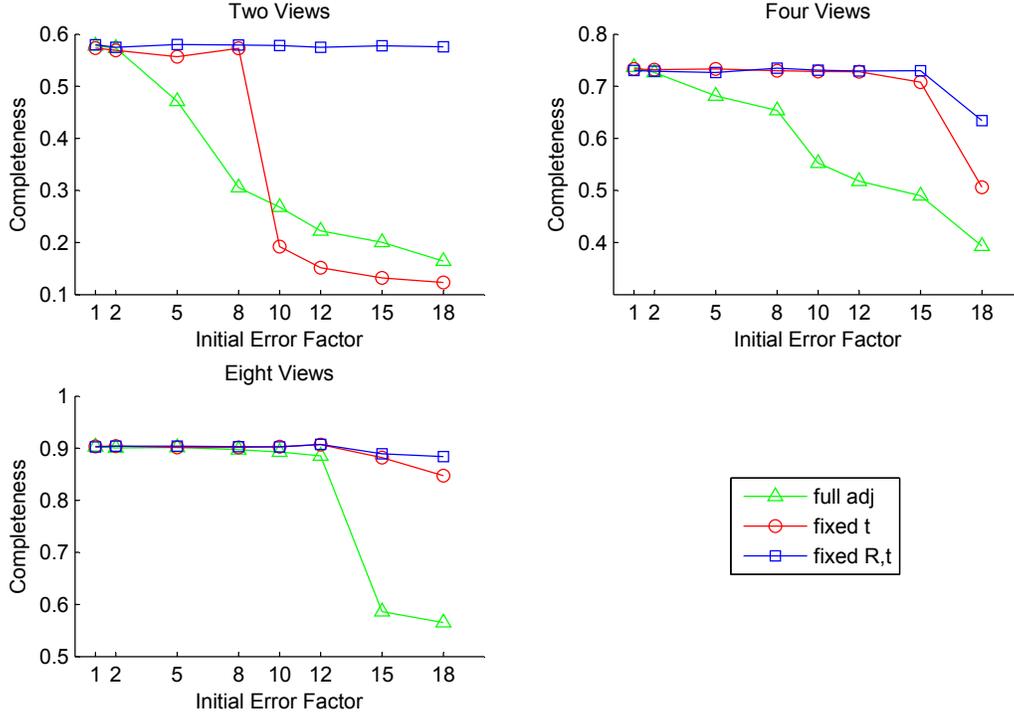


Figure A.5: 3D Reconstruction completeness results for simulated data with varying amounts of error in the initial refraction model parameters. See Subsection 5.3.2

Camera	Initial Calibration			After Bundle Adjustment		
	ϕ	θ	d_0	ϕ	θ	d_0
1	16.38	24.34	0.0341	16.02	23.61	0.0207
2	20.80	15.15	0.0213	19.72	15.21	0.0215
3	19.79	-7.26	0.0289	20.23	-7.39	0.0315
4	14.94	-27.54	0.0282	15.09	-27.82	0.0214
5	-3.92	26.84	0.0409	-4.30	26.59	0.0264
6	-1.78	10.79	0.0472	-2.45	10.86	0.0257
7	-4.74	-3.86	0.0447	-4.74	-4.01	0.0247
8	-2.40	-22.42	0.0358	-2.57	-23.06	0.0258

Table A.1: Calibration data for our 3D reconstruction experiments with a water tank. The cameras are in two rows of four, and numbered left-to-right and top-to-bottom. The Euler angles ϕ and θ give the rotation of the axis of refraction about the X and Y axes respectively, and are specified in a left-handed coordinate system. See Subsection 5.3.3.

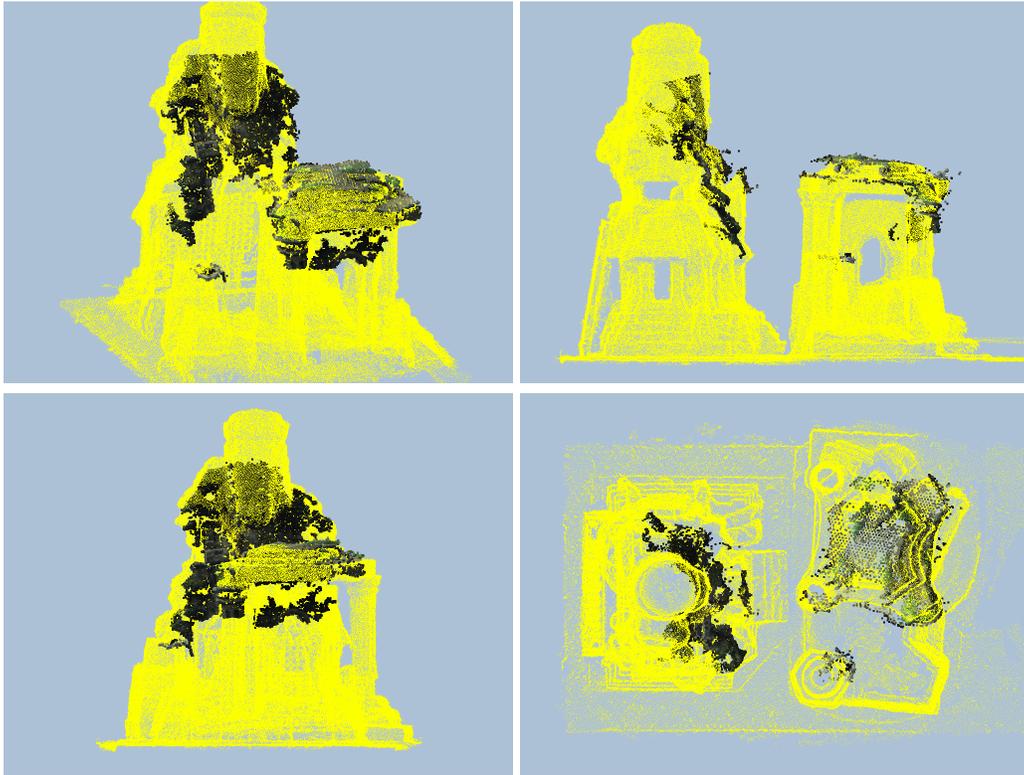


Figure A.6: 3D reconstruction results for Rdist+Fadj using two views (fixed t), overlaid on yellow points of the ground truth model. See Subsection 5.3.3.

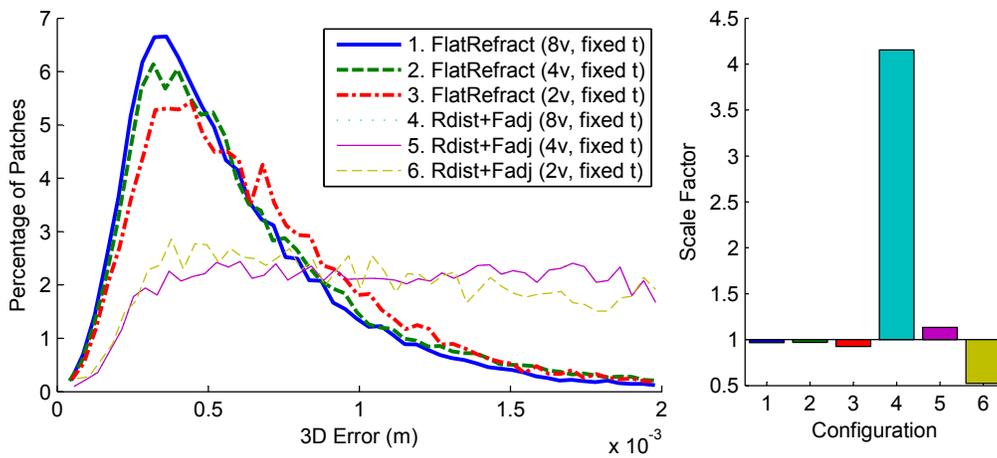


Figure A.7: Accuracy of real data 3D reconstructions. See Subsection 5.3.3.

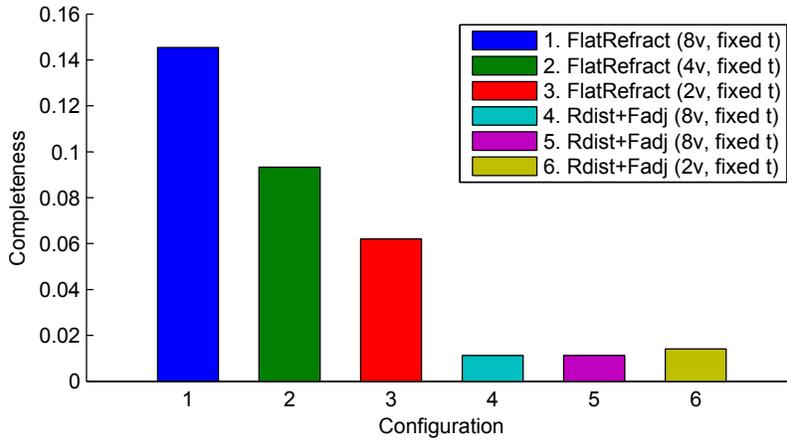


Figure A.8: Completeness of real data 3D reconstructions. See Subsection 5.3.3.

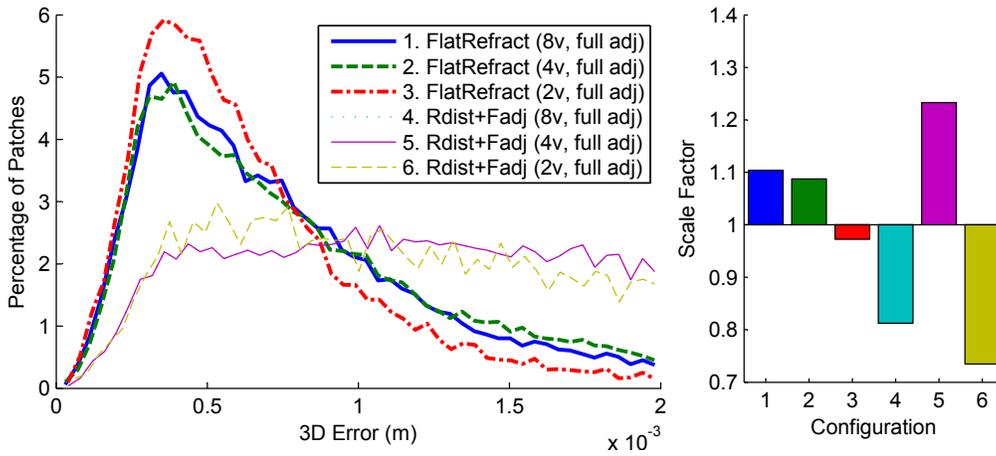


Figure A.9: Accuracy of real data 3D reconstructions. See Subsection 5.3.3.

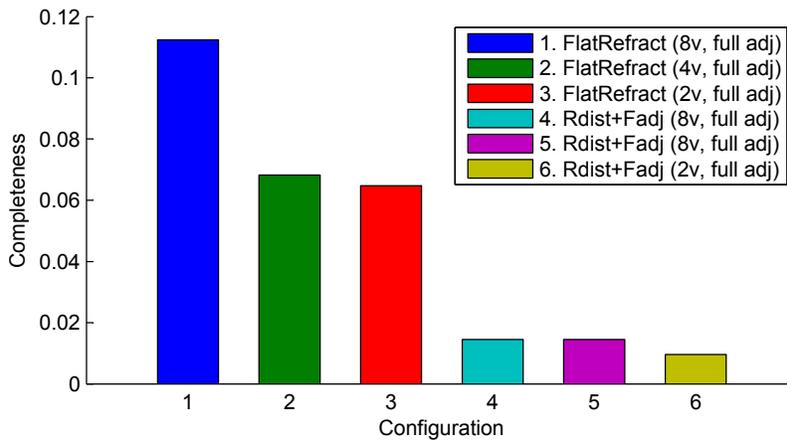


Figure A.10: Completeness of real data 3D reconstructions. See Subsection 5.3.3.

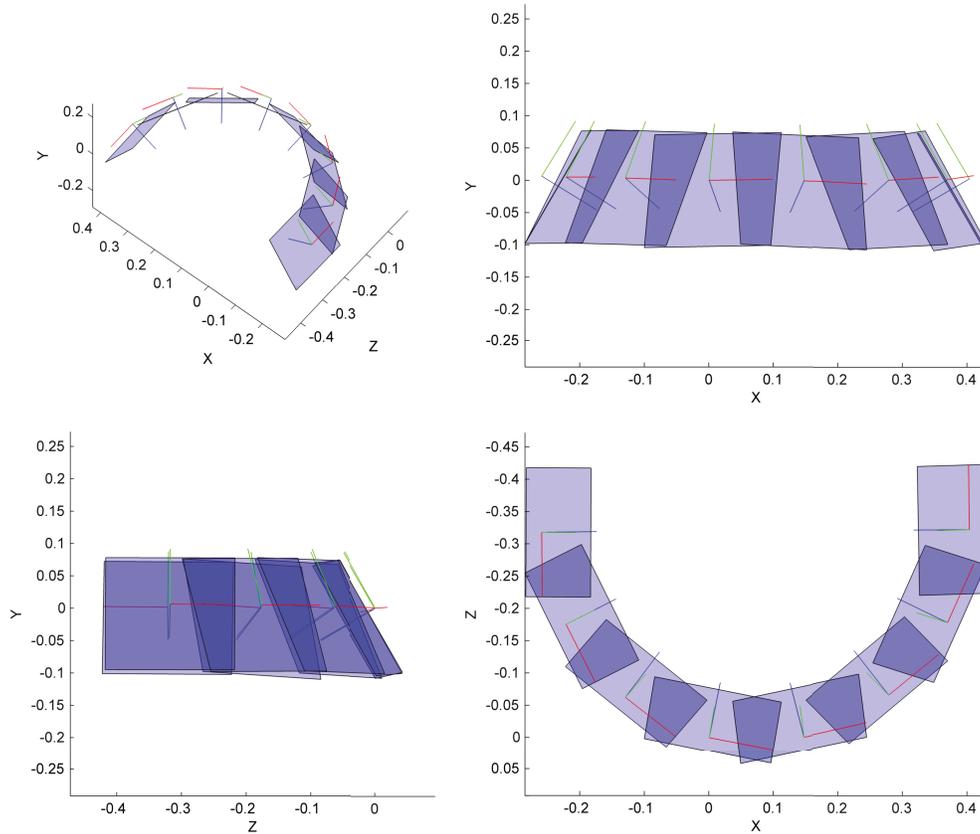


Figure A.11: Calibration result for the Folger Passage underwater observation platform camera array, after bundle adjustment. Each camera is shown as a set of XYZ axes (red, green, blue), and translucent blue squares represent the refraction interface closest to it. See Subsection 5.3.4.

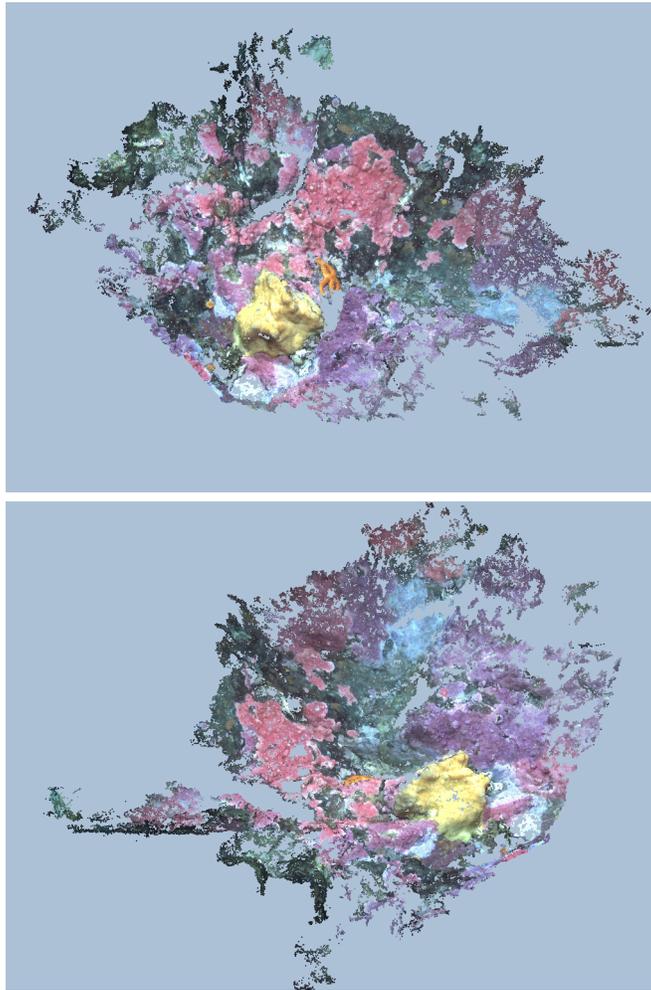


Figure A.12: Additional views of the point cloud reconstructed with the FlatRefract method. See Subsection 5.3.4.

Camera	Initial Calibration			After Bundle Adjustment		
	ϕ	θ	d_0	ϕ	θ	d_0
1	1.12	-0.19	0.0275	1.50	-0.81	0.0300
2	0.04	0.14	0.0257	0.30	-0.57	0.0314
3	-0.37	-0.22	0.0280	-0.27	-0.81	0.0313
4	0.02	-0.01	0.0251	-0.00	-0.44	0.0313
5	0.24	-0.67	0.0253	0.15	-0.81	0.0325
6	0.74	0.71	0.0284	0.70	0.85	0.0335
7	0.48	-0.37	0.0306	0.49	-0.04	0.0358
8	0.35	-0.14	0.0289	0.44	0.38	0.0334

Table A.2: Calibration data for the Folger Passage underwater observation platform camera array. The Euler angles ϕ and θ give the rotation of the axis of refraction about the X and Y axes respectively, and are specified in a left-handed coordinate system. See Subsection 5.3.4.

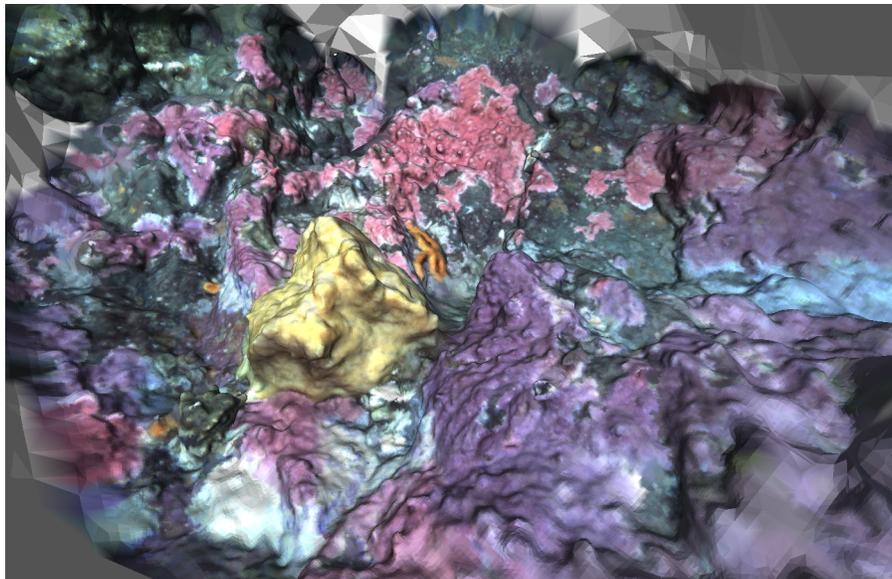


Figure A.13: Reconstructed 3D model with the FlatRefract method, after applying Poisson Surface Reconstruction. See Subsection 5.3.4.