Structured Light Methods: From Land to Undersea

by

Xida Chen

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

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

Extracting 3D geometry of an object from 2D images has been a popular topic in computer vision for decades. Many methods have been proposed to solve this problem with high accuracy and structured light methods are one of the most commonly used. Despite their high accuracy, there are limitations for existing methods. First, most existing methods can either be applied to obtain dense correspondences but limited to static scenes, or be applied to dynamic scenes but limited to sparse correspondences. Second, existing methods cannot be applied to scenes with global illuminations such as inter-reflection and projector defocus. Last but not least, existing structured light methods are rarely applied to underwater applications due to the difficulties of underwater camera calibration.

Motivated by the above limitations, a new structured light method is presented to establish dense correspondences for dynamic scenes. Many simulated and real datasets are used to test the robustness of this method. Moreover, another novel structured light method is presented in this thesis to account for global illuminations such as inter-reflection, subsurface scattering and severe projector defocus. The experimental results demonstrate that the proposed method consistently outperforms two of the state-of-the-art methods. To address the limitation in underwater applications, two new underwater camera calibration methods are presented by using the physically correct refraction model. All the experimental results of applying these two methods are evaluated against the ground truth, and the simulated experiments are compared to the methods that produces the best results. The comparison demon-

strates that our results are promising. The most significant advantage of these two methods is that no calibration object is required, which can be very difficult to access when the cameras are deployed undersea. Finally, a multi-camera multi-projector system is designed and developed to monitor the undersea habitat. Since 2014, the system has been deployed along the coast of B.C., collecting data that can be shared with researchers from around the world.

Preface

Research for this thesis was conducted under the supervision of Dr. Herbert Yang at the University of Alberta. Part of Chapter 3 was published as X. Chen and Y.H. Yang, "Recovering Stereo Depth Maps using a Single Gaussian Blurred Structured Light Pattern," *Canadian Conference on Computer and Robot Vision*, May 28-31, 2013, pp. 295-302. Portions of Chapter 4 were published as X. Chen and Y.H. Yang, "Scene Adaptive Structured Light using Error Detection and Correction," *Pattern Recognition*, Vol. 48, Issue 1, 2015, pp. 220-230. Part of Chapter 5 was published as X. Chen and Y.H. Yang, "Two-view Camera Housing Parameters Calibration for Multi-Layer Flat Refractive Interface," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 24-27, 2014, Columbus, Ohio. I, Xida Chen, was in charge of developing the concept and algorithms, performing experiments and analysis, and manuscripts composition, while my supervisor provided guidance and comments on the manuscripts.

Acknowledgements

First and the most important, I would like to express my deepest gratitude to Dr. Herbert Yang for his invaluable guidance and constant support throughout my entire Ph.D program. He provided much knowledge and insight. I have learned a tremendous amount under his supervision. This thesis would not be possible without him.

I also thank all my committee members for their precious time and valuable suggestions on this thesis.

I would like to specially thank Steve Sutphen from the Department of Computing Science, who has provided lots of technical assistance. He was responsible for the detailed hardware design and assembly of the deployed undersea camera system described in this thesis. The system would not be successfully built and deployed without his help.

I would also like to thank all the members of the Computer Graphics Group for their helpful comments throughout my program. The group members includes Cheng Lei, Jason Gedge, Zhao Pei, Lai Kang, Ding Liu, Samaneh Eskandari, Timothy Yau, Rouzbeh Maani, Yi Gu, Xiaoqiang Zhang, Jia Lin and Yiming Qian.

I truly appreciate the team from Ocean Networks Canada at the University of Victoria, who helped with the deployment and the maintenance of the underwater camera system. In particular, I would like to thank Paul Macoun who coordinated my on-site visits and helped me to access the on-site equipment.

I received financial support from the Natural Sciences and Engineering Research

Council of Canada, Alberta Innovates Technology Futures, Canada Foundation for Innovation, Department of Computing Science, and the Faculty of Graduate Studies and Research.

I thank my parents for their unconditional love and support. Last but not least, I would also like to thank my wife, Xiaohui Sun, for her endless support and understanding for my research work.

Contents

Abstract ii

Preface iv

Acknowledgements v

List of Tables xi

List of Figures xii

1 Introduction 1

1.1 Motivation 1

1.2 Background 3

1.3 Thesis Organization 7

2 Related Work 9

2.1 Decoding 9

2.1.1 Temporal Coding 10

2.1.2 Spatial Coding 18

2.1.3 Other Coding Methods 26
2.2 Calibration 27
2.3 Undersea Imaging Systems 37

3 Single Gaussian Blurred Pattern **40**
3.1 Gaussian Blurred De Bruijn Pattern 41
3.1.1 Establishing Correspondences 47
3.2 Experimental Results 49
3.2.1 Simulated Datasets 50
3.2.2 Real-World Datasets 54
3.3 Discussion 57

4 Scene Adaptive Patterns **60**
4.1 Short-range Effects 61
4.1.1 Image Formation Model 63
4.1.2 Minimum Stripe Width Determination 64
4.2 Long-range Effects 68
4.2.1 Unshifted Images 68
4.2.2 Shifted Images 70
4.3 Implementation Details 74
4.4 Experimental Results 74

5 Camera Housing Calibration **86**
5.1 Stereo Camera Housing Calibration 88

5.1.1 Given refractive normal, estimate layer thickness 90
5.1.2 Search space 91
5.1.3 Implementation Details 93
5.1.4 Experimental Results 94
5.1.4.1 Simulations 94
5.1.4.2 Simulations with Outliers 99
5.1.4.3 Real Data 101
5.1.5 Limitation 106
5.2 Single Camera Housing Calibration 108
5.2.1 Normal Computation 109
5.2.2 Interface Distance Computation 110
5.2.3 3D Reconstruction 120
5.2.4 Implementation Details 120
5.2.5 Experimental Results 122
5.2.5.1 Simulations 122
5.2.5.2 Importance of Calibration 124
5.2.5.3 Real Data 125
5.2.6 Discussion and Limitation 131

6 Undersea Camera System **135**

6.1 Image Capture Module 136
6.2 Control Module 138
6.2.1 Housing 140

6.3 Software Development 142
6.3.1 System Calibration 142
6.3.2 3D Reconstruction 143
6.4 Results 143

7 Conclusion and Future Work **147**
7.1 Contributions 147
7.2 Future Work 149

Bibliography **150**

List of Tables

2.1 Categorization of most existing structured light methods. 10

3.1 Evaluation of two methods with two types of patterns. 59

4.1 The 3D error ΔX for all the datasets in Figure 4.11. The unit for the numbers is mm. 85

5.1 Comparison between the estimated parameters by the new method and the ground truth. "GT" denotes the ground truth. 107

5.2 Evaluation of the new method against the ground truth. 131

5.3 Comparison of different methods. 132

List of Figures

1.1 (a) Configuration of a typical structured light system (From Wikipedia). (b) Triangulation with established correspondence. See text for details. 6

2.1 (a) The recovered phase with ambiguity. (b) The unwrapped phase without ambiguity. 16

2.2 A sample De Bruijn pattern (Image from [92]). See text for details. . 20

2.3 Errors could exist when the monotonicity assumption is violated. See text for details. 22

2.4 A sample grid pattern that is designed based on pseudo-random sequence (Image adapted from [77]). 23

2.5 (a) A color coded grid pattern (Image from [68]). (b) A 2D color dot pattern designed based on pseudo-random 2D array (Image from [22]). 24

2.6 Illustration that using radial distortion to approximate the refraction effects would fail. 30

2.7 A camera that is placed in a water-tight housing. 31

2.8 Illustration of the proposed method of Chang and Chen [16]. (Image from [16]). In the 3D case, the y axis is orthogonal to the page. . . . 32

2.9 Illustration of the proposed method in [1]. (Image from [1]) 34

2.10 Illustration of the proposed method in [90]. (Image from [90]) 36

3.1 Stripe pattern used in Zhang *et al.* [92] (Image from [92]). 41

3.2 Blurred pattern after Gaussian blur is applied to the pattern in Figure 3.1. 42

3.3 A sub-region of a captured image in one of the experiments. 48

3.4 Result of simulated experiment. (a) Captured images using Maya. (b) Simulated ground truth. (c) 3D mesh corresponding to (b). (d) Depth map recovered by the new method. (e) 3D mesh by the new method. (f) Close up comparison. 51

3.5 Graphs of accuracy percentage and RMS error in the simulated experiment with varying Gaussian kernel sizes and stripe widths. (a) Accuracy percentage graph. (b) RMS error graph. 53

3.6 Quantitative evaluation. (a) Captured image with a De Bruijn pattern projected onto a wall. (b) Captured image when a grid pattern is projected. (c) Ground truth. (d) Depth map recovered by the new method. 55

3.7 Results of face scene. (a,c) Two images with different facial expression and defocus level. (b,d) Depth maps recovered by the new method for (a) and (c), respectively. 56

3.8 Results of cup and mug scene. (a,c) Captured images of cup and mug scene. (b,d) Depth maps recovered by the new method for (a) and (c), respectively. 57

4.1 Pipeline of the presented method. 61

4.2 (a) A captured image under the illumination of a black/white pattern. (b) Image with its inverse pattern. (c) Decoded bit for (a). 62

4.3 The patterns used in the new method. 62

4.4 (a, c, d) $I(2)$, $I(6)$, $I(9)$ and selected 1D windows, respectively. (b) Decoded bit for $I(2)$. (e) Intensity curves of the 1D windows. (f) Depth obtained by the new method and by using gray code patterns, for the red dotted line in (a). 65

4.5 The newly designed patterns. 67

4.6 The properties embedded in binary code patterns. See text for details. 69

4.7 (a, b) Captured images with $P(1)$ and its inverse pattern. (c) Decoded bit for 4.7(a). (d) The regions colored pure green are error regions detected by the new method. (e) Pattern mask. The black regions are correspondences from image pixels that are in the no-error regions. In the next iteration, this mask is applied to all the projected patterns. (f) Newly decoded bit for 4.7(a). Compare to 4.7(c), decoded bit are changed in the error regions only. (g) Error regions detected in 4.7(f). (h) Final decoded bit where there is no error region detected. 72

4.8 Experimental results for the "Bowl Scene". Each row shows the results of a dataset. (a) One of captured images. (b-d) Final correspondence map by the new method, [35] and [61], respectively. Red and green are used for the x and y coordinates, respectively. 77
4.9 Comparing details of correspondence maps in Figure 4.8. 78
4.10 (a) Comparison of the error (in percent) among the new method, [61] and [35]. The x axis is the row index in Figure 4.8. (b) $e(t_{5:9})$ is obtained by the new method described in Section 4.1. 80
4.11 Comparison of results produced by different method: the new method, conventional gray code, [35] and [61]. 81
4.12 Comparing details of correspondence maps in Figure 4.11. 82
4.13 Comparison of 3D models recovered from correspondence maps produced by different method: the new method (left column), [35] (middle column) and [61] (right column). 83

5.1 Demonstration that the typical triangulation method does not work for an underwater camera system. 87
5.2 (a) Flat refractive geometry with multi-layer of refractive interface for a stereo camera system. (b) 2D diagram for (a). 89
5.3 The search space for the normal of the refractive interfaces. 92
5.4 Experimental results for **Case 1.** 96
5.5 Experimental results for **Case 2.** 96
5.6 Experimental results for **Case 3.** 97

5.7 Experimental results for **Case 4.** 98

5.8 Experimental results for **Case 5.** 99

5.9 Experimental results for **Case 6.** 100

5.10 Experimental results for **Case 2** with ourliers in the input. 101

5.11 Setup for the real experiments. 102

5.12 Results of the "Plane" scene. From left to right: captured image, 3D reconstruction results for the case when refraction is not accommodated, results of **Case 5** and **6** using the presented method. The two rows shows the 3D reconstruction results in two different views. The ground truth is shown in red for comparison. 103

5.13 Top row: results of the "Cave" dataset. Bottom row: results of the "Boat" dataset. The ground truth is shown in red for comparison. . . 105

5.14 Refractive model. 108

5.15 Computing the interface distance d_0. See text for details. 110

5.16 Diagram for derivation of Eq. 5.18. See text for more details. 115

5.17 Extension for Eq. 5.18. 117

5.18 Plot of d_{12}, d_{13}, d_{23} when the observed pixel is (a) Noise free (b). Corrupted by Gaussian noise with variance $\sigma = 0.5$ pixels. 121

5.19 Results for the simulated experiments. (a, b) Angular error before and after nonlinear refinement. (c, d) Normalized error for d_0 before and after nonlinear refinement. 123

5.20 Impact of the housing parameters to the final 3D reconstruction results. See text for details. 125
5.21 Setup for the real experiments. 126
5.22 The dispersion effect observed by (a) Canon 6D camera and (b) Blackfly camera. 127
5.23 Experimental results using the Canon 6D camera. Left: Images captured under ambient light. Middle two: Reconstructed 3D point cloud from two viewpoints. Right: Comparison of 3D point cloud using the ground truth (red) and the estimated parameters(green) by the new method. 128
5.24 Experimental results using the Point Grey Blackfly camera. Left: Images captured under ambient light. Middle two: Reconstructed 3D point cloud from two viewpoints. Right: Comparison of 3D point cloud using the ground truth (red) and the estimated parameters(green) by the new method. 129
5.25 Demonstration of the limitation of the presented method. 134

6.1 Diagram of the image capture module. 137
6.2 Diagram of the control module. 138
6.3 Left: A camera and its housing. Right: A camera, a projector and their housings. 140
6.4 The system before its final assembly. 141
6.5 The system before it is deployed undersea. 142

6.6 3D reconstruction results of a coral reef in the lab: front view (left) and top view (right). 143
6.7 3D reconstruction results using PMVS2 [27]. Left: front view. Right: Top view. 144
6.8 Images captured by two different cameras. 145
6.9 Top row: 3D reconstruction result by the structured light method. Bottom row: result by using the method in [27]. 146

Chapter 1

Introduction

1.1 Motivation

Three dimensional (3D) surface reconstruction refers to the process of extracting three dimensional geometry of an object from 2D images. It is an important research topic in computer vision because it has many applications in different fields. For example, in digital heritage [54], 3D models of cultural art works are built. The preserved digital 3D models help people remember history, and re-tell the story of an artifact even if it is damaged due to the war or other reasons. Other applications include industrial inspection of manufactured parts [88] and object recognition [76]. Many methods have been developed to obtain the 3D geometry of an object and among them, stereo methods are most commonly used. Stereo methods capture images of the scene from two or more viewpoints, and then correspondences of scene points among different images are used to compute the 3D positions of the scene points

using a procedure called triangulation.

Stereo methods can be categorized into two groups: *passive* and *active* stereo. Passive stereo methods rely on the captured images only and do not require *a priori* information. The research in passive stereo methods has reached a new era since the launch of a publicly available performance evaluation website [41], which allows researchers to compare their methods with the state-of-the-art ones using a set of standard stereo datasets. The major limitations of passive stereo methods include the difficulties of handling occlusions and finding correspondences in textureless regions in the scene. In contrast, active stereo methods such as laser scanning and structured light project illumination patterns into the scene to create identifiable features, and hence minimize the difficulty of establishing correspondences. Because of its accuracy, the structured light method [80] is often used to construct the ground truth of the depth maps used for evaluating passive stereo methods [41]. However, developing robust and practical structured light methods remains an active research area and is still an open problem in computer vision.

Most of the existing structured light methods are proposed for a land-based camera system, and cannot be applied directly to an underwater camera system. The biggest challenge is in calibrating the cameras, which is an essential step for 3D reconstruction. Even though the system may be correctly calibrated on land, the strong water turbulence could potentially change the positions and orientations of the cameras and projectors. As well, routine maintenance such as cleaning the lenses by divers may also alter the calibration parameters. Such changes require re-calibrating

the entire system, which is costly and impractical when the system has been deployed permanently undersea. This challenge usually does not exist for a typical land-based structured light system. Additionally, in a typical underwater camera system, the camera is placed inside a watertight housing, and views the scene through a flat piece of glass. As a result, the light that travels into the camera undergoes two refractions and its path is not a straight line, the result of which causes distortion in the captured images. The distortion depend on the scene depth and cannot be simply modeled as lens radial distortion. To be able to compute refraction correctly requires estimating more parameters of the underwater camera system than a similar land-based camera system. Therefore, calibrating underwater cameras remains a challenging problem in computer vision.

1.2 Background

Many structured light methods have been proposed in the past [2, 3, 4, 15, 21, 22]. Some of them can achieve real-time performance [50, 74], and some build remarkably accurate 3D models [80]. However, there are two major limitations in existing structured light methods. First, these methods either project multiple patterns onto the object and utilize temporal information to establish correspondences, or project a single pattern and use neighboring information to establish correspondences. The former ones typically produces dense and more accurate results compared to the latter ones, but they cannot be applied to dynamic scenes because they require to project multiple patterns. The latter ones can be applied to dynamic scenes since only one

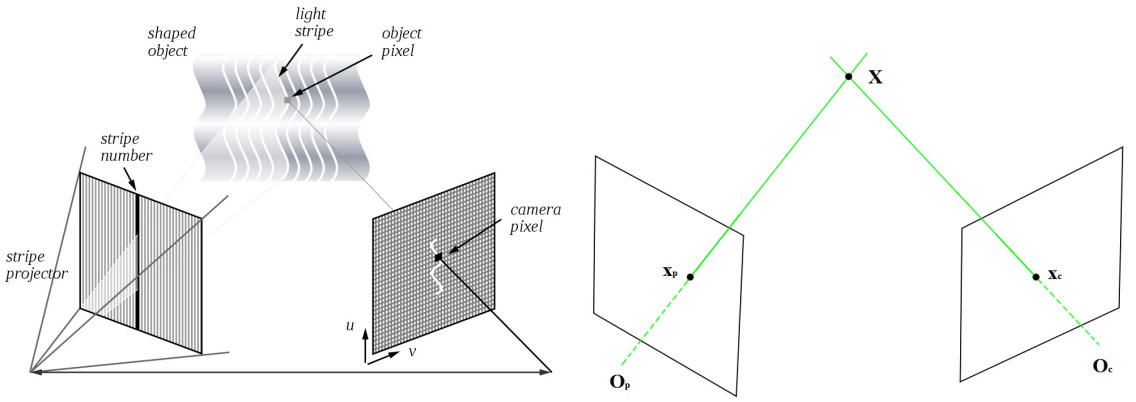
pattern is required. However, the established correspondences are usually sparse. Motivated by this limitation, a new structured light pattern, which can be applied to dynamic scenes and produce dense correspondences, is designed in this thesis.

The second limitation is that most state-of-the-art structured light methods have a fundamental assumption that every scene point receives more direct illumination from the light source than from indirect illumination. The indirect illumination is usually called global illumination which includes inter-reflection, subsurface scattering and projector defocus. Nevertheless, this assumption can be violated. An example is to perform structured light 3D scanning of the inside of a bowl where global illumination in the form of inter-reflection can be the dominant source for many scene points inside the bowl. Under this circumstance, the correctness of most structured light methods cannot be guaranteed, in which case large errors may be present in the reconstructed model. Moreover, projector defocus is also a major source of errors for most structured light methods. When the projected pattern is defocused, the projected identifiable features may not be correctly identified and results in errors in establishing correspondences. Various methods have been developed to handle scenes in the presence of either global illumination or projector defocus effects [35, 37, 39, 89, 94]. Even the few that can handle both of them require some restricted assumptions. For example, the method developed in [35, 61] assumes that patterns with stripe width larger than or equal to 8 pixels are not blurred. Furthermore, it is assumed that there is only one kind of global illumination effect at each scene point. To address these limitations, in this thesis, a novel method that handles both global

illumination and projector defocus is presented.

Once the correspondences are established, the 3D points can be computed by triangulation, which can only be accurately performed provided the cameras are calibrated. In camera calibration, the parameters, which include the focal length of the camera, the lens distortion parameters, and the principal point of the optical axis, are determined. In a stereo camera system, the additional information of the relative position between the two cameras are required. Figure 1.1(a) shows the setup of a typical structured light system and Figure 1.1(b) the principle of triangulating the 3D position once the correspondence between the camera and the projector is established. In Figure 1.1(b), x_p and x_c are the established correspondence between the projector and the camera. O_p and O_c are the centers of projection for the projector and the camera, respectively. X is the triangulated 3D position. In this thesis, the focus is on structured light systems using 2 cameras instead of one because such systems offer better accuracy and can avoid the problem of mismatched resolution and color between the camera and the projector in a single camera system. As well, there is no need to calibrate the projector, which is used to project patterns only.

Despite the simplicity of triangulation in air, it is much more difficult in underwater because of refraction. To account for the refraction effects, the camera housing parameters need to be estimated in addition to the above described parameters. Existing underwater camera calibration methods usually have certain limitations. For example, some require using a checkerboard pattern [1, 90], which is very difficult to use when the camera system is deployed undersea. Some require the refractive inter-



(a) (b)

Figure 1.1: (a) Configuration of a typical structured light system (From Wikipedia). (b) Triangulation with established correspondence. See text for details.

face to be parallel to the image plane [48], which may not be practical. In this thesis, two novel underwater camera calibration methods that do not require any calibration target such as a checkerboard pattern are presented.

There are many existing camera systems that are deployed undersea for studying the underwater species and their habitat. Researchers from different discipline study them for different purposes. For example, marine biologists try to understand the marine environment which helps to support life on earth [78, 5]. Studying underwater species helps them understand the effects of climate change, pollution, and invasive species and so on. The environmentalists may require the information on water turbidity [57]. Other information such as the change in the population of underwater species in a certain period of time can be useful to researchers studying underwater ecological systems [73]. The major limitation of the existing undersea camera system

is that only 2D images are provided. To address this problem, a new multi-camera multi-projector undersea system with the capability of performing accurate 3D surface reconstruction is presented in this thesis.

1.3 Thesis Organization

The organization of this thesis is as follows. Chapter 2 provides an overview of structured light systems and different methods of underwater camera calibration. This chapter gives the introduction and categorization to the existing structured light methods, as well as their limitations. The existing underwater camera calibration methods and their limitations are also discussed.

Chapter 3 presents a new structured light method that recovers a dense depth map using a single color pattern. Compared to the existing methods, the advantages of the new method include: 1) it can establish dense correspondences and 2) it can be applied to dynamic scenes. Part of this chapter is published as:

- X. Chen and Y.H. Yang, Recovering Stereo Depth Maps using a Single Gaussian Blurred Structured Light Pattern. *Canadian Conference on Computer and Robot Vision*, May 28-31, 2013, pp. 295-302.

In Chapter 4, a novel structured light method is presented which can handle global illuminations such subsurface scattering, inter-reflection, as well as projector defocus. Compared to the state-of-the-art methods, this method can handle more severe projector defocus. As well, it detects and corrects errors caused by inter-

reflection iteratively. This method has been published in:

- X. Chen and Y.H. Yang, Scene Adaptive Structured Light using Error Detection and Correction. *Pattern Recognition*, Vol. 48, Issue 1, 2015, pp. 220-230. (http://dx.doi.org/10.1016/j.patcog.2014.07.014)

Chapter 5 presents two novel methods for underwater calibration, as well as a new method that performs 3D reconstruction using a single camera. Compared to the methods in literature, this method does not make any assumption about the camera configuration, such as the refractive interface needs to be parallel to the image plane. It requires no calibration target such as a checkerboard pattern. One of the methods is submitted to the International Conference on Computer Vision (2015) and is currently under review. The other method has been published in:

- X. Chen and Y.H. Yang, Two-view Camera Housing Parameters Calibration for Multi-Layer Flat Refractive Interface. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 24-27, 2014, Columbus, Ohio.

Chapter 6 presents the design and development of a new multi-camera multi-projector undersea system, which is able to provide 3D reconstruction to the undersea habitat. This system has been deployed in Saanich Inlet, B.C., where it monitors the undersea habitat, and uploads the collected data, making it accessible to researchers around the world. This chapter will be submitted to a journal or a conference for publication.

Finally, conclusion and future work are described in Chapter 7.

Chapter 2

Related Work

A typical structured light method consists of two major components which include decoding and calibration.

❶ Decoding is the most important component of a structured light system. It takes image(s) captured by the camera as input, and establishes pixel correspondences between the captured image(s) and the projected pattern as appears in the projector.

❷ Calibration obtains the camera parameters which are necessary for triangulation. After triangulation, a 3D point cloud can be obtained from the correspondences that are established from the decoding step.

2.1 Decoding

Structured light methods project one or more patterns to the scene in order to create identifiable features. Most of the existing methods can be categorized into two

Table 2.1: Categorization of most existing structured light methods.

Temporal Coding	Binary Code [44, 71, 85] Gray Code [3, 4, 8, 30, 34, 35, 61, 74, 87, 89] N-ary Code [15] Phase Shifting [42, 43, 97, 98, 99]
Spatial Coding	De Bruijn Stripe Patterns [66, 67, 92] Pseudo-random Grid Patterns [2, 32, 56, 59, 68]
Others	Space-time Stereo [21, 93, 95] Viewpoint Coding [91]

groups based on the number of required projected patterns (multiple-shot or single-shot). The advantage of multiple-shot methods is that they usually produce a higher accuracy than those using only single shot. However, single-shot methods can be applied to dynamically deformed or moving objects while most of the multiple-shot methods cannot. Table 2.1 summarizes the categories of most existing structured light methods.

2.1.1 Temporal Coding

Multiple-shot methods are usually called temporal coding methods. It is one of the earliest and most commonly used structured light methods. A series of patterns are projected onto the surface of an object, and each pixel is coded by a sequence of illumination values by temporally varying the projected pattern.

Binary code patterns [71, 85] include only black and white stripes. By applying this set of patterns, each point can be assigned to a unique code (e.g. black is 0 and white is 1). In general, N binary code patterns can encode 2^N stripes. Binary coding methods are reliable and not too sensitive to the object surface characteristics since they do not rely on any object's color information. Therefore, binary code patterns are widely used in modern structured light systems to achieve high speed and high accuracy reconstruction [44]. However, a disadvantage of using binary code patterns is that the stripe width of the pattern with the highest spatial frequency is only one pixel in order to uniquely encode each pixel, and this pattern can be easily blurred when projected. Therefore, it can be hard to distinguish a black pixel from a white pixel when this pattern is projected.

Gray code patterns [8, 34] are very similar to binary code patterns in which every pixel is uniquely coded as well. In this set of patterns, the stripe width of the pattern with the highest spatial frequency is two pixels. Recently, gray code patterns have been combined with other techniques such as photometric processing [3, 4]. There are two main advantages of incorporating photometric information into a structured light system, which can recover more surface details and self-calibrate the cameras and the projectors. By incorporating a photo-geometric optimization stage, both geometric and photometric data are used for optimization so that the final 3D surface is the best fit for both sets of data. With the optimization step, the results show significant improvements in both accuracy and resolution in the reconstructed 3D models. However, this method is computationally intensive. In one of the experiments

using 4 cameras and 2 projectors, the photo-geometric optimization step implemented using C/C++ takes over 30 minutes using a computer with 3.2 GHz CPU and 2GB memory.

The gray code patterns have been applied to challenging scenes such as the ones with inter-reflections [89]. Inter-reflections exist when reflecting items such as a ceramic bowl or a piece of metal is placed in the scene. It is challenging because the decoding can be wrong. For example, a pixel that corresponds to a scene point illuminated by a black stripe might be brighter than when the same scene point illuminated by a white stripe because of inter-reflection from other parts of the scene. A robust categorization method is proposed in [89] to label a camera pixel as certain or uncertain. For the camera pixels that are labeled certain, their corresponding pixels in the projected patterns are obtained. Then, new patterns are designed so that the corresponding pixels in the projected patterns are set to be black so that there is no illumination coming from them. In this case, the total amount of illumination from the projector is reduced since some pixels are set to be black. After that, the new patterns are projected iteratively until every camera pixel is labeled as certain. The results shown in [89] indicate that there are still errors exist. Moreover, this method is not designed to account for any projector defocus. Also, this method requires a large number of iterations (10-20) and results in a large number of captured images (400-800). Last but not least, the method applies a pre-processing technique [62] which is to separate the direct and global components of the illumination. This step is quite complex, which could fail when the global illumination is too strong. Besides inter-

reflection, the method proposed in [35] handles another kind of global illumination effects which is sub-surface scattering, and projector defocus as well. This method uses four sets of patterns including gray code patterns in order to label a pixel as certain or uncertain. The error correction step is also an iterative process which is similar to that of [89]. A basic assumption of this method is that there is only one kind of effect at each scene point. Therefore, it cannot handle the scenario where multiple global illumination effects are present at the same scene point. Furthermore, the method handles projector defocus only to a limited extent and large errors still exist when the projector defocus is severe.

Gray code patterns can be applied to achieve high accuracy and high resolution. A multi-camera multi-projector system is designed using gray code patterns in the work of [87]. High Dynamic Range (HDR) imaging technique is applied to handle challenging scenes such as the ones with shinny objects. A super resolution method is incorporated so that the reconstructed models have a higher resolution. Although accurate results have been obtained, this method requires a massive number of images since multiple cameras and multiple exposure settings are used to handle specular highlights. The number of captured images is 172,140 in one of their experiments. On the other hand, high efficiency performance is made possible when modification is applied to systems using gray code patterns [74]. Different kinds of gray code patterns have been designed, where the number of patterns is reduced to four [74] and eight [30] in total. Under this circumstance, the method can reconstruct 3D models efficiently with a high speed camera and projector.

When the binary and gray code patterns are applied, it may not be simple to obtain the correct code for a pixel because of the difference of surface albedo of a scene. A scene point that is lit by a white stripe may appear darker than another scene point that is lit by a black stripe, in which case, using simple thresholding can result in the wrong code. To solve this problem, both the patterns and their inverse patterns are usually projected. The decoding for each pixel for each pattern is then performed by comparing its brightness between the pattern and its inverse pattern. Using this approach, the number of patterns required is doubled.

N-ary code patterns [15] can effectively reduce the number of required patterns. It is noteworthy that the patterns described above encode each pixel in two possibilities only, i.e. black or white. Therefore, the intensity or color information is never exploited. Caspi *et al.* [15] propose a technique to code patterns in the RGB color space. The design of N-ary patterns depends on some parameters described as follows: the number of colors to be used (N), the number of projected patterns (M) and the noise immunity factor (α). The method proposes a reflectivity model which is similar to a hash function. It transforms the code for each projector pixel into a set of projected intensity in each color channel. Comparing to binary and gray code scheme, the advantage of N-ary code is that the number of required patterns is reduced significantly. In particular, when M patterns are used with N colors, N^M stripes can be uniquely coded using the N-ary coding strategy. However, only 2^M stripes can be uniquely coded using either binary code or gray code patterns. The disadvantage of N-ary code is that it is sensitive to the object's color since the pat-

terns are designed in the RGB space. Therefore, it is less robust comparing with the binary and gray codes.

Phase shifting is a set of well-known temporal coding methods for object surface measuring. This kind of techniques takes advantage of the gray level resolution of modern projectors by projecting a set of sinusoidal patterns. The patterns are usually designed by the following equation

$$I(x,y,t) = I'(x,y) + I''(x,y)\cos\big(\phi(x,y) + \delta(t)\big). \tag{2.1}$$

In Eq. 2.1, $\delta(t)$ is the time-varying phase-shift angle which is pre-defined when designing the patterns. $\phi(x,y)$ is the unknown phase related to this temporal phase shift. $I'(x,y)$ is the intensity-bias and $I''(x,y)$ the modulation signal amplitude. $I(x,y,t)$ is the intensity in the designed patterns.

There are three unknowns in Eq. (2.1). Therefore, the minimum number of required patterns is three. In this case, the phase-shift angle $\delta(t)$ is normally set to be $2\pi/3$. Hence, Eq. (2.1) can be re-written as three equations shown in Eq. (2.2)

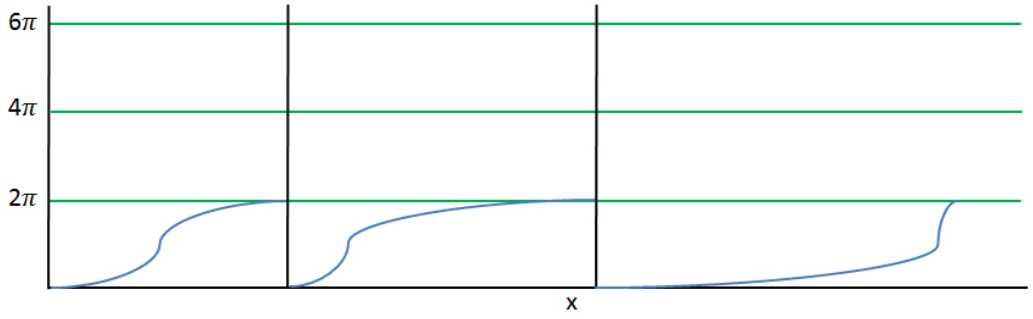
$$\begin{aligned} I_0(x,y) &= I'(x,y) + I''(x,y)\cos\big(\phi(x,y) - 2\pi/3\big) \\ I_1(x,y) &= I'(x,y) + I''(x,y)\cos\big(\phi(x,y)\big) \\ I_2(x,y) &= I'(x,y) + I''(x,y)\cos\big(\phi(x,y) + 2\pi/3\big). \end{aligned} \tag{2.2}$$

According to Eq. (2.2), the unknown variable $\phi(x,y)$ can be calculated by Eq. (2.3)

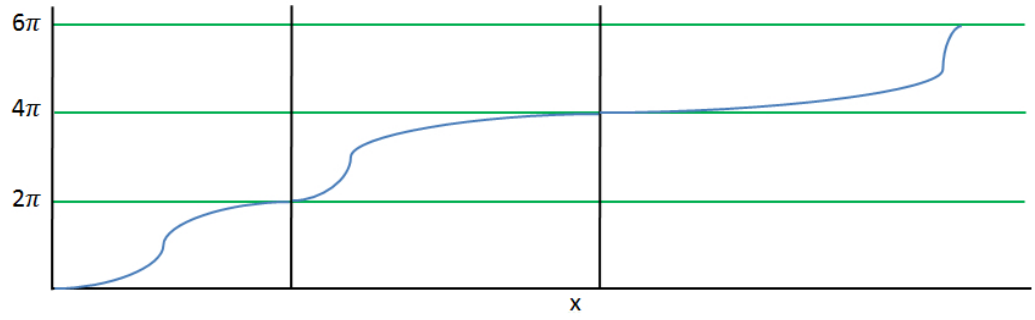
$$\phi(x,y) = \tan^{-1}\left(\sqrt{3}\frac{I_0(x,y) - I_2(x,y)}{2I_1(x,y) - I_0(x,y) - I_2(x,y)}\right). \tag{2.3}$$

When applying a phase shifting method, the projection period is often designed to be $2k\pi$ in order to achieve high accuracy. Therefore, the recovered phase $\phi(x,y)$

has an ambiguity of $2k\pi$, where k is an integer representing the projection period. In particular, there are discontinuities in the arc tangent function every time when ϕ changes by 2π. Figure 2.1(a) shows an example of the recovered phase where ambiguity exists. To solve this problem, phase unwrapping is normally applied to convert the wrapped phase to the absolute phase. Figure 2.1(b) shows the absolute phase by applying phase unwrapping to the wrapped phase in 2.1(a).



(a)



(b)

Figure 2.1: (a) The recovered phase with ambiguity. (b) The unwrapped phase without ambiguity.

Phase shifting has been applied in [39] to address the problem of inter-reflection

present in the scene. The method uses a sliding projector and recover depth maps by analyzing the intensity profile of each pixel in the frequency domain. The assumption of this method is that direct illumination is the dominant source for every scene point which may not hold in general. Furthermore, it introduces the overhead of translating the projector at a constant velocity which is also not very practical. Last but not least, the method is unable to handle severe projector defocus. The structured light method in [61] applies phase shifting patterns to reconstruct scenes including regions that have specular highlights or are in shadow. It places a diffuser in front of the light source so that the effects of specular highlights and of the shadow can be reduced. Although a diffuser could potentially blur the projected patterns, the method takes advantage of the observation that most structured light patterns are designed to be either vertical or horizontal. Using a vertical pattern as an example, each pixel has the same color comparing to all the other pixels in the same vertical line. The method is carefully designed and the diffuser is placed so that the light is scattered only along the line that has the same color, in which case, the pattern is preserved after going through the diffuser. The diffuser is a lenticular array that is 12" $\times$ 12" in size. The method can perform 3D reconstruction for regions that are in shadow or have specular highlights. However, this method cannot handle global illumination.

There are some advantages of phase shifting methods when compared to using the gray code patterns. First, phase shifting methods [42, 43, 99] can achieve high accuracy comparable to the methods using the gray code patterns. Second, since the number of required patterns is reduced to a minimum of three, high efficiency

can be achieved when the camera and the projector are synchronized [97, 98]. In particular, a modern projector can project 60 frames per second, which means that if a camera can capture 60 images per second when synchronized with the projector, then three images are captured in 1/20th of a second. With a proper design of the phase shifting algorithm, a structured light system has the possibility of achieving an efficiency of 20fps (frames per second). Last but not least, phase shifting methods can be adapted to capturing semi-dynamic scenes. Since the image capturing time is only 1/20 seconds, and under the assumption that the motion of the objects in a scene is small within this time, the phase shifting method can be applied. By applying certain motion compensation techniques [96], phase shifting methods can provide 3D reconstruction of dynamic objects with acceptable error. Despite the above advantages, there are some disadvantages of phase shifting methods. The major disadvantage is that the phase unwrapping methods can be sensitive to noise, and hence complex algorithms are required [29]. Furthermore, phase shifting methods are more sensitive to projector defocus compared to methods using gray code patterns.

2.1.2 Spatial Coding

Temporal coding techniques establish correspondences by the identifiable features created temporally, in particular, by capturing a set of images. On the contrary, spatial coding methods create identifiable features within one single pattern. This is achieved by utilizing neighborhood information in the pattern. For example, every 3 $\times$ 3 window can be designed to appear only once in the pattern. With this kind of

design, correspondences can be established using only one pattern.

Although temporal coding methods usually achieve high accuracy in the reconstructed 3D models, and can be applicable to dynamic scenes in certain circumstances, they are rarely applied to dynamic scenes in practice. The reason is that multiple shots are required and motion exist among different captured images, and hence error can exist in the code due to the motion. Even with synchronized camera and projector where each image is captured in 1/60 seconds and the number of captured images is only three [97, 99], the result is inaccurate due to the motion that cannot be neglected within $\frac{1}{60} \times 3 = \frac{1}{20}$ seconds. Therefore, complex techniques to compensate the motion between captured images are required [99]. To sum up, a structured light system using temporal coding is normally applied to static scenes. Instead, spatial coding methods which require only one shot are used to handle dynamic scenes.

Kinect [40] is one of the most popular commercial products that is able to produce depth map. It projects one infrared structured light pattern onto the object and recovers the depth map in real-time. The infrared pattern is used such that the method is less sensitive to the object surface color. Kinect is originally designed for interactive gaming instead of recovering depth maps. The reason is that the recovered depth map is not accurate. In particular, it generates sparse correspondences using the single infrared pattern, and post processing techniques are used to recover dense depth map.

The De Bruijn pattern, which is designed based on the De Bruijn sequence [26], is one of the most commonly used spatial coding patterns. A De Bruijn sequence can be

generated using a publicly available generator [75]. Using the De Bruijn sequence, De Bruijn structured light patterns can be generated. Due to the *windowed uniqueness* property of a De Bruijn pattern, global optimization such as dynamic programming can be applied to establish correspondences. A sample De Bruijn pattern is shown in Figure 2.2. This pattern consists of 125 stripes.

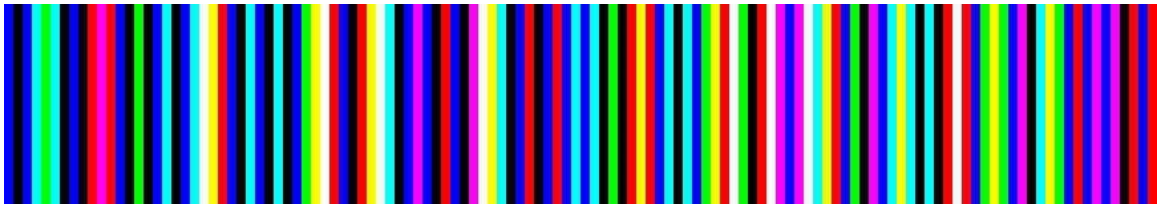


Figure 2.2: A sample De Bruijn pattern (Image from [92]). See text for details.

Various methods based on the De Bruijn patterns have been proposed [66, 67, 92]. Although they are different from each other, the general procedure is similar. First, most methods use stripe patterns such as the one shown in Figure 2.2. The pattern is projected onto the scene and the camera captures the image. Second, these methods usually require an image processing step to extract the stripe edges in the captured image. Finally, a global optimization method is applied to match the extracted stripe edges in the captured image and in the projected pattern. Once a set of matches are obtained, triangulation can be applied to reconstruct the object surface. The differences among these methods are normally in three aspects: patterns, image processing techniques and global optimization methods. For example, Zhang *et al.* [92] apply the De Bruijn pattern shown in Figure 2.2. A different pattern is

designed in the work of Pages *et al.* [66, 67]. The pattern is a classic stripe pattern with *windowed uniqueness* property in the RGB space. Furthermore, the intensity channel of this pattern is a multi-slit pattern which has a similar property as the *windowed uniqueness* property.

In a typical structured light system using a De Bruijn stripe pattern, an image processing technique is usually required to detect the edges of the stripes. For example, the second order derivative of the captured image is obtained and the stripe edges are located at the local maxima of the second order derivative [67]. The edges can be detected to sub-pixel accuracy. A more complicated stripe color classification algorithm is proposed by Fechteler *et al.* [23, 24]. The method first applies first-order derivative to the captured image to locate candidate stripe edges. Then, the location of each stripe is refined using a color classification technique.

After the edges of the stripes are detected in both the captured images and the projected pattern, global optimization is applied to obtain the correspondences. Dynamic programming is frequently employed in this step because it is fast and can be easily parallelized. However, since monotonicity is an assumption for dynamic programming, the produced results generally contain some errors when this assumption is violated such as the scenario indicated in Figure 2.3. This figure shows that the projector indices $P_i < P_{i+1}$. However, the camera indices $C_i > C_{i+1}$ indicate that monotonicity does not hold anymore. To solve this problem, a multi-pass dynamic programming procedure is proposed in the work of Zhang *et al.* [92].

Besides the above described stripe patterns, grid patterns are commonly applied

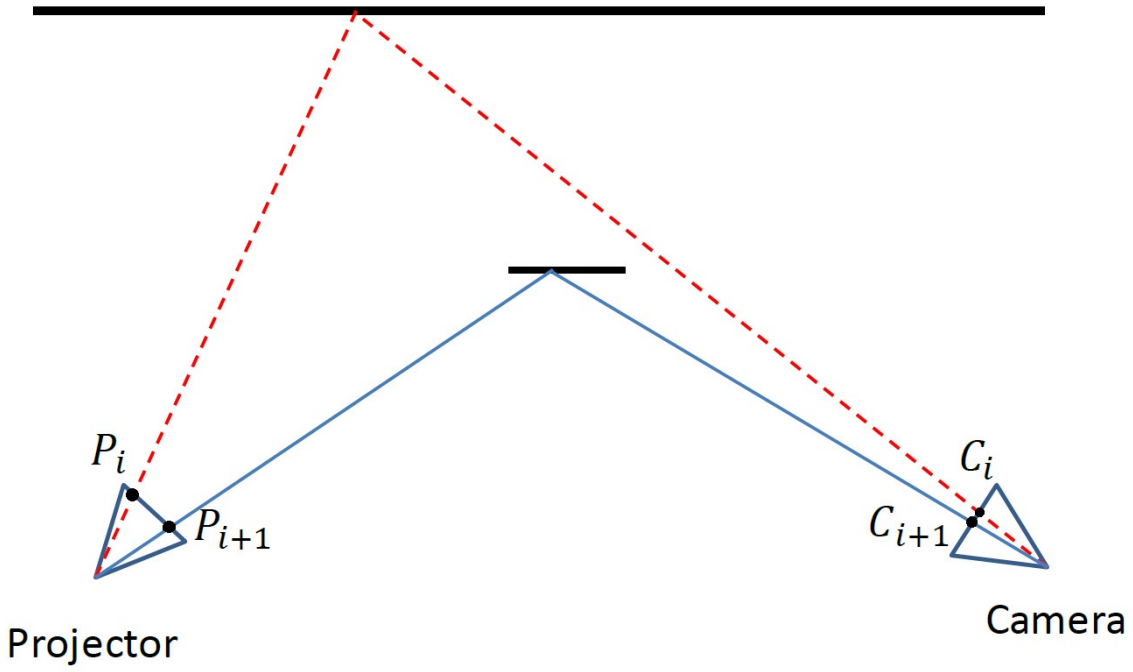


Figure 2.3: Errors could exist when the monotonicity assumption is violated. See text for details.

as well. The grid patterns are also designed based on a De Bruijn sequence. To be more specific, every sub-window is uniquely labeled in a 2D grid pattern so that its position is identifiable. Several representative grid patterns are described as follows.

Pseudo-random sequences can be used to generate a grid pattern so that the grid corners can be marked using dots or other primitives. Within this array, any sub-window of a certain size that slides over the entire array is uniquely coded. The known code for a sub-window can be used to locate the position in the array. Various grid patterns have been designed [32, 56, 59]. A sample pattern that is designed based on a pseudo-random sequence is shown in Figure 2.4. Three different primitives are used

in this pattern. Any 3×3 sub-window is unique in this pattern and one sub-window of the designed patterns is shown in the bottom left corner. Once a sub-window is decoded, its location in the pattern can be obtained. Other primitives such as disc, circle and stripe can be used as well [2].

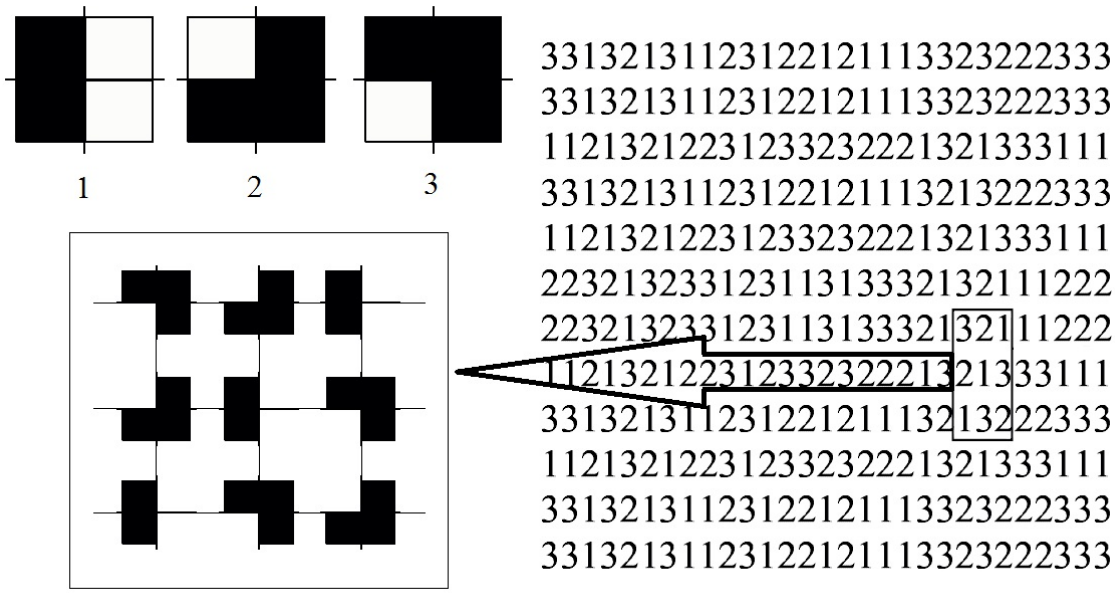
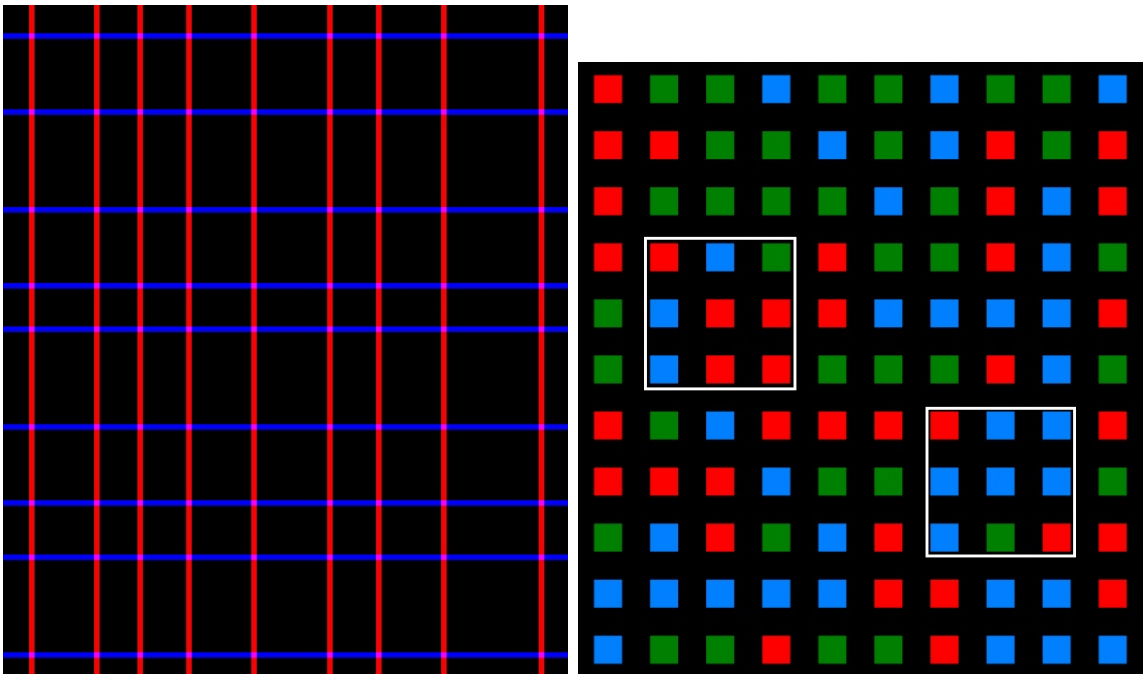


Figure 2.4: A sample grid pattern that is designed based on pseudo-random sequence (Image adapted from [77]).

The pattern in Figure 2.4 consists of only black and white colors. Color coded grid patterns are commonly used, which are an extension of the De Bruijn stripe pattern. The pattern in Figure 2.2 is coded in one direction. By extending the coding strategy to both vertical and horizontal directions and with some other modifications, a 2D grid indexing pattern can be designed [68]. A grid pattern with colors is shown

in Figure 2.5(a). Depending on the applications, the coding strategies for the two directions can be either the same or different.



(a) (b)

Figure 2.5: (a) A color coded grid pattern (Image from [68]). (b) A 2D color dot pattern designed based on pseudo-random 2D array (Image from [22]).

A pseudo-random 2D array has the property that any sub-window within the array is unique. Hence, a brute force algorithm [58, 65] has been proposed to generate a 2D array such that the uniqueness property of any sub-window is preserved. The algorithm can be briefly described as follows. An iterative procedure is performed to add a new code to the pattern and check against all the previously added ones. If the newly added code is different from all the others, then it is accepted; otherwise it

is rejected. The iteration is performed until the entire pattern is created. A pattern designed using this algorithm is shown in Figure 2.5(b) which is applied in [22]. This pattern has the property that every 3×3 array of color dots is unique in the pattern. For example, there are two white rectangles marked in the pattern and each rectangle is a 3×3 sub-window. It is obvious that the codes in these two rectangles are different.

In a typical structured light system, temporal coding methods usually can achieve higher accuracy than spatial coding methods since each projector pixel is uniquely labeled. The accuracy of spatial coding methods can be impacted by the robustness of the image processing step in detecting the stripe edges or the grid corners. However, temporal coding techniques usually do not require such a step. Furthermore, the former can achieve higher resolution since the spatial coding methods establish correspondences along the stripe edges or the grid corners only. Last but not least, temporal coding methods such as gray and binary codes are usually more robust because they are not sensitive to object's color. In contrast, most patterns used in the spatial coding methods are designed in the RGB color space. Since they require only one shot, they are frequently applied to dynamic scenes and some of them can achieve real time performance [50, 51], while typical temporal coding methods are applied to static scene.

2.1.3 Other Coding Methods

Space-time stereo [21, 93] is a technique that combines the advantages of temporal coding and spatial coding methods. A traditional stereo method establishes correspondences between two captured images of a scene at a certain temporal frame. However, space-time stereo matches a pair of video streams. The observation is that the appearance of a real-world scene varies over time, due to lighting changes, motion and shading. Such a variation over time is used as a cue to establish correspondences in this method. In particular, different patterns are projected onto a scene, which can be either dynamic or static, and two space-time windows are defined to compute the matching cost. The experimental results of space-time stereo are significantly better than that of traditional stereo methods. This method has been extended and applied to 3D reconstruction of facial expressions [95]. The accuracy of space-time stereo methods is normally in between the spatial and temporal coding methods.

Young *et al.* [91] develop a theoretical framework to replace time-coded structured light patterns with viewpoint codes. In particular, the method projects only one pattern but uses several cameras to view the scenes from different viewpoints to achieve the same effect as that of using temporal patterns. The method can be applied to dynamic scenes because only one pattern is required. Since cameras are cheap and affordable nowadays, this framework is quite applicable to real-world applications. However, the cameras have to be synchronized in the system, and need to be placed accurately in pre-designed positions.

Unstructured light patterns [19] is presented to account for inter-reflection. It

uses a large number of random patterns such that the code ambiguity is minimized. Shape from caustics [84] is another coding scheme that uses caustics created by the natural sun light in an underwater scene to encode each scene point. However, there is no guarantee that the coding can be unique. As well, the method is appreciable to shallow water environment in the presence of bright sunlight. O'Toole *et al.* [64] present a one-shot structured light method that is able to recover the 3D shape with global illumination. The method uses a high speed projector synchronized with a camera such that the captured image is indeed an integration of many random patterns. Time-of-Flight imaging is a 3D reconstruction technique that is applied in the second generation of Kinect [40]. It uses only one-shot, and the basic principle is use the round trip time of the light from the sensor to the scene and back to measure the distance to the scene.

2.2 Calibration

Geometric calibration for a single camera has been extensively studied over the last 40 years [12, 14, 100]. A camera is normally modeled as a pinhole camera with intrinsic and extrinsic parameters. The intrinsic parameters include the focal length, the principal point, and lens distortion of the camera. The extrinsic parameters include the rotation and translation from the world coordinate system to the camera coordinate system. The goal of geometric calibration is to recover both sets of parameters of the camera.

A commonly used camera calibration method proposed in [100] requires a planar

checkerboard pattern imaged at different positions and orientations. The corners in the checkerboard pattern are either automatically detected or manually selected, and the corresponding points in different images are matched. With a sufficient number of corner points, the intrinsic and extrinsic parameters can be obtained using a closed form solution [100]. Then, the results are refined using maximum likelihood inference. A user-friendly camera calibration toolbox [10] is implemented based on the method proposed in [100].

In a typical underwater system, a camera is placed inside water-tight housing and completely sub-merged in water. An example is shown in Figure 2.7 where the ray from the object points (green points) to the camera center does not go through a straight line because of the refraction at the interface of two different media. In this case, knowing the intrinsic and extrinsic parameters of the camera is not enough. In particular, the housing parameters such as the refractive interface normal and the distance from the camera center to the interface need to be measured. Underwater camera calibration is more difficult compared to land-based calibration due to the refraction effects. Since perspective projection cannot be applied directly, almost all land-based computer vision algorithms cannot be used in the underwater environment. For example, the epipolar geometry does not hold underwater. As well, triangulation is more complex. Moreover, the refraction effects result in scene dependent distortion in the image that cannot be simply modeled as lens radial distortion [70]. Figure 2.6 demonstrates that radial distortion cannot be used to approximate the refraction effects. In particular, this diagram shows that two points P_1 and P_2,

which are at different scene depths, map to the same image pixel x after refraction. In order for radial distortion to correctly model the refraction, x has to be corrected to another image pixel that corresponds to the perspective projection of the actual scene point. However, x could potentially be the image of either P_1 or P_2, and they map to different image pixels under perspective projection. In other words, it is impossible to have a unique solution, which means that the approximation would always fail. Therefore, calibrating an underwater camera system with multiple refractive layers with an unknown refractive axis and layer thickness is still an open problem. Notice that Figure 2.7 demonstrates the configuration of most underwater camera systems [1, 16, 18, 17] where the glass interface is flat. Some other systems use dome housing [52, 63] where the glass is a hemispheric dome. In this case, the camera calibration is even more difficult because the interface normal is different at different position of the refractive interface unless the camera center coincides with the center of the dome, in which case there is no refraction effect. However, a precise alignment is not trivial. Therefore, this thesis focuses on the case where the refractive interface is flat.

The refraction effects are either ignored in the early works of underwater computer vision [72], or approximated, such as using focal length adjustment [25, 47, 53]. However, it is a known fact that the refraction effects are highly non-linear and depend on the scene geometry. Therefore, approximation methods usually produce errors in the results.

Chari and Sturm [17] provide a theoretical analysis of refraction to demonstrate

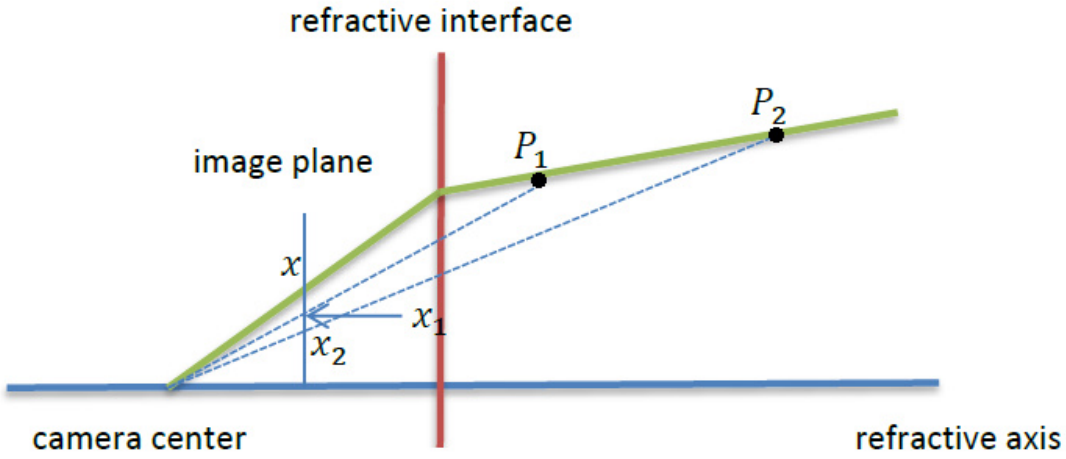


Figure 2.6: Illustration that using radial distortion to approximate the refraction effects would fail.

that the refractive fundamental matrix exists between two cameras viewing a scene through the same refractive interface. By formulating the refraction in terms of quadratic lifted coordinates, a 12×12 refractive fundamental matrix is derived. However, only theoretical results are provided instead of practical implementation. Moreover, there is no follow-up work to implement a practical algorithm based on their idea.

Chang and Chen [16] solve the calibration problem provided some of the parameters are known. The method uses an Inertial Measurement Unit (IMU) that is built in the camera to measure the vertical direction of each view, which is perpendicular to a single refraction interface, i.e., the water surface. With this constraint, the relative pose among the cameras and the 3D object points can be computed. The method can be outlined as follows and the different parameters are shown in Figure 2.8. The

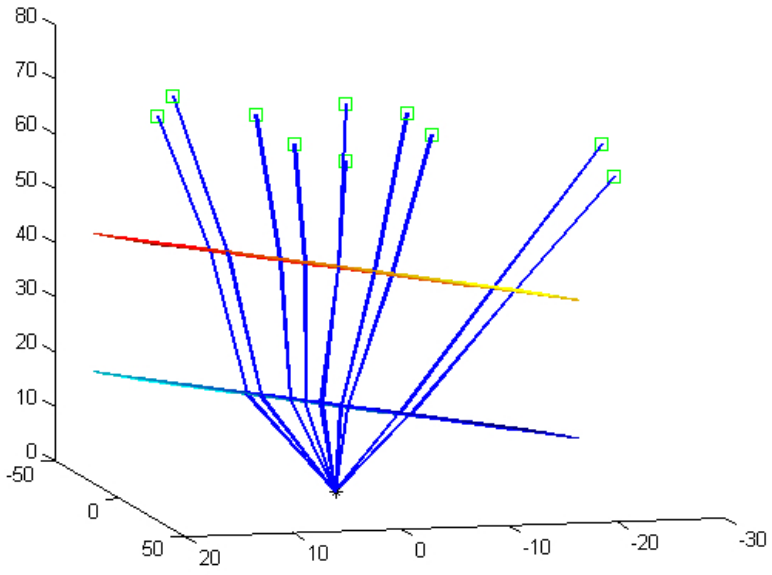


Figure 2.7: A camera that is placed in a water-tight housing.

coordinate system is aligned with the refractive plane, and the camera center **c** is along the Z axis. In this case, the angle θ_1 can be determined, which means that the direction of the refracted ray can be obtained as well. As a result, the coordinate p_x can be written as a linear function of the camera height c_z and the point depth p_z. After that, a set of linear equations can be obtained by making the coordinate of **p** the same from correspondences between two views and by eliminating p_z. The unknown camera center **c** can be obtained by solving the equations. A major limitation of this method is that it requires an IMU to measure the vertical direction of each view. Moreover, an IMU cannot give accurate result when the refractive plane is not perpendicular to the earth's gravity, which is the case when the interface surface is not a calm water surface and when the camera is deployed underwater. The method also

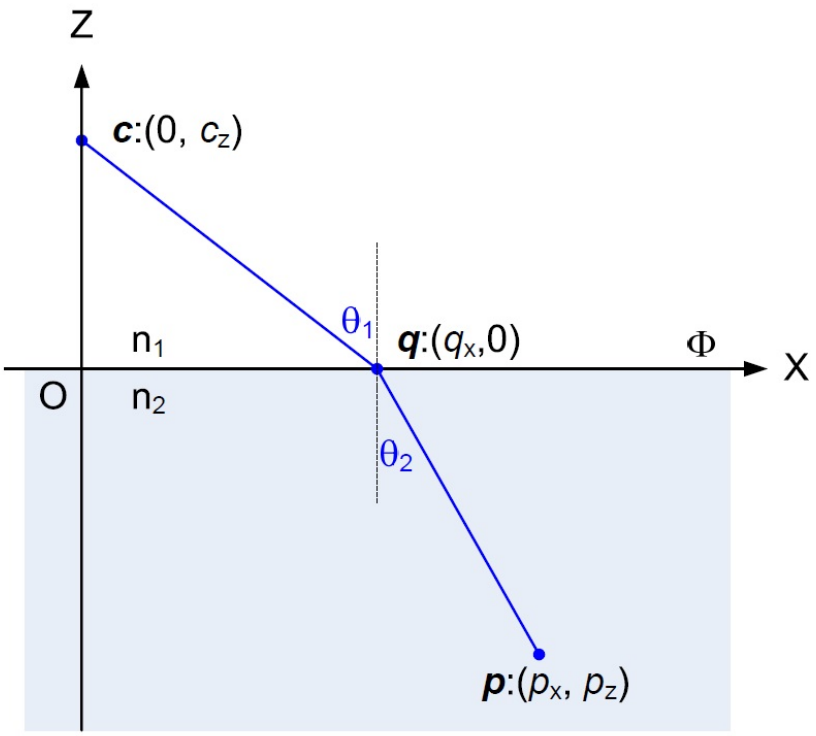


Figure 2.8: Illustration of the proposed method of Chang and Chen [16]. (Image from [16]). In the 3D case, the y axis is orthogonal to the page.

assumes that all the cameras share the same interface which may not be practical.

A flexible method is presented in [81] to calibrate the housing parameters for underwater stereo rigs. The method detects corresponding features from an underwater stereo image pair, and applies a sequence of nonlinear optimizations over the triangulated points, camera poses and refractive parameters. The novelty of this method is the derivation of a "virtual camera" error function, in order to avoid the computation for the refractively-projected point at each optimization iteration. The authors claim that such an error function can improve efficiency. However, the running time

of the optimization process is about 3 hours. Moreover, the result of real data is not evaluated against the ground truth.

Kang *et al.* [48] study two-view structure from motion using cameras that do not share the same interface. This method also starts with a set of pixel correspondences between two camera views. It first demonstrates that when the camera rotations are known, then minimizing the reprojection error based on camera translations, 3D points, and interface distances can be formulated as a convex optimization. Moreover, the method also demonstrates that the camera rotations can be estimated by an algorithm based on Differential Evolution. This method provides some good 3D reconstruction results. However, there are still limitations. First, this method assumes that there is only one refraction, where in most of the cameras there are two refractions which are air $\rightarrow$ glass, and glass $\rightarrow$ water. Second, the refractive interface is assumed to be parallel to the image plane is a major limitation because it is difficult to place the camera inside the housing such that the image plane is exactly parallel to the refractive interface. Last but not least, no quantitative evaluation result against the ground truth is provided.

An efficient calibration method is proposed in [1], which estimate the parameters for a flat refraction camera model. In particular, this model assumes that a camera is viewing the object through multiple parallel flat refractive interface. Fig 2.9 demonstrates the idea of this method. The insight of this method is that the flat refraction model can be regarded as an axial camera, which means that all the imaged rays go through an axis that is perpendicular to the refractive interface, and this axis is

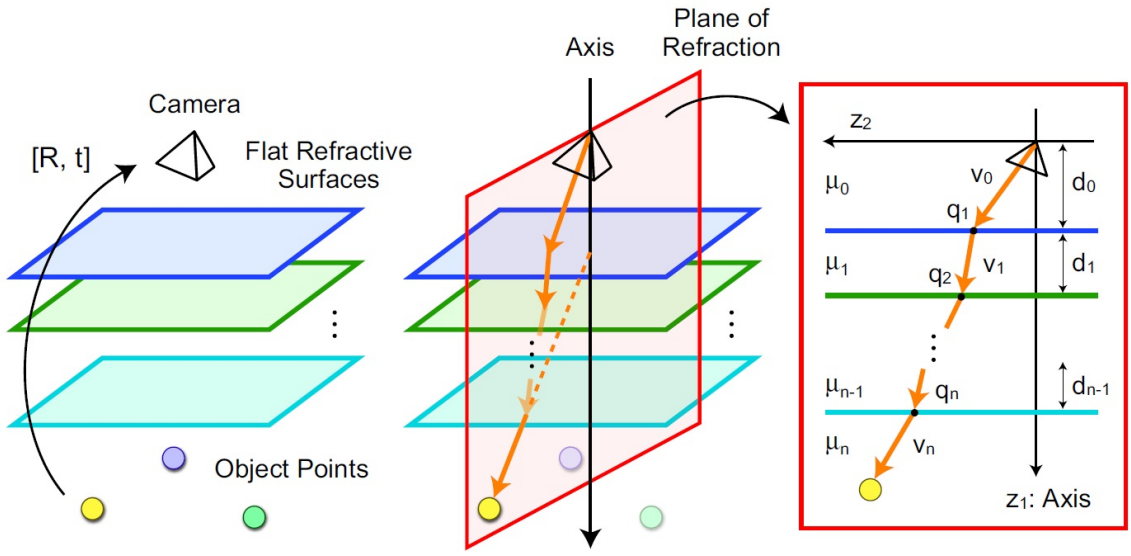


Figure 2.9: Illustration of the proposed method in [1]. (Image from [1])

called the "axis of refraction". Moreover, the path of a particular light ray lies on a plane called the "plane of refraction". With these two important observations, all the refractive parameters can be computed by solving two sets of linear equations. The limitation of this method is that it assumes that the 3D geometry of the calibration target to be known. Therefore, a typical implementation requires to use a checkerboard pattern as the calibration target, which may not be practical when the camera is deployed undersea.

The method proposed in [45] derives a "virtual camera" error function where each 3D object point is projected using an imaginary perspective camera (the "virtual camera"), which is a modified version of their earlier work [81]. An iterative nonlinear optimization strategy is applied to minimize the reprojection error. The authors claim that the method can be applied to estimate both the housing parameters and the

relative camera pose. However, in their simulated experiments the results of housing parameters are not shown and not evaluated. Moreover, in their real experiments, the estimation of relative camera pose is not shown, and the estimated housing parameters are not evaluated against the ground truth.

More recently, Yau *et al.* [90] extend the work of [1] by making use of light dispersion and significant improvement in accuracy is achieved. Light dispersion is a common phenomenon when lights with different wavelengths refract differently at the interface because of different refractive indices. In other words, with a single dichromatic light source, two different wavelengths will be observed at two different locations in the image due to dispersion. The diagram of this method is shown in the left figure of Figure 2.10. It shows that when a light source emits two different wavelengths corresponding to red and blue lights, it reaches the camera at two different locations. The right figure shows the captured image of an array of point light sources emitting red and blue lights. It can be observed that each light source is observed at two pixel locations in the image. With such a useful insight, the authors derive a linear system based on the observation that the two rays reaching the camera center are on the same plane as the refractive normal. By solving this set of linear equations, the refractive normal can be computed. Then, the distance from the camera center to the refractive interface and the layer thickness can be estimated using a similar but simpler method as that presented in [1]. Their experimental results show that their method can achieve a much higher accuracy compared to that of [1]. A major limitation of this method is that it requires a heavy custom-built light box which

weighs over 60lbs, and is impossible to use when the camera system is undersea.

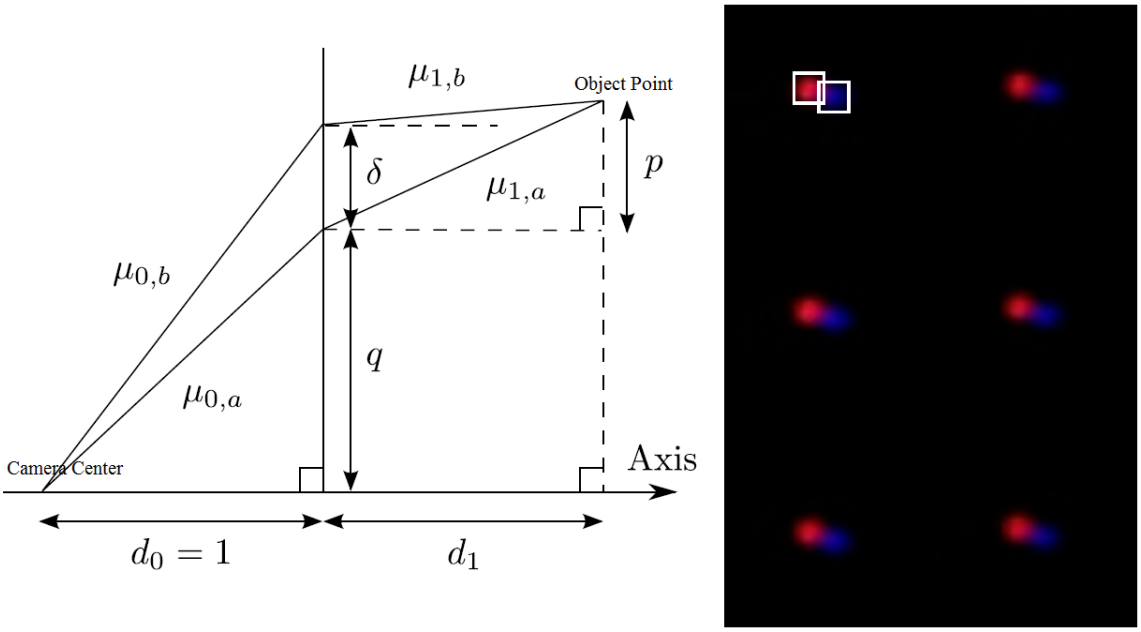


Figure 2.10: Illustration of the proposed method in [90]. (Image from [90])

A virtual camera method presented in [49] models refraction by assuming that each pixel has a different focal length. With this approach, the perspective camera model can be applied. In this case, the relative camera pose can be easily estimated among multiple cameras. The main limitation of this method is that in order to compute the per-pixel focal length, the housing parameters of the camera are required. When a camera system is deployed undersea, the camera may move slightly with regards to its housing equipment due to strong underwater turbulence. Therefore, this method may not be practical in such a scenario.

2.3 Undersea Imaging Systems

Using ROVs or AUVs [31, 33] to explore the ocean is still a popular approach nowadays. They are useful in many ways, mostly because of human safety and the ability to conduct continuous operations. In the intense underwater pressure of the marine environment, ROVs or AUVs eliminate the risk of accidents to the occupants of manned submersibles. The failure of pressure seals or of the life-support system that provides oxygen and removes carbon dioxide is not a concern in ROVs or AUVs. Moreover, ROVs can operate in a continuous manner. In contrast, due to safety concerns, shipboard recovery of these submersibles can often restrict operations to daylight hours. Despite of their advantages, there are still limitations of using ROVs or AUVs to explore the ocean. First, ROVs or AUVs are usually deployed undersea every 4 to 6 months. During each trip, the digital camera(s) and video camera(s) mounted on them are used to collect as much data as possible. However, one fundamental issue is that the data is only available every 4 to 6 months and there is no data in between. Furthermore, it is difficult if not impossible to go back to exactly the same location because of the lack of landmarks and the accuracy of the GPS. In this case, the data may not be very helpful to biologists or ecologists. For example, in order to study how the pollution impact the undersea habitat, the image or video data may need to be continuous. Furthermore, the cost of each trip is very high. In particular, in addition to the time for planning, a typical ROV trip takes 3-7 days to explore different observation sites. A large ship and its entire crew are needed to carry the ROV to the observation site. Additionally, a group of scientists has to be

onsite to navigate the ROV in order to collect data at the location that would be useful to them. Last but not least, the trip can also be impacted by weather. When the undersea turbulence is too strong due to bad weather, the ROV will be very hard to control. Careless control of the ROVs or AUVs could damage both the vehicle and the undersea habitat. Besides ROVs and AUVs, some system [11] uses structured light scanning to create 3D reconstruction for the undersea habitat, but requires a diver to carry the system.

Due to the above limitations of ROVs and AUVs, there are systems being developed to long-term surveillance purpose. A camera surveillance system is presented in [9] to monitor the fish populations. It focuses on the development of software to recognize and track fish from a very large database of videos. Ocean Networks Canada [38, 7] has deployed many undersea systems along the western coast of Canada. Some systems are deployed at over 2000m deep, while others are at about 100m deep. The entire Ocean Networks Canada consists of NEPTUNE and VENUS cabled observatories, where NEPTUNE is an earthquake and tsunami research lab and VENUS an underwater landslide research lab. The collected data are shared with many researchers for data visualization and analysis. The data is potentially useful for many applications, such as studying ocean/climate change, ocean acidification, recognizing and mitigating natural hazards, non-renewable and renewable natural resources. With such a huge observation network, different functionalities can be achieved. Consider NEPTUNE as an example. It can be used to determine the presence of fish sounds by examining the deep sea acoustic recordings [86]. It is also used to study faunal

grouping [46]. The study shows that distinct seasonal faunal groupings are observed, together with summer and winter trends in temperature, salinity and current patterns. Obviously a long-term surveillance system can achieve much more than that from discrete ROV or AUV trips.

The system developed by [83] recover the stereo depth map for an underwater scene. The method does not account for the refraction effects. It is worthy to note that refraction is not the only problem in underwater imaging. The backscattering is another problem when applying a structured light method to an underwater scene. The methods presented in [36, 60] handles underwater imaging with poor-visibility conditions. However, this problem is beyond the scope of this thesis.

Chapter 3

Single Gaussian Blurred Pattern

In this chapter, a new single shot structured light method is presented to recover dense depth maps. Contrary to most temporal coding methods which require projecting a series of patterns, the new method needs one color pattern only. Unlike most single shot spatial coding methods which establish correspondence along the edges of the captured images, it produces a dense set of correspondences. The new method is built based on an important observation that a Gaussian blurred De Bruijn pattern preserves the desirable windowed uniqueness property. A Gaussian blurred De Bruijn pattern is used so that the color of every illuminated pixel is used to its fullest advantage. The simulated experiments show that the proposed method establishes a correspondence set whose density and accuracy are close to that of using a temporal coding method. The robustness of the new approach is demonstrated by applying it to several real-world datasets.

3.1 Gaussian Blurred De Bruijn Pattern

The pattern used in this method is based on a stripe pattern generated by Zhang *et al.* [92], which is shown in Figure 3.1. The pattern consists of 125 vertical stripes using eight different colors. It is generated by the generator that is publically available online [75] and is based on the De Bruijn sequence [26]. A De Bruijn sequence of order o over an alphabet of a symbols is a cyclic string of length a^o. It has the property that each sub-sequence of length o appears exactly once, which is referred to as the *windowed uniqueness* property. An example of a De Bruijn sequence is $S = 00101110$ which has an order $o = 3$. It means that any substring of S with a length of 3 appears exactly once. Because of the windowed uniqueness property, global optimization methods such as dynamic programming (DP) can be applied to establish correspondences.

Figure 3.1: Stripe pattern used in Zhang *et al.* [92] (Image from [92]).

Gaussian blur is applied to the pattern shown in Figure 3.1, which results in a blurred pattern shown in Figure 3.2. It is observed that after applying Gaussian blur, the resulting pattern maintains the windowed uniqueness property as long as the original pattern has such a property. Based on this property, a blurred De Bruijn pattern can be used in a structured light system, and can produce a dense set of

correspondences. In order to mathematically prove the correctness of this property, a lemma which explores an important feature of Gaussian blur is introduced first.

Figure 3.2: Blurred pattern after Gaussian blur is applied to the pattern in Figure 3.1.

Throughout this section, the kernel size for Gaussian blur is assumed to be $2m+1$. The weight can be written as $\mathbf{W} = \{w_1, ..., w_{2m+1}\}$ and $w_i = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(i-(m+1))^2}{2\sigma^2}}$. Here $i \in \{1, ..., 2m+1\}$ and $m+1$ is the index of the kernel center. Given two sets of *integers* $\mathbf{C} = \{c_1, ..., c_{2m+1}\}$ and $\mathbf{C'} = \{c'_1, ..., c'_{2m+1}\}$ where $\mathbf{C} \neq \mathbf{C'}$, the following lemma is introduced.

Lemma 1. *Given the above described* $\boldsymbol{W}$, $\boldsymbol{C}$ *and* $\boldsymbol{C'}$, *assuming that* σ *is a rational number, then* $\sum_{i=1}^{2m+1}(c_i - c'_i)w_i \neq 0$ *as long as the following constraint is satisfied:* $c_{m+1} \neq c'_{m+1}$ *or there exists at least one* j *such that* $c_{m+1-j}+c_{m+1+j} \neq c'_{m+1-j}+c'_{m+1+j}$, *where* $j \in \{1, ..., m\}$.

Proof. Since the weight of a Gaussian kernel is symmetric with respect to its center, the following equation can be derived

$$\sum_{i=1}^{2m+1}(c_i - c'_i)w_i = \sum_{i=1}^{m}\Big((c_{m+1-i} + c_{m+1+i}) - (c'_{m+1-i}+ c'_{m+1+i})\Big)w_{m+1-i} + (c_{m+1} - c'_{m+1})w_{m+1}. \tag{3.1}$$

Notice that even if **C** $\neq$ **C'**, there could exist some $c_k = c'_k$ and that $w_k c_k = w_k c'_k$. However, all the same items are canceled with the operation of $c_i - c'_i$ in Eq. 3.1. Because it is assumed that **C** $\neq$ **C'**, there is always a certain i such that $c_i \neq c'_i$. With the above discussion, Eq. 3.1 can be re-written as follows:

$$\sum_{i=1}^{2m+1}(c_i - c'_i)w_i = \frac{1}{\sqrt{2\pi\sigma^2}}\left(f_1 e^{-\frac{k_1^2}{2\sigma^2}} + f_2 e^{-\frac{k_2^2}{2\sigma^2}} + ... + f_l e^{-\frac{k_l^2}{2\sigma^2}}\right), \tag{3.2}$$

where

$$f_i = c_{m+1-k_i} + c_{m+1+k_i} - (c'_{m+1-k_i} + c'_{m+1+k_i}). \tag{3.3}$$

Note that none of the terms in Eq. 3.2 is 0, i.e. $f_1 \neq 0, ..., f_l \neq 0$, because all the $c_i = c'_i$ are canceled. Denote $a = e^{-\frac{1}{2\sigma^2}}$. Since e is a transcendental number and it is assumed that σ is a rational number, it follows that a is also transcendental [6]. On the other hand, Eq. 3.2 can be written as

$$\sum_{i=1}^{2m+1}(c_i - c'_i)w_i = \frac{1}{\sqrt{2\pi\sigma^2}}\left(f_1 a^{k_1^2} + f_2 a^{k_2^2} + ... + f_l a^{k_l^2}\right). \tag{3.4}$$

Since $f_1 \neq 0, ..., f_l \neq 0$, Eq. 3.4 can never equal to 0. It can be seen by *reductio ad absurdum*, if $\sum_{i=1}^{2m+1}(c_i - c'_i)w_i = 0$, then $a = e^{-\frac{1}{2\sigma^2}}$ is algebraic, which is clearly a contradiction. □

In order to better understand the constraint, an example that violates the constraint is provided. The two sets of integers are {0, 100, 200, 150, 50} and {20, 120, 200, 130, 30} which are different. However, since $(0 + 50) = (20 + 30)$ and $(100 + 150) = (120 + 130)$ and $200 = 200$, hence $(0 + 50) \times w_1 + (100 + 150) \times w_2 + 200 \times w_3 = (20 + 30) \times w_1 + (120 + 130) \times w_2 + 200 \times w_3$. This example illustrates a

scenario that violates the constraint, and hence the results are the same even if the input sequences are different.

Using Lemma 1, it can be proved that the windowed uniqueness property is preserved when applying Gaussian blur to a De Bruijn pattern. Given a De Bruijn pattern without applying Gaussian blur and assume that a set of n contiguous pixels appear exactly once. Suppose that there are two sets of n contiguous pixels that are different from each other, then there exists at least one color channel that is different and this method focuses on that channel only. Denote the pixel value of that channel to be P and Q for the two sets. To be more specific, $P = p_1...p_n$ and $Q = q_1...q_n$. $P \neq Q$ represents that there exists one or more i such that $p_i \neq q_i$ where $i \in \{1, ..., n\}$. After applying Gaussian blur to P and Q, the results are denoted as $P' = p'_1...p'_n$ and $Q' = q'_1...q'_n$, respectively. Theorem 1 proves that $P \neq Q \rightarrow P' \neq Q'$. It states that if two sets of n contiguous pixels are different, then the results of applying Gaussian blur to them are different, which means that the result has the windowed uniqueness property.

Theorem 1. *Given the above assumptions,* $P \neq Q \rightarrow P' \neq Q'$.

Proof. Denote the neighboring pixels of p_i as $\mathbf{N_{p_i}} = \{p_{i-m}, ..., p_i, ..., p_{i+m}\}$, and $\mathbf{N_{q_i}} = \{q_{i-m}, ..., q_i, ..., q_{i+m}\}$ for q_i. According to the definition of Gaussian blur, p'_i is the weighted sum of $\mathbf{N_{p_i}}$. That is, $p'_i = \sum_{j=i-m}^{i+m} w_j p_j$ and $q'_i = \sum_{j=i-m}^{i+m} w_j q_j$. Since $P \neq Q$, there exist a certain i such that $\mathbf{N_{p_i}} \neq \mathbf{N_{q_i}}$. Let's divide the relationship between $\mathbf{N_{p_i}}$ and $\mathbf{N_{q_i}}$ into two cases.

1) $\mathbf{N_{p_i}}$ and $\mathbf{N_{q_i}}$ satisfy the constraint in Lemma 1, and

2) They do not satisfy the constraint.

❶ From Lemma 1, it can be easily deduced that in the first case, $p'_i \neq q'_i$, therefore $P' \neq Q'$.

❷ In the second case, $\mathbf{N_{p_i}}$ and $\mathbf{N_{q_i}}$ violate the constraint, which means that $p'_i = q'_i$. In order to prove the theorem, it is required to show that if $\mathbf{N_{p_i}}$ and $\mathbf{N_{q_i}}$ violate the constraint such as the example shown after Lemma 1, then $\mathbf{N_{p_{i+1}}}$ and $\mathbf{N_{q_{i+1}}}$ can never violate the constraint. If that is true, it is inferred that $p'_{i+1} \neq q'_{i+1}$ by Lemma 1, and hence $P' \neq Q'$.

The basic procedure of proving that $\mathbf{N_{p_{i+1}}}$ and $\mathbf{N_{q_{i+1}}}$ never violate the constraint is by *reductio ad absurdum* and is described as follows. From the definition of $\mathbf{N_{p_i}}$ and $\mathbf{N_{q_i}}$, it can be derived that $\mathbf{N_{p_{i+1}}} = \{p_{i-m+1}, ..., p_i, p_{i+1}, p_{i+2}, ..., p_{i+m+1}\}$ and $\mathbf{N_{q_{i+1}}} = \{q_{i-m+1}, ..., q_i, q_{i+1}, q_{i+2}, ..., q_{i+m+1}\}$, where $i+1$ is now the index of the center. Suppose $\mathbf{N_{p_{i+1}}}$ and $\mathbf{N_{q_{i+1}}}$ also violate the constraint, if it can be proved that $\mathbf{N_{p_i}} = \mathbf{N_{q_i}}$, which is clearly a contradiction, then the theorem follows.

Suppose $\mathbf{N_{p_{i+1}}}$ and $\mathbf{N_{q_{i+1}}}$ violate the constraint, it can be inferred that,

$$p_{i+1} = q_{i+1} \tag{3.5}$$

and

$$p_{i+1-j} + p_{i+1+j} = q_{i+1-j} + q_{i+1+j} \quad j \in \{1, ..., m\}. \tag{3.6}$$

Since $\mathbf{N_{p_i}}$ and $\mathbf{N_{q_i}}$ violate the constraint, it is inferred that,

$$p_i = q_i \tag{3.7}$$

and

$$p_{i-j} + p_{i+j} = q_{i-j} + q_{i+j} \qquad j \in \{1, ..., m\}. \tag{3.8}$$

Putting (3.5) and (3.8) together it can be inferred that $p_{i-1} = q_{i-1}$. (3.6) and (3.7) infer that $p_{i+2} = q_{i+2}$. $p_{i+2} = q_{i+2}$ and (3.8) infer that $p_{i-2} = q_{i-2}$. $p_{i-1} = q_{i-1}$ and (3.6) infer that $p_{i+3} = q_{i+3}$, and this last one with (3.8) infer that $p_{i-3} = q_{i-3}$. $p_{i-2} = q_{i-2}$, with (3.6) infer that $p_{i+4} = q_{i+4}$ and this together with (3.8) infer that $p_{i-4} = q_{i-4}$. Hence, by putting Eq. 3.5, 3.6, 3.7, 3.8 together, it can be inferred that $p_{i-j} = q_{i-j}$ for all $j \in \{1, ..., m\}$. Furthermore, since $p_i = q_i$, it can be inferred that $\mathbf{N_{p_i}} = \mathbf{N_{q_i}}$, which contradicts the assumption. Therefore, $P' \neq Q'$. □

It is worthy to note that Gaussian blur is carefully selected because it maintains the windowed uniqueness property and other linear filters may not. The following is an example to illustrate that. Given an order 3 De Bruijn sequence {0,120,150,150,120,0,180, 90,150,180}. If an average filter is applied over a 1-D window of 3 values, where each boundary is padded with 0, the result is {40,90,140,140,90,100,90,140,140,110}. It is obvious that the sub-sequence {90,140,140} appears twice, and hence the windowed uniqueness property is broken. The result of applying Gaussian blur with kernel size set to be 3 and $\sigma = 0.95$ is {29,87,128,128,87,72,97,117,128,112}, where the windowed uniqueness property still holds. Another very important assumption in Lemma 1 is that σ is a rational number. If this constraint is not satisfied, the lemma may fail. Given a sequence {0,100,50,100,150,0,100,100,100,0} as an example. Let the size of Gaussian kernel to be 3 and assume that the scaled weights of the Gaussian kernel is $\{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$. The result of applying the scaled weight to the sequence

is {25,62.5,75,100,100,62.5,75,100,75,25}. It can be seen that the sub-sequence of {62.5,75,100} appears more than once. In this case, $\sigma = (2\ln 2)^{-\frac{1}{2}}$ and is clearly not a rational number. Hence, only by using such a "reverse engineering" technique which computes the specific weight first and calculates a non-rational σ accordingly, the windowed uniqueness property might be violated. Furthermore, a rational number is commonly used in practice when selecting σ for Gaussian blur. Therefore, the windowed uniqueness property is preserved after Gaussian blur by selecting a rational σ. Also, when selecting σ, extremely small number such as 0 should be avoided so that the pattern can be blurred. The determination of an optimal value for σ is an interesting problem for future research.

It has been proved that Gaussian blur preserves the windowed uniqueness property of a De Bruijn pattern, such property of a Gaussian blurred De Bruijn (GBDB) pattern has never been studied. Because of such property, global minimization methods can be applied to establish dense correspondences when a GBDB pattern is used.

3.1.1 Establishing Correspondences

Based on the windowed uniqueness property of a GBDB pattern, DP can be used to establish correspondences.

Traditional spatial coding methods normally have two steps. The first is to detect edges from both the captured image and the projected pattern. The second is to match the detected edges using edge gradients. Therefore, these methods only recover matched pixels along edges, which are sparse. As a result, these methods are rarely

used in depth map recovery. One major advantage of the new method is that it provides a dense set of correspondences. In the experimental setup, a projector is used to project pattern(s) and two cameras are used as a stereo rig instead of only one camera. The pixels in the two captured images are matched instead of matching the edges in the captured image with the edges in the projected pattern. Using this setup, the number of established correspondences is not limited by the resolution of the projector anymore. Furthermore, it is natural to match the pixels since the illumination condition provided by the projector will be similar from both cameras' viewpoints. It is discovered that although pixels belonging to the same color stripe have the same color in the pattern, their color can be different when the pattern is blurred. This can be seen in Figure 3.3, which is a cropped image from one of the

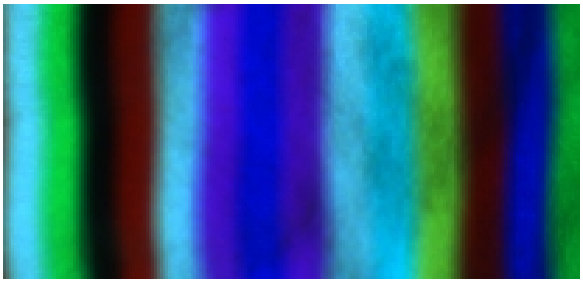


Figure 3.3: A sub-region of a captured image in one of the experiments.

experimental datasets. It is clear that surface points that are illuminated by the same color stripe are likely to have different colors. Combining this observation with the windowed uniqueness property of a GBDB pattern, DP can be applied to establish correspondence for almost every illuminated pixel that is not occluded.

It is assumed that the captured images are rectified so that the correspondence can be established one scanline at a time. DP is applied to the new method in order to find the matching and the process is similar to the method described in [20]. That is, the color of every illuminated pixel is matched along each scanline in the two captured images. This is valid due to two reasons. First, the windowed uniqueness property of a GBDB pattern guarantees that a set of contiguous pixels along each scanline appears only once. Second, the color within one color stripe will be different in the captured images because of the blur. The cost function used in the new method is defined as

$$cost(I_1, I_2) = \frac{\sum_{k=R,G,B} \left(\frac{I_1(k) - I_2(k)}{255} \right)^2}{3} + \epsilon. \tag{3.9}$$

Here I_1, I_2 are the colors of two pixels along the same scanline from the two captured images. The cost function measures the similarity between two colors. ϵ is a very small number set to be 10^{-5} to make sure that the cost is larger than 0. The pseudo-code for establishing correspondences can be found in [20].

3.2 Experimental Results

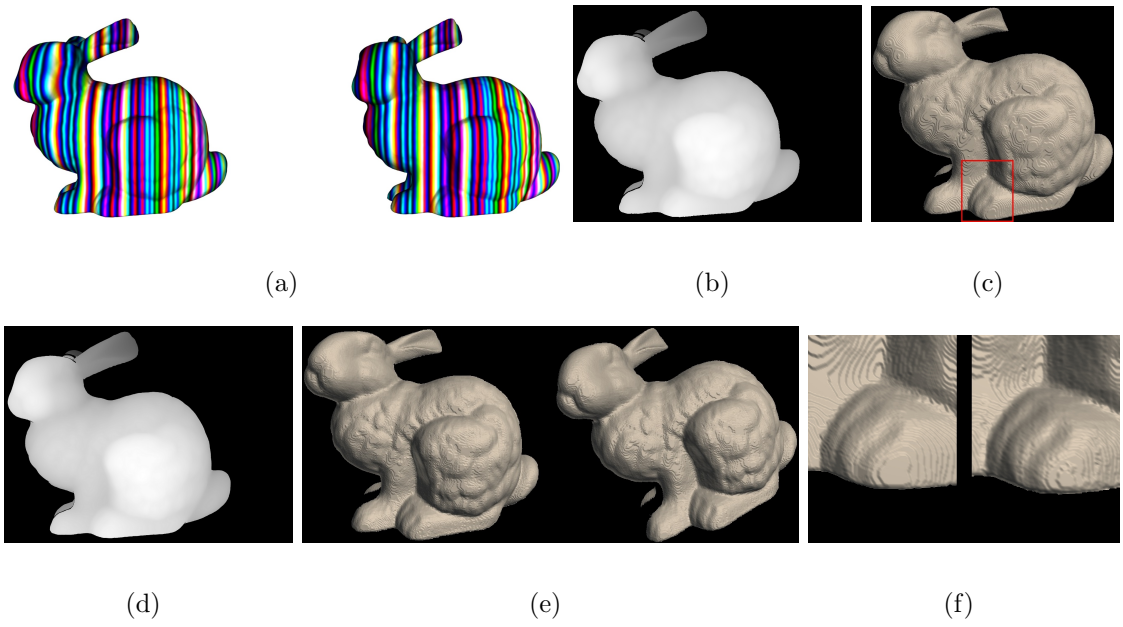
The GBDB pattern is applied to several datasets to demonstrate that the property is useful to a structured light system, and both quantitative and qualitative evaluations are provided.

3.2.1 Simulated Datasets

The Stanford Bunny model serves as an object in the experiments. Maya is used to project the GBDB pattern onto the bunny. Two cameras are set in Maya to capture the scene from different viewpoints. The captured images are shown in Figure 3.4(a). The blurred pattern used is the result of applying Gaussian blur with kernel size of 15×15, and its resolution is 868×768. In the captured images, the edges of the color stripes in the pattern are blurred. The ground truth of the depth map from one viewpoint is shown in Figure 3.4(b) and is obtained using Maya as well. Figure 3.4(c) shows the 3D mesh corresponding to the ground truth. Figure 3.4(d) shows the recovered depth map by the new method and Figure 3.4(e) shows the reconstructed 3D surface mesh from two different viewpoints. Both the depth map and the 3D mesh demonstrate that the result produced by the new method is very hard to distinguish from the ground truth. Furthermore, the comparison with the ground truth in the close up view of the part inside the rectangle in Figure 3.4(c) is shown in Figure 3.4(f). In this figure, the left one is from the ground truth and the right one from the result by the new method. Obviously this figure shows that the new method can recover very fine details on the object surface.

To evaluate the new method quantitatively, two measures metrics are used.

1. Accuracy Percentage: An important advantage of the new method is to provide dense correspondences with only one shot. Suppose the number of pixels whose depth are recovered by the new method is M, and the number of pixels whose depth values are valid in the simulated ground truth is N, then the accuracy percentage is



(a) (b) (c)

(d) (e) (f)

Figure 3.4: Result of simulated experiment. (a) Captured images using Maya. (b) Simulated ground truth. (c) 3D mesh corresponding to (b). (d) Depth map recovered by the new method. (e) 3D mesh by the new method. (f) Close up comparison.

defined to be $\frac{M}{N} \times 100\%$.

2. RMS error: Another advantage of the new method is high accuracy. In order to measure that, an error metric is applied which is the same as the one most commonly applied in stereo vision [79]. For every pixel whose disparity is valid in the simulated ground truth, its disparity is denoted as $d_T(x, y)$ in the ground truth and $d_C(x, y)$ recovered by the new method. Then the RMS (root-mean-square) error is defined as

$$RMS = \sqrt{\frac{1}{N} \sum_{(x,y)} |d_C(x, y) - d_T(x, y)|^2}. \tag{3.10}$$

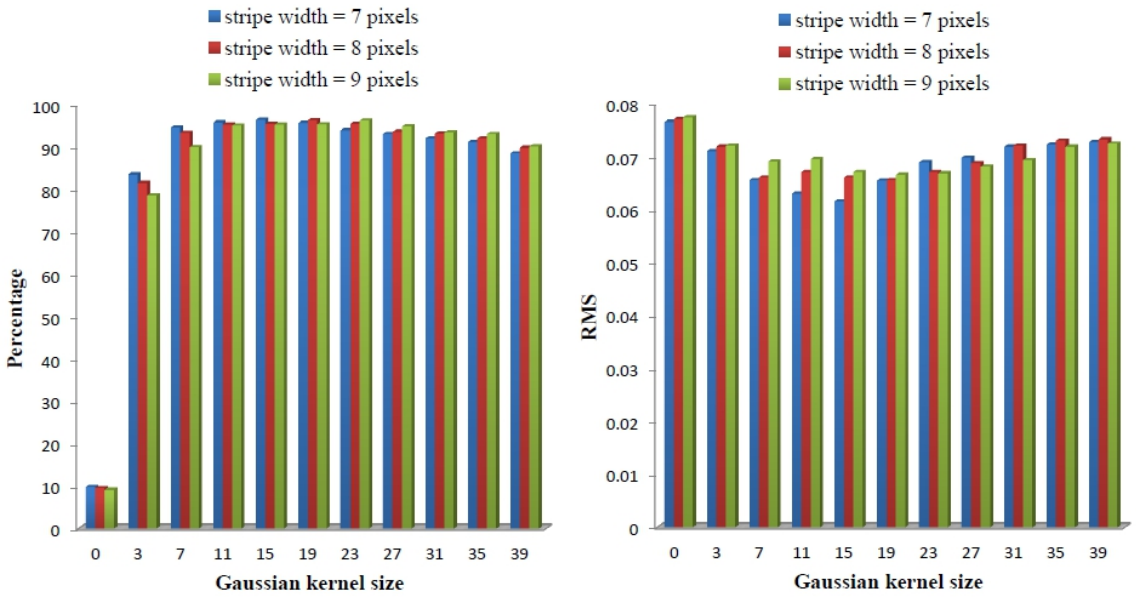
Comparison with Temporal Coding

Figure 3.4(d) shows the recovered depth map by the new method, and the accuracy percentage is 96.39% with RMS error = 0.06155. To better understand the numbers, another simulated experiment is designed. This experiment is the same as the one shown in [80], which serves as to obtain the ground truth depth maps for the datasets used in [41]. It is a temporal coding technique and can be briefly described as follows. A set of 20 coded patterns and their inverse patterns are projected onto the scene and the codewords for each pixel from the captured images are obtained so that the correspondences can be established via codewords. Applying this method, the correspondence of every pixel can be found unless it is occluded. This experiment is employed in order to find the accuracy percentage of occluded pixels from the captured images. The accuracy percentage of pixels that are not occluded is 98.11%. Therefore, only a small part (1.72%) of the un-occluded pixels is not recovered by the new method. The RMS error for the temporal coding method is 0.05899. Comparing with this number, the accuracy of the new method is close to the temporal method, and the new method requires projecting only one pattern instead of 40 in the temporal coding method.

Robustness Test

More datasets are generated using Maya to demonstrate that the new method is robust to noise. To be more specific, datasets are generated by applying Gaussian blur to Figure 3.1 with different kernel size. The Gaussian kernel size varies from 3×3 to 39×39. Besides, the width of each color stripe in Figure 3.1 is 7 pixels, and more datasets are generated with width of 8 pixels and 9 pixels. The graphs of both

accuracy percentages and RMS error are shown in Figure 3.5.



(a) (b)

Figure 3.5: Graphs of accuracy percentage and RMS error in the simulated experiment with varying Gaussian kernel sizes and stripe widths. (a) Accuracy percentage graph. (b) RMS error graph.

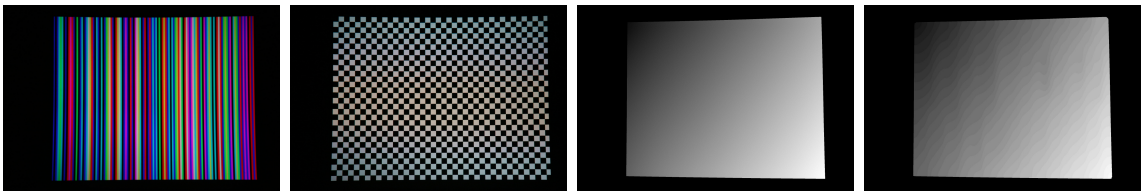
A traditional spatial coding method is implemented and its accuracy percentage and RMS error are shown as the point when the kernel size is 0. Since the traditional spatial coding method requires edge detection, it can be applied to the dataset only when a non-blur pattern is projected. A U-shaped or V-shaped bar graph is expected for the error graph, in which case the accuracy first improves and then degrades as the kernel size increases. The accuracy percentage graph is expected to be inverted U-shaped or inverted V-shaped. The shape of the graphs can be explained by the nature

of Gaussian blur and DP. When the size of Gaussian kernel is small, the color within one stripe is similar in the captured images. Hence, the matching percentage is smaller and there are ambiguities which cause a larger error when applying DP. When the kernel is increased to a certain size, the color within one color stripe varies significantly in the captured images. Therefore, more correspondences can be established with fewer ambiguities. However, when the size of Gaussian kernel is further increased, more neighboring pixels are mixed, and hence, more pixels appear similar again. As a result, when the kernel size is too big, the number of matched pixels decreases and the error increases. Through this figure, it shows that the density of the established correspondences increases dramatically by applying the new method when comparing with the traditional spatial coding method. Moreover, the accuracy is always higher than that of applying the traditional spatial coding method, as shown in Figure 3.5(b). It is clear that the shapes of both graphs are similar to what are expected.

3.2.2 Real-World Datasets

In the real-world experiments, the blurred pattern is not used anymore. Instead a more elegant idea is applied which is to set the projector out of focus to achieve a similar effect of projecting a Gaussian blurred pattern. Indeed, projector defocus has been used in the computer vision area. In particular, it has been pointed out [69] [98] [35] [13] that when the projector is out-of-focus, the display is uniformly blurred by a PSF (point-spread function), which can be modeled as a blur kernel. This blur kernel is generally assumed to be smooth and well approximated by a 2D Gaussian.

Therefore, when the projector is out-of-focus, it is similar to projecting a Gaussian blurred pattern. In the real world, projectors may not be able to focus on every part of the object, in which case the traditional spatial coding methods would have trouble with detecting edges in the parts that are out of focus. It can also cause problems to temporal coding methods when identifying the codewords. Throughout all the experiments, two Canon 5D cameras are used to capture images and a Samsung SP-P410M LED projector is used to project patterns. The resolution of the cameras is 2496×1664, and the projector is 800×600. It shows that even if the projector is out of focus, the new method can provide dense correspondences and accurate results.



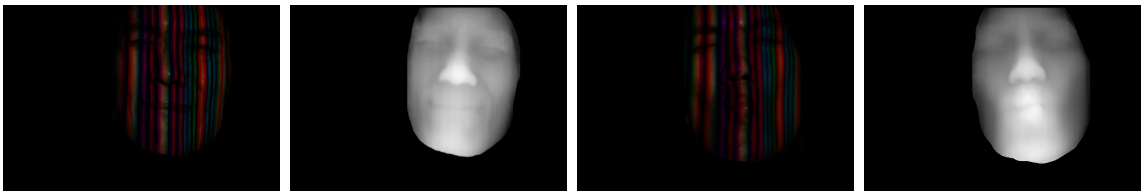
(a) (b) (c) (d)

Figure 3.6: Quantitative evaluation. (a) Captured image with a De Bruijn pattern projected onto a wall. (b) Captured image when a grid pattern is projected. (c) Ground truth. (d) Depth map recovered by the new method.

Quantitative evaluation is provided in the first experiment. In particular, the pattern is projected onto a wall which is a flat surface, and the captured image is shown in Figure 3.6(a). To obtain the ground truth, several shifted checkerboard patterns are projected and the captured image with one of the projected pattern is shown in Figure 3.6(b). The corners are extracted in both camera views and matched

to obtain the ground truth in subpixel accuracy. When projecting the checkerboard patterns, the projector is set to focus on the wall so that the corners are sharp. However, after that, its focus setting is carefully changed without moving its position so that the Gaussian blurred pattern can be projected. The ground truth of the depth map is shown in Figure 3.6(c). The result by the new method is shown in Figure 3.6(d), and its RMS error is 0.273 pixels which is very small.

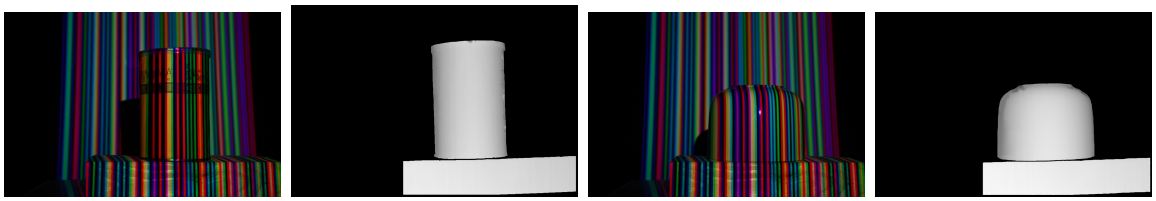
Since the new method requires no temporal information, it can be applied to both static and dynamic scenes as well. In these experiments, the new method is applied to several different scenes with static and dynamic objects.



(a) (b) (c) (d)

Figure 3.7: Results of face scene. (a,c) Two images with different facial expression and defocus level. (b,d) Depth maps recovered by the new method for (a) and (c), respectively.

The depth maps are shown for visual qualitative evaluation. Figure 3.7 shows the result when the new method is applied to capture facial expressions. Two images with different facial expressions and different defocus levels from one viewpoint are shown. Only images from one viewpoint are shown and the regions that include human hair or are not illuminated are masked out. The depth maps shown in Figure 3.7(b) and



(a) (b) (c) (d)

Figure 3.8: Results of cup and mug scene. (a,c) Captured images of cup and mug scene. (b,d) Depth maps recovered by the new method for (a) and (c), respectively.

3.7(d) indicate that the results are visually pleasant. The new method is also applied to object whose surface color is not neutral, as shown in Figure 3.8. It shows that even when there are colors on the object surface, the new method can still recover the depth map.

3.3 Discussion

First, let's compare the new method to conventional temporal and spatial coding methods. Comparing to the temporal coding methods, it is not limited to static scenes, and still gives comparable accuracy as a temporal coding method which has been demonstrated in simulated experiments. Comparing to the spatial coding methods, it produces dense depth maps, while the former only recovers the depth along edges.

In Section 3.1, a proof of an important property is presented that a GBDB pattern also maintains the windowed uniqueness property, so that DP can be applied to establish correspondences. Based on the GBDB pattern, the color information of the

illuminated pixels is fully used to provide much denser correspondences comparing to the traditional spatial coding methods. This property has been verified in both simulated and real-world experiments. It is noteworthy that traditional spatial coding methods usually require the detection of edges in the captured images. This step may not be robust enough due to the image processing techniques that are applied. However, the new method does not require edge detection, and hence is more robust to noisy conditions. Last but not least, since projectors normally have large apertures, the in-focus region is very limited. Therefore, the patterns can be blurred when projected onto the scene if the depth range of the scene is big, in which case, the edge detection in the traditional spatial coding method would fail. However, the new method can still establish correspondences because it takes advantage of a blurred pattern.

One final experiment is designed to compare the new method with other methods. The traditional spatial coding methods are edge-based while the new method is intensity-based. Moreover, the pattern can be blurred or non-blurred. Therefore, there are four possible combinations. These four combinations are applied to the simulated datasets. In particular, a non blurred pattern and a Gaussian blurred pattern with kernel size of 7×7 are used. Quantitative evaluation is provided and the accuracy percentage and RMS error are used as measurement metrics. Table 3.1 shows the evaluation results. From the table it can be seen that using an intensity-based method provides better results. Moreover, the combination of intensity-based with blurred pattern provides the best results.

Accuray Percentage + RMS Error	Blurred Pattern	Non-blurred Pattern
Edge-based	11.34% + 0.1153	13.44% + 0.07578
Intensity-based	95.01% + 0.06148	76.00% + 0.07395

Table 3.1: Evaluation of two methods with two types of patterns.

There are certain limitations of our method. First of all, it has been shown in Figure 3.5 that the accuracy of the proposed method decreases when the Gaussian kernel size increases. When the kernel size is very large, the windowed uniqueness property may be violated broke and the proposed method could fail. Moreover, surface colors could also alter the observed colors and hence, could impact the windowed uniqueness property.

In summary, a depth recovery method is presented which requires only one shot and yet provides accurate depth map with density much higher than traditional spatial coding methods. Moreover, the projector needs not be focused on the objects, which is more realistic in real world applications.

Chapter 4

Scene Adaptive Patterns

Even though the previously designed Gaussian blurred pattern can be used to recover dense depth map, it cannot be applied to scenes with global illumination because it could potentially broke the uniqueness property of the pattern. In this chapter, a new structured light 3D reconstruction method is presented that can be applied to scenes in the presence of global illumination such as inter-reflection, subsurface scattering and severe projector defocus. In particular, inter-reflection is regarded as long-range effects, whereas subsurface scattering and projector defocus as short-range. The proposed method takes advantage of important properties embedded in the binary code patterns, and determines the minimum stripe width that eliminates the effects of subsurface scattering and projector defocus in the capturing stage. Moreover, errors caused by inter-reflection are detected, and corrected by an iterative approach. The new method can be applied to scenes where more than one global illumination effect appears in a scene point. The accuracy of the new method is

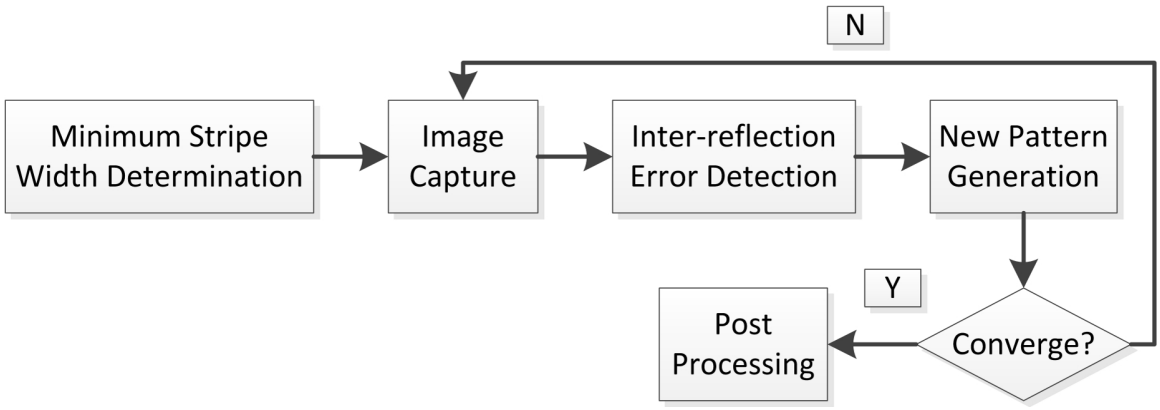


Figure 4.1: Pipeline of the presented method.

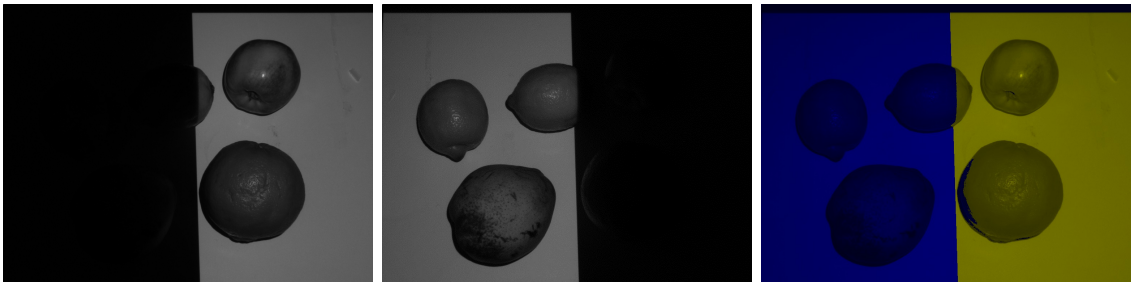
demonstrated by quantitative evaluation, and its robustness by qualitative evaluation when applying to real-world scenes with various characteristics, both of which are better than two of the currently known best methods.

The pipeline of the new method is shown in Figure 4.1. The first stage is to determine the minimum stripe width that eliminates the blurry effects in the subsequent image capturing process. The second stage is an iterative approach by which the error caused by inter-reflection is detected and corrected. Following that, a post-processing method is applied to refine the correspondences.

4.1 Short-range Effects

When gray code or binary code patterns are used, both the patterns and their inverse patterns are usually projected in order to determine whether an image pixel is under the illumination of a white or black stripe. This process is illustrated in Figure 4.2.

Figure 4.2(a) is a captured image when a black/white pattern is projected, and 4.2(b) is when its inverse pattern is projected. Figure 4.2(c) is the decoded bit for 4.2(a), where blue indicates that the region is under the illumination of a black stripe and yellow a white stripe. That is, 0 for blue and 1 for yellow. In this section, the term "bit-change" is used to denote the scenario of two neighboring pixels having different bits. To determine the minimum stripe width that could eliminate the short-range



(a) (b) (c)

Figure 4.2: (a) A captured image under the illumination of a black/white pattern. (b) Image with its inverse pattern. (c) Decoded bit for (a).

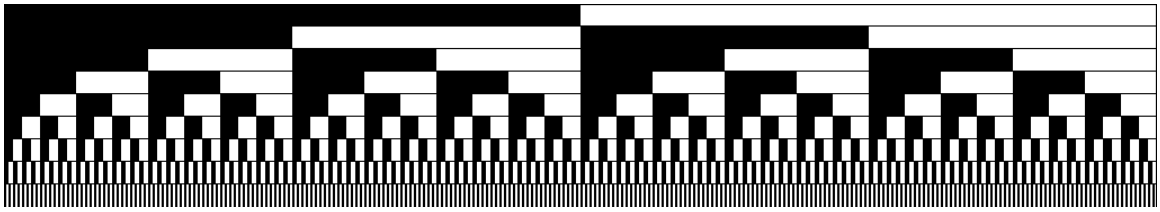


Figure 4.3: The patterns used in the new method.

effects, binary code patterns are used which are shown in Figure 4.3. Notice that

the new method does not use the pattern with a stripe width of 1 pixel because this pattern can be easily blurred when projected. Therefore, Figure 4.3 has 9 patterns instead of 10. the binary code is chosen over the gray code patterns because of certain desired properties of which the new method will take advantage.

4.1.1 Image Formation Model

The notation $t_{n_1:n_2}$ is used to denote that t is an integer in the range $[n_1, n_2]$, where both n_1 and n_2 are integers and $n_2 > n_1$. Moreover, $T_{n_1:n_2}$ denotes the set of all integers in the range $[n_1, n_2]$. $P(T_{1:9}) = \{P(t_{1:9})\}$ denotes the set of projected patterns shown in Figure 4.3, where $P(t_{1:9})$ denotes a projected pattern with stripe width $SW(t_{1:9}) = 2^{10-t_{1:9}}$. The unit of the stripe width is pixel. The new method is presented based on vertical stripe patterns where the horizontal ones are very similar. Therefore bit-change refers to two neighboring pixels on the same horizontal scanline and have different decoded bits. It can be easily modified for horizontal patterns. Assume that the intensity of the captured image is denoted as $I(t_{1:9})$ when projecting $P(t_{1:9})$ and is obtained by

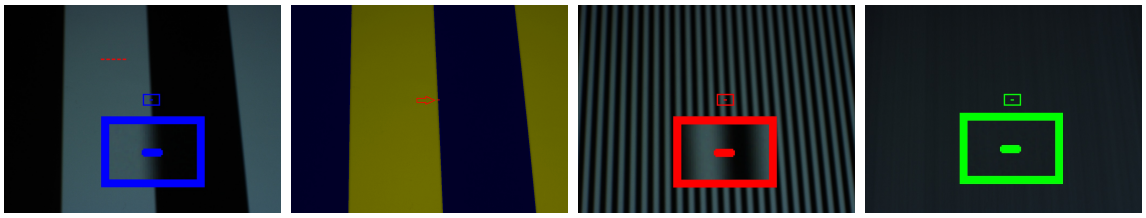
$$I(t_{1:9}) = P(t_{1:9}) * PSF_p \cdot R * PSF_c. \tag{4.1}$$

In here, $*$ denotes convolution and $\cdot$ multiplication. PSF_p represents the point spread function for the projector and PSF_c for the camera. R stands for the surface reflectance which depends on the object material and is invariant to $t_{1:9}$. For a static scene point, PSF_p, PSF_c and R are fixed during the capturing stage once the camera and projector settings are fixed.

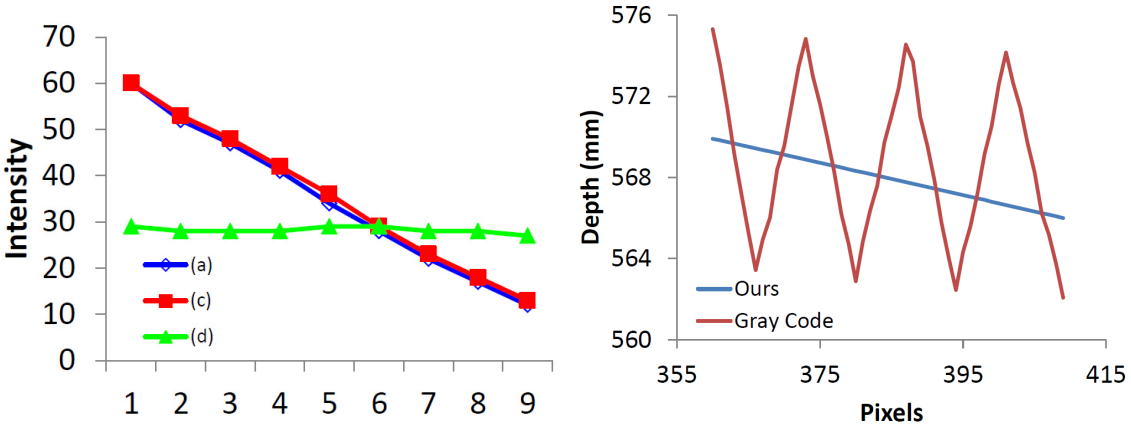
4.1.2 Minimum Stripe Width Determination

The key idea can be described as follows. Assume that there is no blurry effects in $I(T_{1:4})$ at all. As a result, $I(T_{1:4})$ are used as reference images and an error function is applied to each $I(t_{5:9})$ to measure the degree of blurriness. Whenever the error for a certain $I(t_{5:9})$ is larger than a threshold, the corresponding pattern $P(t_{5:9})$ is discarded because it causes blurry effects.

Denote the intensity of a 1D window w around any bit-change position in the patterns as $P_w(t_{1:9})$, and an example is $P_w(t_{1:9}) = \{255, 255, 0, 0\}$. According to Eq. 4.1, the intensity of a 1D window w arround the bit-change in the captured image which is $I_w(t_{1:9}) = P_w(t_{1:9}) * PSF_p \cdot R * PSF_c$ remains constant if R is unchanged. This phenomenon is demonstrated in Figure 4.4. Figure 4.4(a) 4.4(c) 4.4(d) are the captured images $I(2), I(6), I(9)$, respectively. The scene contains a planar object and the projector is defocused. Figure 4.4(b) is the decoded bit for $I(2)$ where the red arrow is pointing at a 1D window around a bit-change position. The 1D window at the same position in $I(2), I(6), I(9)$ is zoomed in and attached next to itself. The intensity curve of each 1D window is shown in Figure 4.4(e). Each 1D window is colored the same as its intensity curve. Observe that the intensity of the 1D window from different images is similar to each other under the condition that there is no blurring inside the window. That is, the red curve is similar to the blue one, but the green curve is not similar to either one. The assumption of the new is that the surface reflectance R is constant inside these windows used for comparison. It is noteworthy that an important property of binary code patterns is that the bit-change in the low



(a) (b) (c) (d)



(e) (f)

Figure 4.4: (a, c, d) $I(2)$, $I(6)$, $I(9)$ and selected 1D windows, respectively. (b) Decoded bit for $I(2)$. (e) Intensity curves of the 1D windows. (f) Depth obtained by the new method and by using gray code patterns, for the red dotted line in (a).

frequency patterns always re-appear at the same position in the higher frequency patterns. Therefore, the captured images has a similar property. For example, a bit-change in $I(2)$ always re-appears at the same position in $I(3) \sim I(9)$, and the surface reflectance R is always the same at the same scene point. With this useful observation, the new method uses those 1D windows around the bit-changes in $I(T_{1:4})$ as reference and estimates the degree of blurriness caused by short-range effects in

$I(T_{5:10})$. The details are described as follows. Assume that the decoded bit has been obtained for captured images $I(T_{1:9})$. The new method sets a 1D window $w(t_{1:4})$ centered at the position of every bit-change in $I(t_{1:4})$. The window size is initialized to be $|w(t_{1:4})| = 9$ pixels. After that, the intensity of each $w(t_{1:4})$ is compared to the 1D window $w(t_{5:9})$ at the same position in the image $I(t_{5:9})$. Notice that the size of the window should be adjusted if necessary to guarantee that there is no more than one bit change inside $w(t_{5:9})$. The reason is that the comparison is meaningless if there is one bit-change in $w(t_{1:4})$ but more than one in $w(t_{5:9})$. After that, the 1D window is flipped reversal. if the bit-change inside the window is $0 \rightarrow 1$. The rotation guarantees the bit-change inside the 1D windows is consistent and therefore the comparison between them is meaningful. The dissimilarity between $w(t_{1:4})$ and $w(t_{5:9})$ is measured by

$$e = \frac{\sum_{i=-\frac{|w(t_{5:9})|}{2}}^{\frac{|w(t_{5:9})|}{2}} |I(x+i, y, t_{5:9}) - I(x+i, y, t_{1:4})|}{255 \times |w(t_{5:9})|}. \tag{4.2}$$

e is smaller if the intensity of two windows are closer. After e is obtained for each 1D window in $I(t_{5:9})$, they are averaged and stored as $e(t_{5:9})$. If $I(t_{5:9})$ is not blurred, then $e(t_{5:9})$ should be smaller comparing to that of when $I(t_{5:9})$ is blurred. For example, $e(6)$ for $I(6)$ shown in Figure 4.4(c) is 0.0154, and $e(9)$ for Figure 4.4(d) is 0.0605. A threshold δ is set to be 0.03. $e(t_{5:9})$ is compared to δ, and the pattern $P(t_{5:9})$ should be discarded if $e(t_{5:9}) > \delta$. δ is set conservatively. More patterns will be discarded if δ is smaller. The range of $[0.03, 0.045]$ is suitable for δ through the experiments.

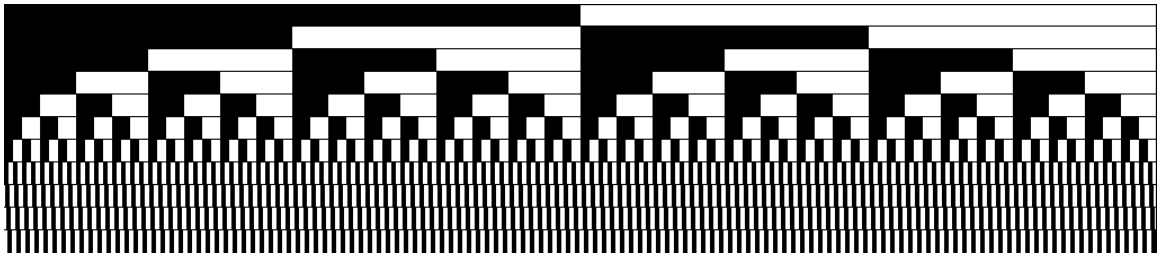


Figure 4.5: The newly designed patterns.

Assume that $P(t_m)$ is the pattern with the minimum stripe width after some patterns are discarded by the new method, then $P(t_m)$ is shifted to create new patterns so that every projector pixel can be uniquely encoded. Each new pattern is obtained by shifting $P(t_m)$ to the right one pixel at a time. In total, $P(t_m)$ is shifted $SW(t_m) - 1$ times. An example is shown in Figure 4.5. In particular, $P(9)$ is discarded by the new method due to blurry effects in $I(9)$. After that, $P(8)$ is shifted 3 times to the right in order to encode every projector pixel. By projecting this set of patterns, the short-range effects are eliminated in the captured images. Figure 4.4(f) compares the depth obtained by using the newly generated patterns and gray code patterns, for the portion that is colored by the red dotted line in Figure 4.4(a). This graph indicates that the result of the new method is better because the object is a planar board.

One could argue that the bit-change in $I(t_{1:4})$ may be due to inter-reflection, which is true. To solve that, the new method presented in Section 4.2.1 detects the bit-changes caused by inter-reflection in $I(t_{1:4})$, and the 1D window around these bit-changes will not be used as reference.

4.2 Long-range Effects

Long-range effects are usually caused by diffuse and specular inter-reflection. Denote the new patterns as $P'(t'_{1:N})$ and the intensity of captured images as $I'(t'_{1:N})$, where the number of patterns is N. A symbol with $'$ is associated with the new patterns. Suppose M patterns are not shifted. Then $P'(t'_{1:M})$ is termed "unshifted patterns" and $I'(t'_{1:M})$ "unshifted images," while $P'(t'_{M+1:N})$ "shifted patterns" (such as the bottom four in Figure 4.5) and $I'(t'_{M+1:N})$ "shifted image." The basic procedure of the new method is to detect the errors caused by inter-reflection whose details are described in the following subsections, and iteratively corrects the error by reducing the incident illumination of the projector. More details about the procedure are given at the end of this section together with Figure 4.7.

The new method casts the problem of detecting inter-reflection errors in the captured images into a labeling problem. That is, each pixel is assigned a label (either consistent or inconsistent) and a cost is associated with each possible labeling. A local winner-take-all optimization is applied to minimize the cost and obtain a label for each pixel.

4.2.1 Unshifted Images

An important property that is embedded in the binary code patterns is used. In particular, a bit-change in a pattern always re-appear at the same position in the patterns that have higher frequencies. As illustrated in Figure 4.6, the bit-change colored red in $P(1)$ re-appears at the same position in $P(T_{2:8})$. The bit-changes colored green

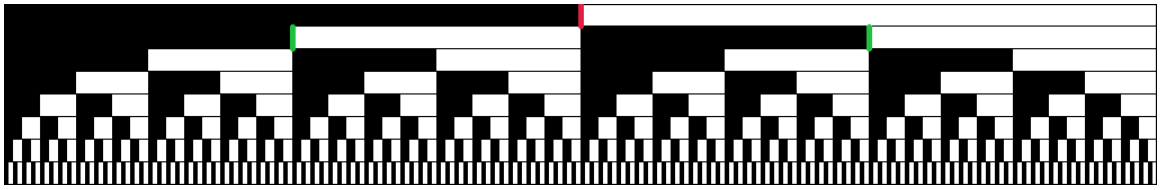


Figure 4.6: The properties embedded in binary code patterns. See text for details.

have this property as well. Due to image noise, two bit-changes at the same position in two patterns may not appear at the same position in two corresponding images. However, their positions should be very close. In the presence of inter-reflection, such a property may not hold in the captured images and can be used to detect and correct errors.

For each bit-change at position (x, y) in $I'(t'_{1:M})$, the spatially closest bit-change is obtained in every $I'(t'_i)$ where $t'_{1:M} < t'_i \leq M$. The bit-change in $I'(t'_i)$ must be on the same horizontal scanline y. Assume that the closest bit-change in $I'(t'_i)$ is at position (x'_i, y), then the distance between these two bit-change positions is $|x'_i - x|$. This distance is obtained for each $I'(t'_i)$. After that, the distances are averaged and the average is denoted as $d'(x, y, t'_{1:M})$. One can infer that $d'(x, y, t'_{1:M})$ should be smaller without the presence of inter-reflection. Denote $l_p = \{con, inc\}$ where $con(inc)$ is a label to pixel p to denote that it is consistent or inconsistent. The corresponding cost for each label is defined as

$$C_p(con) = \begin{cases} 0 & \text{if } b'(x, y, t'_{1:M}) = b'(x-1, y, t'_{1:M}) \\ 2^{d'(x,y,t'_{1:M})} & \text{otherwise} \end{cases} \tag{4.3}$$

$$C_p(inc) = \begin{cases} \infty & \text{if } b'(x,y,t'_{1:M}) = b'(x-1,y,t'_{1:M}) \\ \frac{W}{d'(x,y,t'_{1:M})+\epsilon} & \text{otherwise.} \end{cases} \tag{4.4}$$

$b'(x, y, t'_{1:M})$ is the decoded bit for a pixel (x, y) in $I(t'_{1:M})$. W represents the width of the captured images, and $\epsilon = 1 \times 10^{-5}$ prevents division by 0. The designed cost is explained as follows. $b'(x, y, t'_{1:M}) = b'(x-1, y, t'_{1:M})$ means that there is no bit-change at (x, y) in $I(t'_{1:M})$, which means this pixel should be consistent. The reason is that the errors due to inter-reflection always happen at the bit-change positions. As a result, the cost for assigning *con* to a pixel that has no bit-change should be 0, and ∞ for assigning *inc*. When bit-change occurs at (x, y) in $I(t'_{1:M})$, whether it is consistent or not depends on $d'(x, y, t'_{1:M})$. In particular, when $d'(x, y, t'_{1:M})$ is larger, the bit-change is more likely to be inconsistent due to inter-reflection. Therefore, the cost of assigning *inc* to this pixel is smaller and of assigning *con* to it is larger.

4.2.2 Shifted Images

The presented method detects errors based on the fact that $P'(t'_{M+1:N})$ is consistently shifted based on $P'(t'_i)$ where $t'_i \in T'_{M+1:N} \setminus t'_{M+1:N}$. The assumption is that the effect of inter-reflection causes inconsistency to such shift. The details are as follows. When a bit-change occurs at (x, y) in $I(t'_{M+1:N})$, its spatially closest bit-changes (x', y) are obtained for all t'_i according to the shifting direction. The total amount of shift for each pixel is computed as

$$u'(x, y, t'_{M+1:N}) = \sum_{t'_i \in T'_{M+1:N} \setminus t'_{M+1:N}} |x' - x|. \tag{4.5}$$

The average $\bar{u}'(t'_{M+1:N})$ is obtained for $I'(t'_{M+1:N})$ by

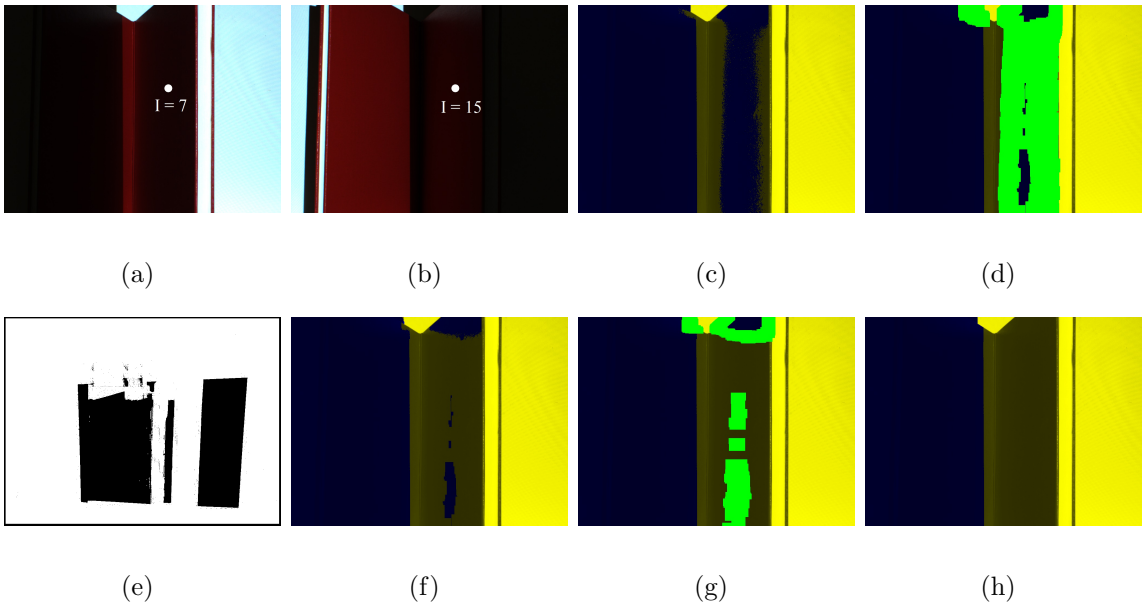
$$\bar{u}'(t'_{M+1:N}) = \frac{\sum_{(x,y)\in I'(t'_{M+1:N})} u'(x, y, t'_{M+1:N})}{W \times H}. \tag{4.6}$$

The W, H in Eq. 4.6 are the width and height of the images, respectively. The amount of shift $u'(x, y, t'_{M+1:N})$ is expected to be similar for all $(x, y) \in I'(t'_{M+1:N})$ with no inter-reflection. Therefore, $u'(x, y, t'_{M+1:N})$ should be close to $\bar{u}'(t'_{M+1:N})$. Finally the cost is defined as

$$C_p(con) = \begin{cases} 0 \quad \text{if } b'(x, y, t'_{M+1:N}) = b'(x-1, y, t'_{M+1:N}) \\ |u'(x, y, t'_{M+1:N}) - \bar{u}'(t'_{M+1:N})| \quad \text{otherwise} \end{cases} \tag{4.7}$$

$$C_p(inc) = \begin{cases} \infty \quad \text{if } b'(x, y, t'_{M+1:N}) = b'(x-1, y, t'_{M+1:N}) \\ \frac{W}{|u'(x,y,t'_{M+1:N}) - \bar{u}'(t'_{M+1:N})| + \epsilon} \quad \text{otherwise.} \end{cases} \tag{4.8}$$

The design of the cost is similar to that of Eq. 4.3 and 4.4. When $b'(x, y, t'_{M+1:N}) = b'(x-1, y, t'_{M+1:N})$ which means that there is no bit-change, the pixel should be consistent. When a bit-change occurs, the cost depends on $|u'(x, y, t'_{M+1:N}) - \bar{u}'(t'_{M+1:N})|$. The cost is minimized by applying a local winner-take-all optimization to one image at a time. In particular, label *con* is assigned to pixel p if $C_p(con) < C_p(inc)$. Global optimization is not applied and the reason is described as follows. Whether a pixel is consistent or not should not have any impact to its neighboring pixels, which means that there is no smoothness prior in the labeling problem. Moreover, a global optimization without any smoothness prior is equivalent to a local winner-take-all optimization.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.7: (a, b) Captured images with $P(1)$ and its inverse pattern. (c) Decoded bit for 4.7(a). (d) The regions colored pure green are error regions detected by the new method. (e) Pattern mask. The black regions are correspondences from image pixels that are in the no-error regions. In the next iteration, this mask is applied to all the projected patterns. (f) Newly decoded bit for 4.7(a). Compare to 4.7(c), decoded bit are changed in the error regions only. (g) Error regions detected in 4.7(f). (h) Final decoded bit where there is no error region detected.

An example of detecting and correcting long-range effects are shown in Figure 4.7, where 4.7(a) is the captured image with pattern $P(1)$ and 4.7(b) with its inverse pattern. The scene contains two books organized in a Vshape. The two white dots are at the same position and their intensities indicate that inter-reflection exists in the scene. Figure 4.7(c) is the decoded bit for 4.7(a) and one can see that the region with error is large. 4.7(d) indicates the error region (colored green) detected by the new

method. In particular, the new method labels certain pixels in 4.7(c) as inconsistent bit-changes. After that, a window with size 50×50 pixels is centered at each pixel labeled inconsistent. These windows are the regions where error exists. Then the new method find correspondences for the image pixels that are not considered as error which are the pixels that are not colored green in 4.7(d). The black regions in 4.7(e) are the correspondences found from the error-free regions in the image. These regions are masked to reduce the incident light so that the long-range effects can be reduced. The masked patterns are projected to the scene, and the decoded bits are updated for each image but only for the regions that are colored green in 4.7(d). The new decoded bits are shown in 4.7(f) and one can see that the bits in those green regions are correct now. One could argue that the 50×50 windows may not be big enough to cover all the error regions. This is true and is indicated in 4.7(d). The new method addresses this problem by iteratively corrects the errors. In particular, when a new decoded bit (4.7(f)) is obtained, the new method is applied again to detect the inconsistent bit-changes. The error regions are colored pure green in 4.7(g) and one can see that the error regions are gradually reduced. In the 4th iteration, no more error regions are detected and the new method converges. The final decoded bits are shown in 4.7(h). One can see that the new method successfully corrects the errors caused by long-range effects. It is noteworthy that the above method to correct the long-range effects is heuristic and iterative. While there is no guarantee that it would converge, it converges in all the experiments described in this thesis.

4.3 Implementation Details

A step-by-step description of the new method is provided first and some details are explained later.

In step 1, the inconsistent bit-changes are detected in $I(T_{1:4})$ and the intensity of the windows around these bit-changes will not be used as reference. Therefore, it is not needed to correct the inconsistent bit-changes in step 1.

The implementation captures two additional images by projecting fully white/black patterns to remove the ambient lighting. These two images are obtained before the entire **Algorithm** 1 starts.

It is well known that binary code may produce more noise in the correspondences comparing to gray code. In order to reduce the noise, median filtering is applied (step 6).

4.4 Experimental Results

The new method is applied to many real-world scenes with different characteristics. Both quantitative and qualitative evaluation are provided to demonstrate the accuracy and robustness of the new method. The experimental setup includes a Point Grey flea2 camera and a Dell M110 projector, both are with resolution 1024×768.

The first experiment is designed using a scene which includes a ceramic bowl placed in a small carton (named "Bowl Scene"). The bowl has texture on its surface. Five datasets are captued, and the projector focus setting is purposely adjusted so

Algorithm 1 Scene Adaptive Structured Light

1. Detect any bit-change that is caused by inter-reflection in $I(T_{1:4})$ based on Eq. 4.3 and 4.4. Only $I(T_{1:4})$ are used because these are the reference images used in Step 2. Details of this step are described in Section 4.2.1.

2. Compute dissimilarity $e(t_{5:9})$ based on Eq. 4.2, for each image $I(t_{5:9})$ using bit-changes in $I(T_{1:4})$ as reference. Notice that the bit-changes detected in Step 1, which are due to inter-reflection, are not used as reference. Determine whether an image $I(t_{5:9})$ is blurred or not based on $e(t_{5:9})$ and discard its corresponding pattern $P(t_{5:9})$. Design new patterns $P'(t'_{1:N})$ by shifting $P(t_m)$ which is the pattern with the minimum stripe width after some patterns are discard. Figure 4.5 is an example of the new patterns. Details are described in Section 4.1.

3. Project new patterns $P'(t'_{1:N})$. Label each pixel in the image $I'(t'_{1:N})$ to be either consistent or inconsistent by Eq. 4.3, 4.4 and 4.7, 4.8 with winner-take-all optimization. Place a 50×50 pixels window centered at each pixel that is labeled inconsistent. These windows are the error regions due to inter-reflection. Details are described in Section 4.2.

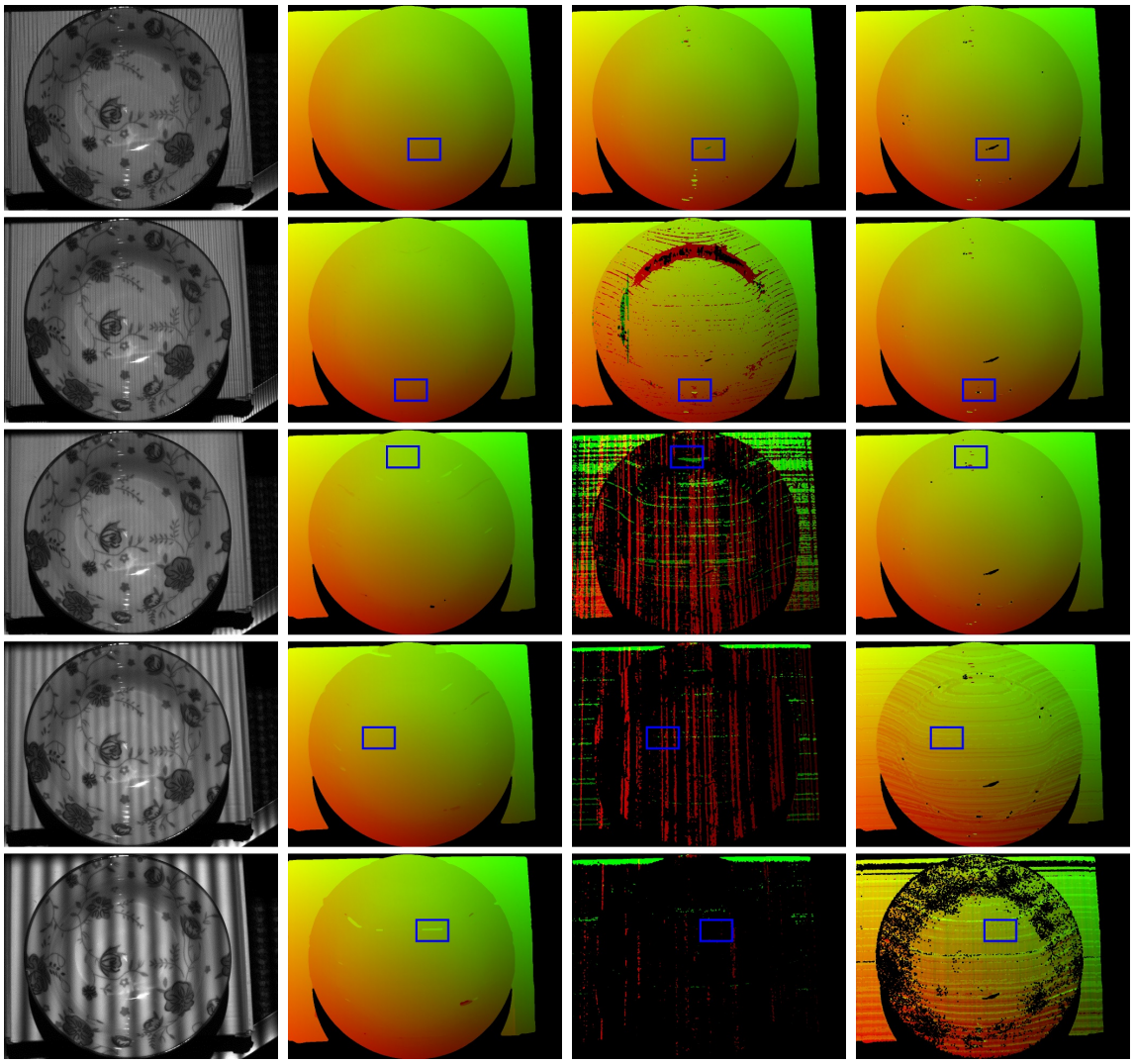
4. Obtain correspondences for pixels with no errors. Mask correspondences in the patterns to reduce the amount of incident light. Generate new patterns with the mask (such as Figure 4.7(e)).

5. Repeat step 3 and 4 until converge.

6. Run post-processing.

that the blurry effects are different among them. Each row of Figure 4.8 shows the results of one dataset and one can see that the blurry effects become more severe from top to bottom. In particular, the five rows correspond to the results when patterns with stripe width = 2, 4, 8, 16, 32 pixels are blurred, respectively. Under these experimental settings, both short- and long-range effects are present simultaneously in some regions of the bowl.

The new method is compared with [35] and [61], which consistently outperform traditional methods such as gray code and phase shifting. The publicly available Matlab code provided on the author's webpage is used. The correspondence map is used for comparison with red and green used for the x and y coordinates, respectively. A black pixel in the correspondence map indicates that the correspondence cannot be recovered for that pixel. The method presented in [35] fails for the last three rows because this method has an implicit assumption that stripe width ≥ 8 cannot be blurred. Although this method is suppose to work for row 1 and 2, there are still errors in the results due to strong inter-reflection. Comparing with [35], the new method can handle stronger inter-reflection which is indicated in row 1 and 2, and certain extreme cases of defocusing where [35] fails (indicated in row 3, 4 and 5). It is noteworthy that the Micro Phase Shifting method [61] requires the user to select the frequency of the band. The authors of [61] recommend using the low frequency band such as 32 pixels wide patterns when the defocus is strong. Therefore, the frequency to be 32 pixels is selected and the number of frequencies is set to be 15 throughout these experiments. Moreover, this method also requires the projected



(a) (b) (c) (d)

Figure 4.8: Experimental results for the "Bowl Scene". Each row shows the results of a dataset. (a) One of captured images. (b-d) Final correspondence map by the new method, [35] and [61], respectively. Red and green are used for the x and y coordinates, respectively.

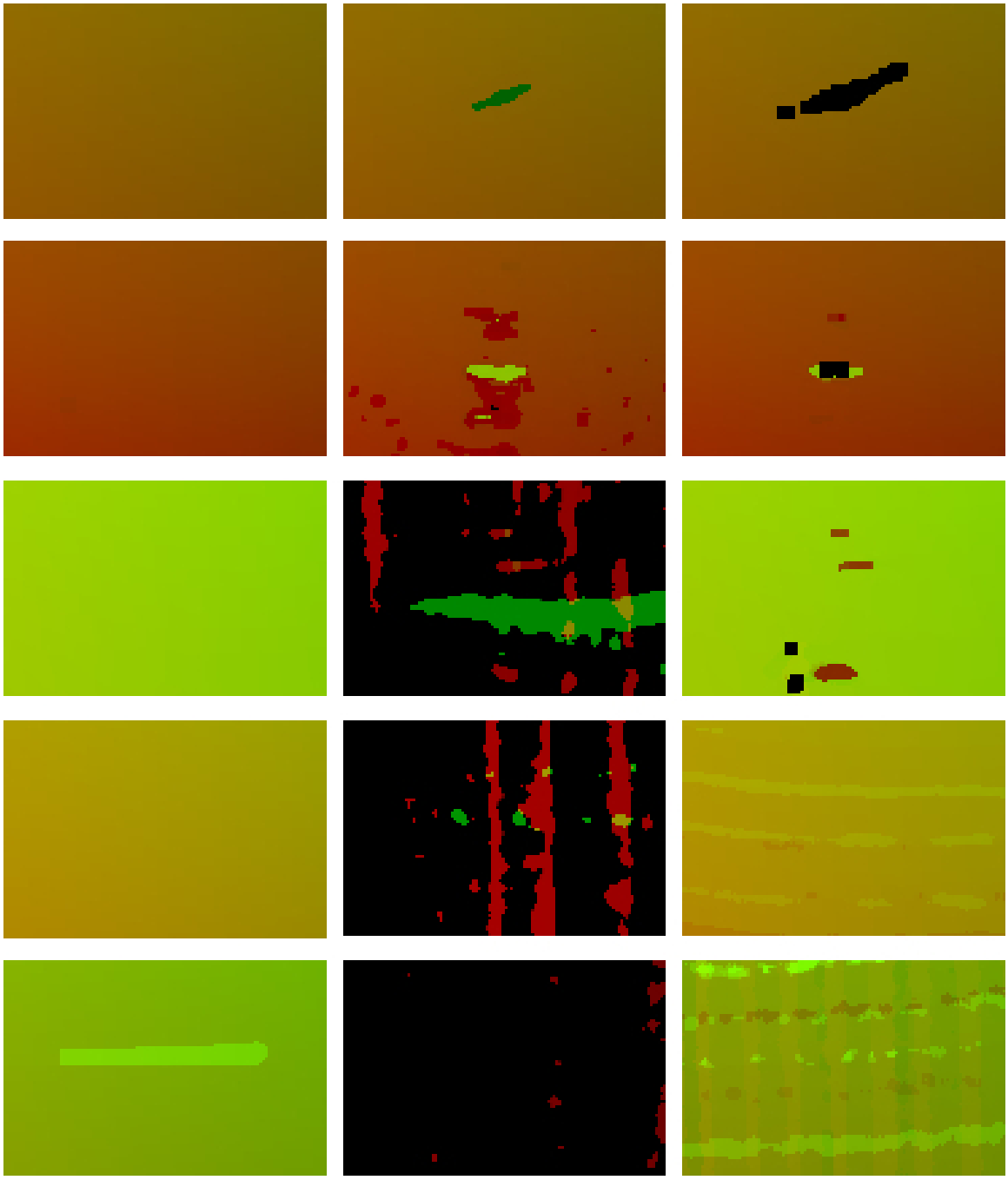
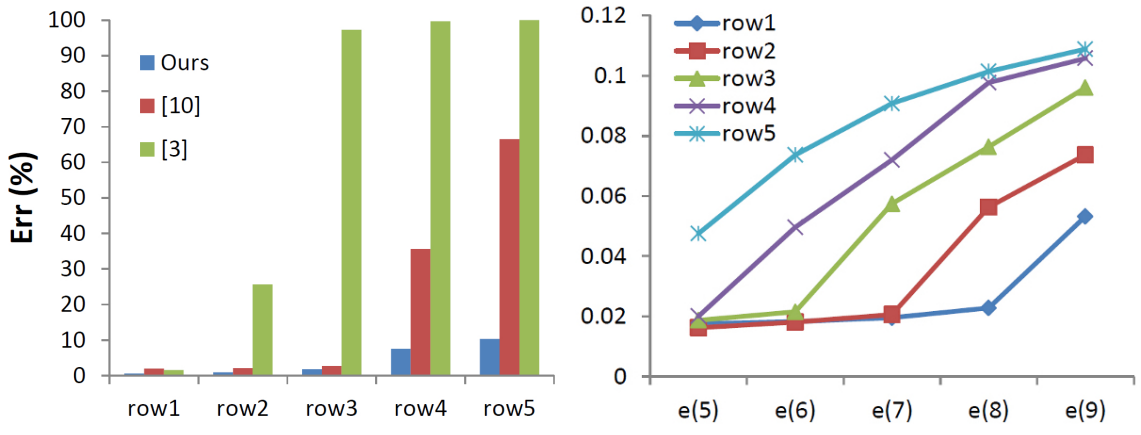


Figure 4.9: Comparing details of correspondence maps in Figure 4.8.

images to be radiometrically calibrated to account for projector's non-linear response. The method presented in [82] is applied for radiometric calibration. Since [61] uses monochrome patterns, gray scale calibration is performed only. As well, calibration is applied whenever the setting of the projector or camera is changed. Certain regions in the correspondence maps are selected and further compared in Figure 4.9. The selected regions are indicated by blue rectangles and they are all on the surface of the ceramic bowl where inter-reflection and projector defocus exist at the same time. Since the surface of the bowl is smooth, the correspondence map in these regions should not have any sharp discontinuity. One can see that the results of the new method are smooth except the one in the last row. However, the results by [35] and [61] have sharp discontinuities which indicate errors caused by either inter-reflection or defocus, or both. The comparison indicates that the results of the new method are better than both [35] and [61], especially with strong blurry effects. The disadvantage of our method is that it requires an iterative process that corrects the error caused by inter-reflection, which results in large number of captured images. In particular, the number of captured images for [35] is always 80, and 17 for [61]. However, the number of iterations for our method is typically 5-6, which requires 100-120 images.

Besides comparing the final correspondence map, quantitative accuracy evaluation is provided for this set of experiments. To be more specific, the ground truth is obtained by manually determining the decode bit. After that, the results of the new method and that of [61] are compared to the ground truth using the following error



(a) (b)

Figure 4.10: (a) Comparison of the error (in percent) among the new method, [61] and [35]. The x axis is the row index in Figure 4.8. (b) $e(t_{5:9})$ is obtained by the new method described in Section 4.1.

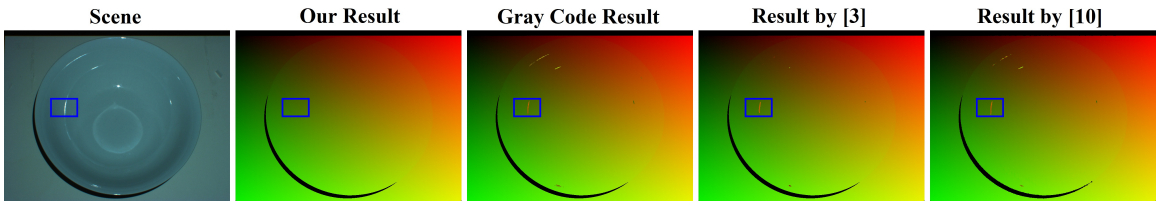
measure

$$Err = (num/NUM) \times 100\%. \tag{4.9}$$

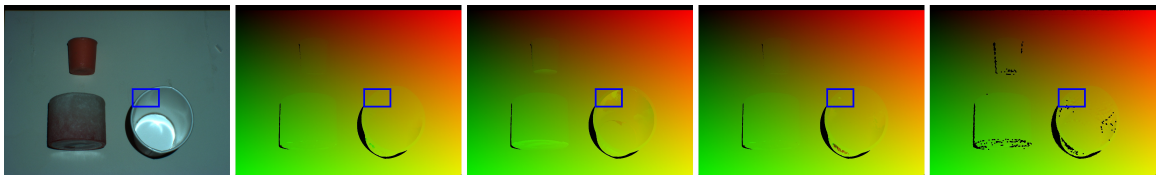
NUM is the number of recovered correspondences in the ground truth, and num the number of pixels whose correspondences are different from the ground truth. Figure 4.10(a) shows the comparison of the error among the new method, [61] and [35]. This graph demonstrates that the new method is better in accuracy. In particular, the error is always lower than that of both [35] and [61]. One can see that very large error exists in the last three rows for [35]. The reason is that this method assumes that patterns with stripe width ≥ 8 pixels cannot be blurred. Starting from row3, the pattern with stripe width $= 8$ pixels is blurred, and hence the method fails.

The result of minimum stripe width determination is an error $e(t_{5:9})$ for each

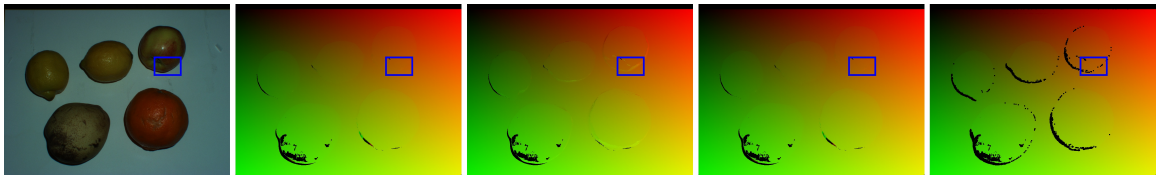
$I(t_{5:9})$. The error is shown in Fig 4.10(b). After $e(t)$ is obtained for $t \in [5, 10]$, the new method discards certain pattern(s) and designs new patterns such as those shown in Figure 4.5.



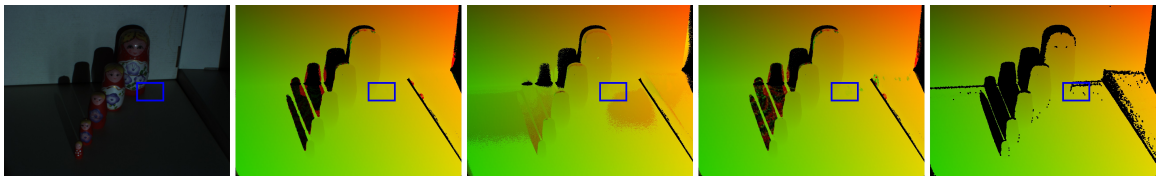
(a) Ceramic Bowl: strong inter-reflection



(b) Candle and Paper cup: subsurface scattering and inter-reflection.



(c) Fruit: subsurface scattering



(d) Russian dolls: projector defocus

Figure 4.11: Comparison of results produced by different method: the new method, conventional gray code, [35] and [61].

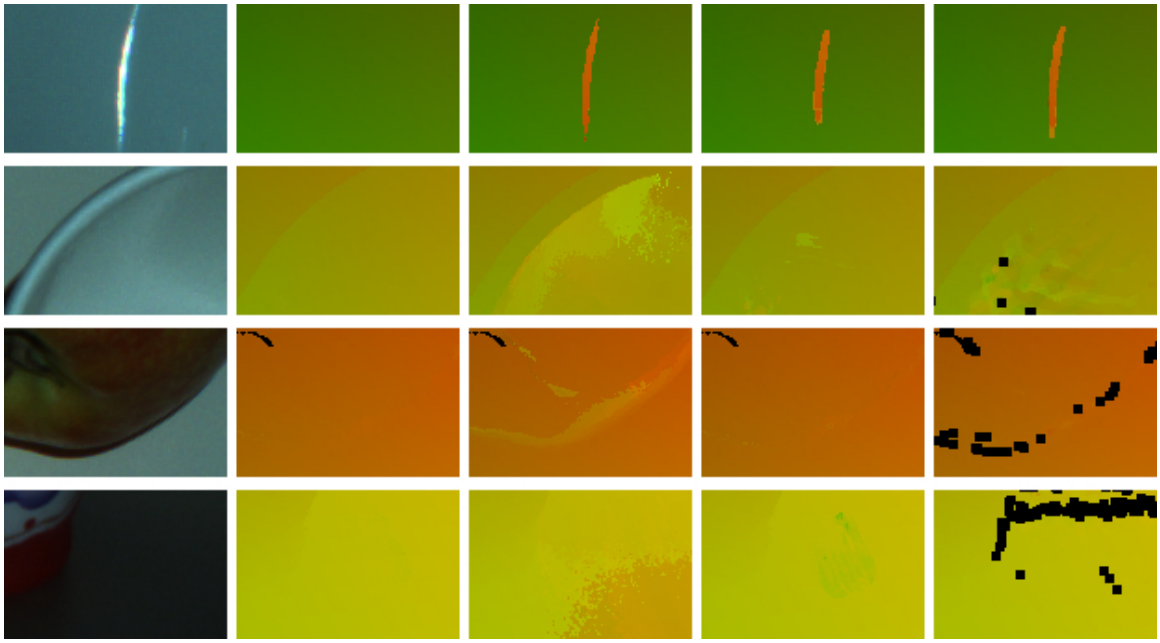


Figure 4.12: Comparing details of correspondence maps in Figure 4.11.

The new method is applied to many other scenes and qualitative evaluation is provided to demonstrate its robustness. The experimental results are shown in Figure 4.11. The experiments include scenes which have different kinds of global illumination effects, and with different characteristics such as texture. The final correspondence maps are used for comparison. Similar to Figure 4.8, red and green are used for the x and y coordinates, respectively. A black pixel indicates that the correspondence cannot be recovered for that pixel. The results produced by the new method are compared with that by the conventional gray code method, [35] and [61]. Some regions are selected and their correspondence maps are compared in Figure 4.12. In these experiments, the frequency is set to be 16 pixels and the number of frequencies to be 8 when applying [61]. In Figure 4.11(d), the smallest doll is 65cm from the projector

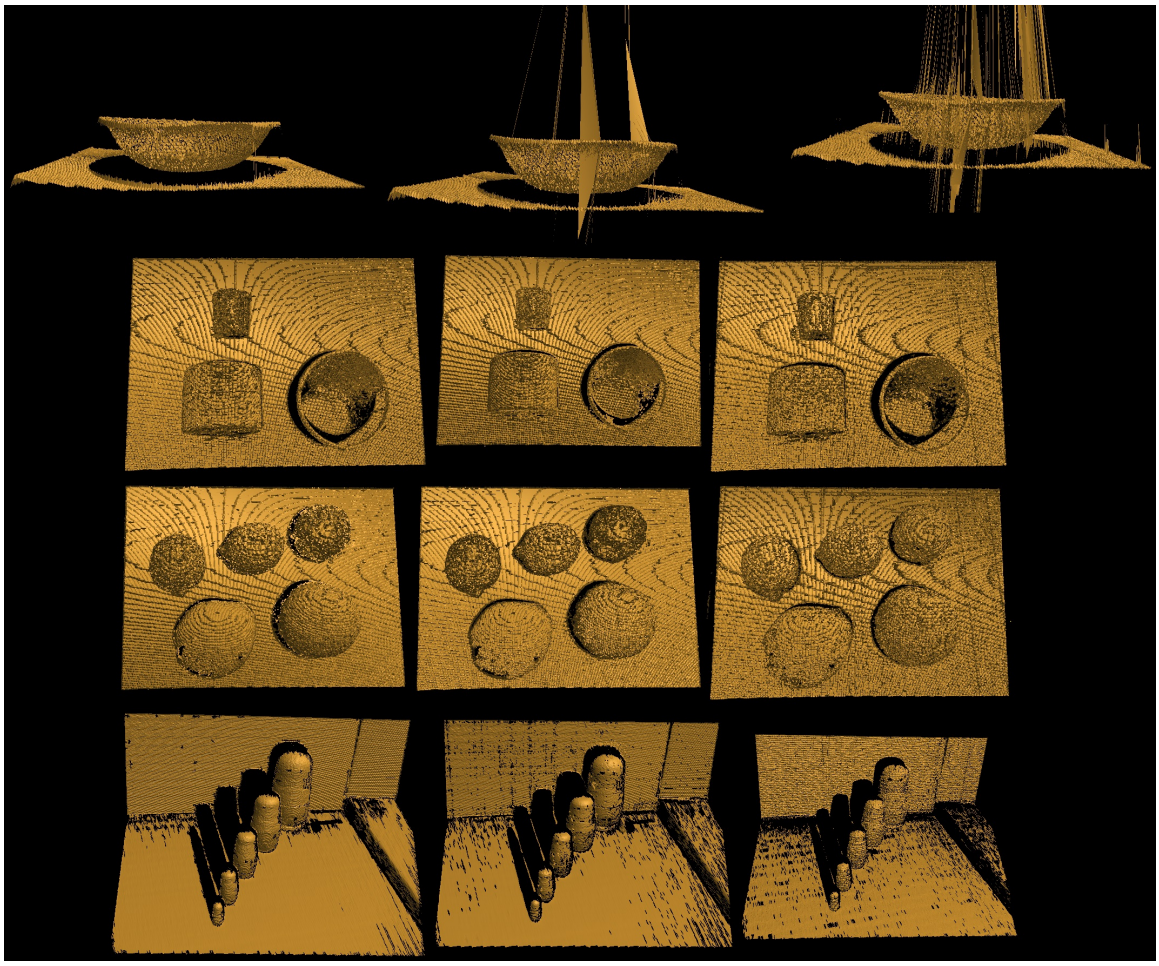


Figure 4.13: Comparison of 3D models recovered from correspondence maps produced by different method: the new method (left column), [35] (middle column) and [61] (right column).

and the largest doll 85cm. Therefore, the projector defocus is quite significant. One can see that there are still some artifacts in the Paper cup dataset when applying [61]. This is due to the frequency selection. Once the number of frequencies is changed from 8 to 15, the result becomes much better. That is, the selection of

frequency can impact the accuracy of the results. The 3D models reconstructed from the correspondence maps are compared and shown in Figure 4.13. In particular, the results of the new method are compared with that of [35] and [61]. The results demonstrate that the new method consistently outperforms the conventional gray code method. In the cases where inter-reflection is severe (Figure 4.11(a)), the new method produces better results comparing with that of both [35] and [61].

Quantitative analysis is provided for the 3D reconstruction results shown in Figure 4.13. The evaluation is performed as follows. The bit for the input images is manually determined for all the datasets shown in Figure 4.11 so that the ground truth for the correspondences can be obtained. For each pair of correspondences between the camera and the projector, a 3D point can be obtained by triangulation. After that, the Euclidean distance between every 3D point computed from the ground truth correspondences and that using the correspondences by the new method is calculated and averaged. Denote the averaged distance by ΔX and one can see that it represents the error for the created 3D model. The same error measurement is used for the gray code method, as well as for methods presented in [35] and [61]. Table 4.4 shows the size of the object, as well as ΔX for all the datasets shown in Figure 4.11 and the unit for the numbers in the table is mm. One can see that the error of the new method is always smaller than the other methods, although both [35] and [61] provide reasonable results. Notice that in "Ceramic Bowl", although there are a few 3D point whose error is large, the overall 3D error is still small because ΔX denotes the averaged error and the number of correspondences is huge.

	Object Size	Proposed Method	[35]	[61]	Gray Code
Figure 4.11(a)	400	0.9483	1.5893	1.7969	2.2003
Figure 4.11(b)	400	1.3635	1.3930	1.5515	2.7528
Figure 4.11(c)	400	0.9842	1.0850	1.1758	1.6672
Figure 4.11(d)	750	1.5906	2.4149	1.9812	123.2214

Table 4.1: The 3D error ΔX for all the datasets in Figure 4.11. The unit for the numbers is mm.

Chapter 5

Camera Housing Calibration

In the previous chapters, two new structured light methods are presented to establish correspondences between two camera views. For a camera system that is placed in air, the triangulation method shown in Figure 1.1(b) is used to compute the 3D points from the established correspondences. However, the method does not work when the camera system is deployed undersea. The reason is demonstrated in Figure 5.1. One can see that when refraction exists, the light path from the object point to the camera is not a straight line anymore. In this case, the resulting 3D point is incorrect if the in-air triangulation method is applied as shown in Figure 1.1(b), which is the intersection of those two green dashed lines. In other words, the in-air triangulation method must be modified to accommodate for the refraction effect.

In this chapter, a novel refractive housing calibration method is presented for an underwater stereo camera system where both cameras are looking through multiple parallel flat refractive interfaces, as the one shown in Figure 5.1. At the heart of this

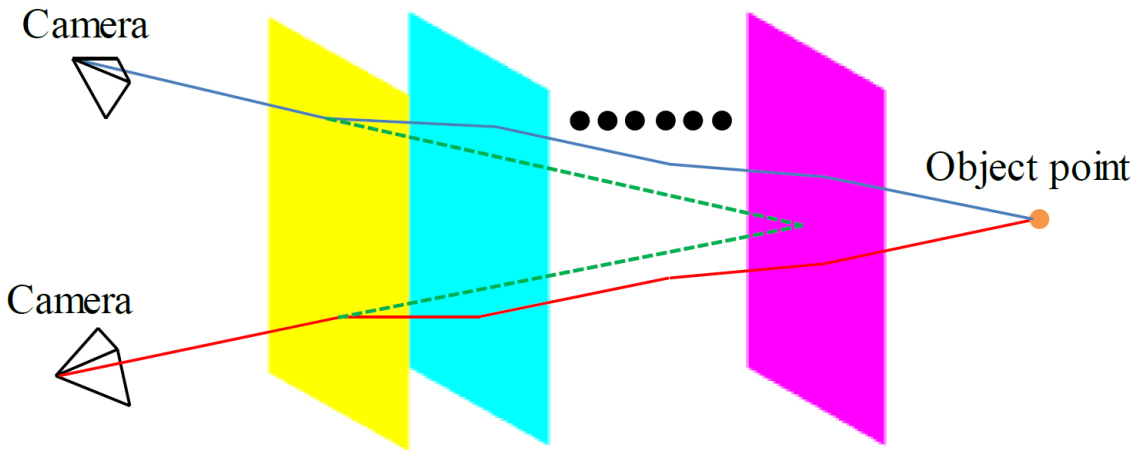


Figure 5.1: Demonstration that the typical triangulation method does not work for an underwater camera system.

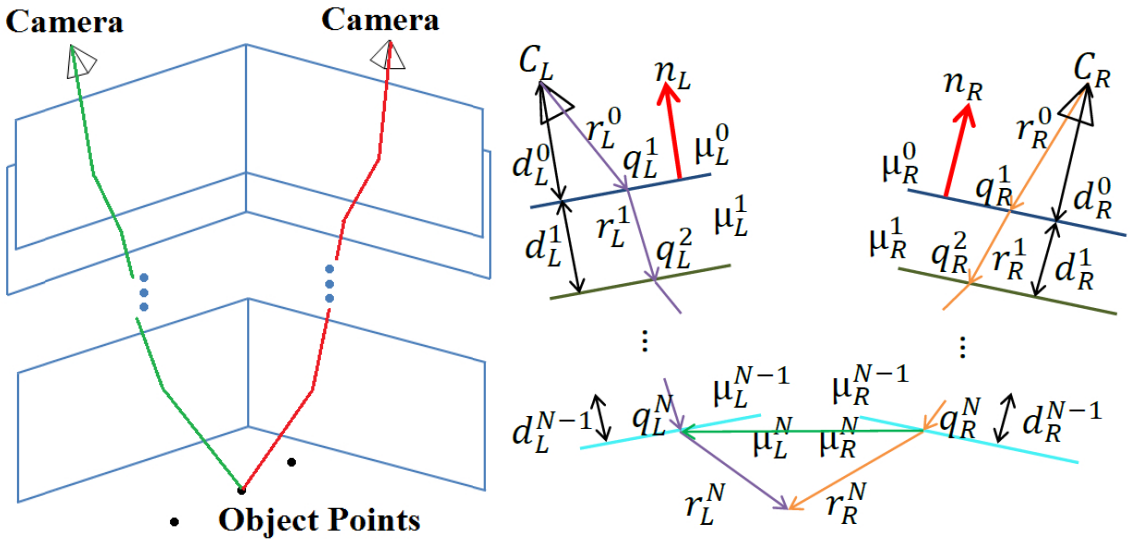
method is an important finding that the thickness of the interface can be estimated from a set of pixel correspondences in the stereo images when the refractive normal is given. Besides that, another novel housing calibration method is presented for a single underwater camera by making full use of triple wavelength dispersion. This method is based on another important finding that there is a closed-form solution to the distance from the camera center to the refractive interface once the refractive normal is known. The correctness of this finding is mathematically proved. As well, the refractive normal can be estimated by solving a set of linear equations using dispersion. Both important findings have not been studied or reported by other researchers. It is noteworthy that the new methods do not require any calibration target such as a checkerboard pattern which may be difficult to manipulate when the cameras are deployed deep undersea. The implementation of both methods is

simple. In particular, they mainly require solving a set of linear equations of the form $Ax = b$. Extensive experiments have been carried out which include simulations to verify the correctness of both methods as well as to test their robustness to noise. The results of real experiments show that the new methods work as expected. The accuracy of the results are evaluated against the ground truth in both simulated and real experiments. Finally, dispersion is used to compute the 3D shape of an object using one single camera.

5.1 Stereo Camera Housing Calibration

Without loss of generality, let's assume that each camera is facing the refractive interface of its own underwater housing. Notice that the two cameras sharing the same interface is a special case. Moreover, assume that there is one or multiple refractive interfaces and they are all parallel to each other. Both 3D and 2D diagrams for this scenario are shown in Figure 5.2. The diagram shows two corresponding rays coming out from the camera centers, passing through multiple interfaces, and intersecting at an object point.

The symbols shown in Figure 5.2(b) are described first. The subscript "L" denotes the left camera view and "R" the right. Suppose there are N refractions in each camera view and the refractive indices μ_L^i and $\mu_R^i, i \in [0, N]$ are given. Here i is always an integer. The normals of the refractive interfaces are denoted as n_L and n_R. Assume that the orientations and positions of the two cameras are known by performing offline calibration in air. In other words, camera centers C_L and C_R



(a) (b)

Figure 5.2: (a) Flat refractive geometry with multi-layer of refractive interface for a stereo camera system. (b) 2D diagram for (a).

are provided. Moreover, assume that a set of corresponding rays coming out of the camera centers has been established, which can be easily done by SIFT/SURF feature matching. For example, $\{r_L^0, r_R^0\}$ is a pair of corresponding rays and there are M pairs. The directions of rays in each refractive interface are denoted as r_L^i and r_R^i, and the rays intersect each interface at points q_L^i and q_R^i. The thickness of the left layer is d_L^i, and of the right $d_R^i, i \in [0, N-1]$. To sum up, C_L, C_R and M pairs of corresponding rays such as $\{r_L^0, r_R^0\}$, and the refractive indices μ_L^i, μ_R^i are assumed to be known. The new method solves for n_L, n_R, d_L^i and $d_R^i, i \in [0, N-1]$ simultaneously. Once the results are obtained, the 3D geometry of the scene can be reconstructed by ray-tracing from the correspondences.

5.1.1 Given refractive normal, estimate layer thickness

Assume that n_L and n_R are given, which will be relaxed later in Section 5.1.2. Let's demonstrate that d_L^i and d_R^i can be obtained by solving a set of linear equations once n_L and n_R are known. The new method is based on an important finding which is expressed as

$$\left(q_L^N - q_R^N\right) \cdot \left(r_L^N \times r_R^N\right) = 0. \tag{5.1}$$

In the equation, $\cdot$ denotes dot product and $\times$ cross product. The equation states that $q_L^N q_R^N$ is on the plane formed by the two rays r_L^N and r_R^N. One can see that this condition is always true. According to Snell's law, it can be derived that

$$q_L^i = q_L^{i-1} - d_L^{i-1} \frac{r_L^{i-1}}{n_L \cdot r_L^{i-1}}, i \in [2, N], \tag{5.2}$$

and q_R^i is obtained similarly. Furthermore,

$$q_L^1 = C_L - d_L^0 \frac{r_L^0}{n_L \cdot r_L^0} \tag{5.3}$$

and a similar equation holds for q_R^1. By substituting Eq. 5.2 and Eq. 5.3 to Eq. 5.1, Eq. 5.1 can be written as

$$\begin{aligned}
&\left(\left(C_L - d_L^0 \frac{r_L^0}{n_L \cdot r_L^0} - d_L^1 \frac{r_L^1}{n_L \cdot r_L^1} - \cdots - d_L^{N-1} \frac{r_L^{N-1}}{n_L \cdot r_L^{N-1}} \right) \right. \\
&\left. - \left(C_R - d_R^0 \frac{r_R^0}{n_R \cdot r_R^0} - d_R^1 \frac{r_R^1}{n_R \cdot r_R^1} - \cdots - d_R^{N-1} \frac{r_R^{N-1}}{n_R \cdot r_R^{N-1}} \right) \right) \cdot \left(r_L^N \times r_R^N\right) = 0.
\end{aligned} \tag{5.4}$$

Let $\mathcal{R} = \left(r_L^N \times r_R^N\right)$, $\mathcal{R}_L^i = \frac{r_L^i}{n_L \cdot r_L^i}$ and $\mathcal{R}_R^i = \frac{r_R^i}{n_R \cdot r_R^i}$. Eq. 5.4 can be further expanded as

$$\left(C_L - C_R - d_L^0 \mathcal{R}_L^0 - \cdots - d_L^{N-1} \mathcal{R}_L^{N-1} + d_R^0 \mathcal{R}_R^0 + \cdots + d_R^{N-1} \mathcal{R}_R^{N-1}\right) \cdot \mathcal{R} = 0. \tag{5.5}$$

Re-arrange the order of terms produces

$$
\begin{aligned}
&- d_L^0(\mathcal{R}_L^0 \cdot \mathcal{R}) - \cdots - d_L^{N-1}(\mathcal{R}_L^{N-1} \cdot \mathcal{R}) + d_R^0(\mathcal{R}_R^0 \cdot \mathcal{R}) \\
&+ \cdots + d_R^{N-1}(\mathcal{R}_R^{N-1} \cdot \mathcal{R}) = (C_R - C_L) \cdot \mathcal{R}.
\end{aligned} \tag{5.6}
$$

Finally Eq. 5.6 can be written as

$$
\begin{aligned}
&\Big(- (\mathcal{R}_L^0 \cdot \mathcal{R}), \ldots, -(\mathcal{R}_L^{N-1} \cdot \mathcal{R}), (\mathcal{R}_R^0 \cdot \mathcal{R}), \ldots, (\mathcal{R}_R^{N-1} \cdot \mathcal{R}) \Big) \\
&\Big(d_L^0, \ldots, d_L^{N-1}, d_R^0, \ldots, d_R^{N-1} \Big)^T = (C_R - C_L) \cdot \mathcal{R}.
\end{aligned} \tag{5.7}
$$

Once n_L and n_R are known, then r_L^i and $r_R^i, i \in [1, N]$ can be obtained since r_L^0 and r_R^0 are known, and the refractive indices are assumed to be given. Therefore, $\mathcal{R}_L^i, i \in [0, N-1]$ which is $\frac{r_L^i}{n_L \cdot r_L^i}$, can be calculated. Similarly, $\mathcal{R}_R^i$ and $\mathcal{R}$ can be computed as well. Up to this point, one can see that $\big(d_L^0, \ldots, d_L^{N-1}, d_R^0, \ldots, d_R^{N-1} \big)^T$ are the unknowns in Eq. 5.7 and all the others can be calculated. Notice that Eq. 5.7 has the same form as $Ax = b$ which can be easily solved. Given M pair of correspondences, the dimension of A is $M \times 2N$, b is $2N \times 1$ and x is $M \times 1$. By solving the set of linear equations, d_L^i and $d_R^i, i \in [0, N-1]$ are obtained.

5.1.2 Search space

The above section demonstrates that once n_L and n_R are provided, d_L^i and d_R^i can be computed by solving a set of linear equations. After that, the 3D geometry of a scene can be obtained as well. This method can be readily incorporated into an existing underwater stereo camera system where the normals of the refractive interfaces are

known. Unfortunately, some systems do not have information of the normal and the method presented in this section is designed to address this problem.

Although the accurate n_L and n_R are not known, they are both inside a search space which is the hemisphere shown in Figure 5.3. That is, the z component of the normal direction is always negative in the camera's coordinate system. Denote $n_L = \big[n_L(x), n_L(y), n_L(z)\big]$. Then, $n_L(x) \in [-1, 1], n_L(y) \in [-1, 1]$, $n_L(x)^2 + n_L(y)^2 \leq 1$

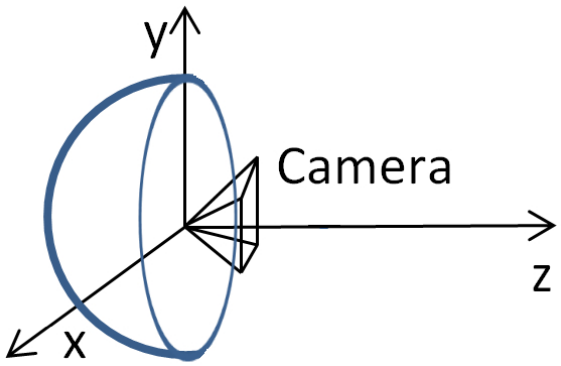


Figure 5.3: The search space for the normal of the refractive interfaces.

and $n_L(z) = -\sqrt{1 - x^2 - y^2}$. Both $n_L(x)$ and $n_L(y)$ are within $[-1, 1]$ because n_L represents the normal direction and its length is 1. The search space for n_R can be defined in the same way. Since the relative pose between the two cameras are calibrated in air, the relationship between their coordinate systems can be obtained. As a result, the search space for both n_L and n_R can be defined.

There are many hypotheses for n_L and n_R in the search spaces, and the reprojection error is used to measure whether or not the result produced by a particular hypothesis is appropriate. To be more specific, for a certain hypothesis of n_L and n_R,

d_L^i and $d_R^i, i \in [0, N-1]$ is computed using the new method in Section 5.1.1. After that, M pairs of corresponding rays generate M 3D points by ray-tracing, and these 3D points are projected back to the images to compute the reprojection error. It is demonstrated in [1] that by solving a 4^{th} degree equation the reprojected pixels can be obtained for a single refraction, and a 12^{th} degree equation for two refractions. Assuming that the measured corresponding pixels in the two images are $\hat{p}_L$ and $\hat{p}_R$, and the corresponding reprojected pixels are p_L and p_R, then the reprojection error is defined as the following root mean square (RMS) error:

$$\mathcal{J} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\Big(p_L(i) - \hat{p}_L(i)\Big)^2 + \Big(p_R(i) - \hat{p}_R(i)\Big)^2}. \tag{5.8}$$

The hypothesis of n_L and n_R that minimizes $\mathcal{J}$ is selected.

5.1.3 Implementation Details

A naive implementation of the method in Section 5.1.2 is a brute force search in the search space for the normals. For example, one can use a step size of 0.01 when searching the entire space for n_L and n_R. However, this implementation is very time consuming and the results may not be accurate enough because the step size is 0.01. Instead, binary search is used in the defined search space. In particular, the proposed implementation terminates at the 10^{th} iteration and the step size at the i^{th} iteration is 0.5^i. At each iteration, there are 25 hypotheses each for n_L and n_R, which result in a total of $(25)^2$ hypotheses. Therefore, this implementation explores $(25)^2 \times 10$ hypotheses and the final step size is 0.5^{10}, which is much finer than 0.01 using the

brute force approach.

Since the established corresponding rays such as $\{r_L^0, r_R^0\}$ can be corrupted by noise, a post-processing step is performed. The sparse bundle adjustment is used [55] to refine the parameters n_L, n_R, d_L^i and $d_R^i, i \in [0, N-1]$.

5.1.4 Experimental Results

Extensive experiments have been performed and the results are reported here. The refractive index for air is 1.0, for water 1.33 and for glass 1.50. Assume that the light path from the camera to the object is air $\rightarrow$ glass $\rightarrow$ water. This is the most common scenario for both the lab environment and for camera systems deployed undersea. For example, water is usually contained in a tank made of glass in the lab environment. Cameras are normally placed in their own housing equipment made of glass before they are deployed undersea. From now on, the term "single approximation" is used to represent that a single refraction of water is used to approximate the refraction of glass + water. Under this circumstance, the glass thickness is not estimated.

5.1.4.1 Simulations

The following six cases are tested in the simulated experiment. **Case 1**: Known refractive normal + single approximation. **Case 2**: Known refractive normal + known glass thickness. **Case 3**: Known refractive normal + estimate both layers' thickness. **Case 4**: Unknown refractive normal + single approximation. **Case 5**: Unknown refractive normal + known glass thickness. **Case 6**: Unknown refractive normal +

estimate both layers' thickness. The results for **Case 4** and **6** are compared with that produced by [1]. The reason for comparing only these two cases is that the method of [1] assumes that both the refractive normal and the glass thickness are unknown. **Case 2** and **Case 5** are designed because all the housing equipment is custom built and the thickness of the glass is normally known. The positions of the cameras, the normals of the refractive interfaces, the distances from the camera centers to the glass are randomly generated. Since the thickness of the glass varies for different systems, it is set to be a factor of the distance from the camera center to the glass. The factor varies from 0.1 to 1.9. The object is set to be a plane in order to easily inspect the quality of the reconstruction. Another reason is that the results of the new method are compared with that produced by [1], which requires a planar calibration object. It is noteworthy that the new method does not require the scene to be a planar object. In the real experiments, arbitrary 3D objects are used. The size of the object is 0.5×0.5 units. Throughout the simulated experiments, the image resolution is 2048×1536 pixels. The number of pixel correspondences established from the object is $M = 50$. The correspondences are generated with Gaussian noise (variance σ^2 pixels) and 100 trials are performed for each noise setting. The results are evaluated as follows. Suppose the refractive normal recovered by the new method is $\hat{n}_L$ and the ground truth is n_L, then the angle between them is computed by $\arccos(\hat{n}_L \cdot n_L) \times 180^\circ$ and termed "angular error." It is applied to n_R as well. Moreover, assume that the layer thickness recovered by the new method is $\hat{d}_L^i, i \in [0, N-1]$ and the ground truth is d_L^i, then the normalized error is computed by $\frac{|\hat{d}_L^i - d_L^i|}{d_L^i}$ where $|\cdot|$ denotes

absolute value. The error of d_R^i is also computed. Besides the above parameters, a 3D error is computed as follows. Combining the estimated parameters and the generated correspondences, the 3D models of the objects can be reconstructed. They are compared with the actual 3D points. Denote the coordinates of each 3D point as P, and the coordinates computed using the estimated parameters as $\hat{P}$. The distance between these two points is obtained and averaged for the entire object. D is used to denote the averaged distance. One can see that all the measures indicate errors in the results. Therefore, a lower value in a curve indicates a better result.

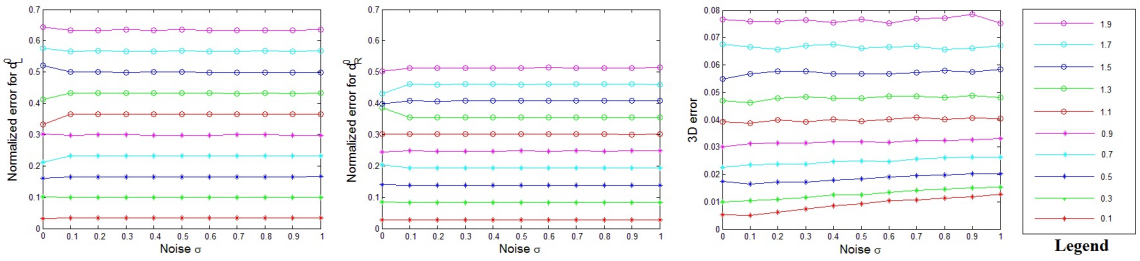


Figure 5.4: Experimental results for **Case 1.**

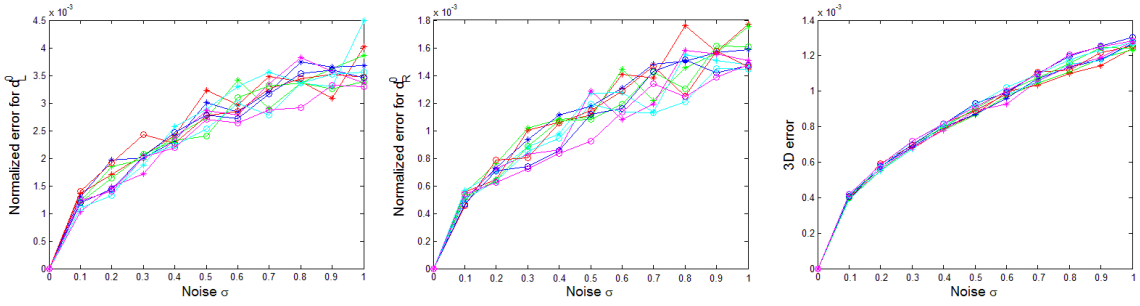


Figure 5.5: Experimental results for **Case 2.**

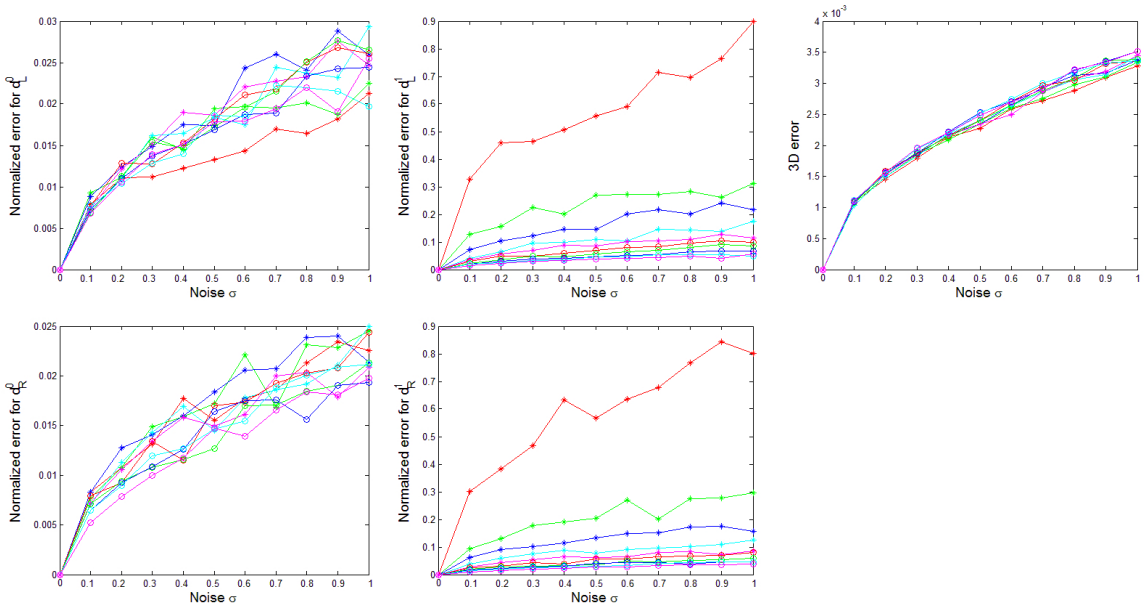


Figure 5.6: Experimental results for **Case 3.**

The results for all the cases are shown in Figure 5.4 to 5.9. Overall, the results in the category when the refractive normal is known (**Case 1-3**) are better than when it is unknown (**Case 4-6**). It implies that the refractive normal has a large impact on the results. It is possibly because when the normal is incorrect, the error in the refracted ray can be large. Another interesting result is that regardless of whether or not the refractive normal is known, the results when the glass thickness is known are the best, and then the results when the glass thickness is unknown are the next, and the results using single approximation are the worst. The results of **Case 4** and **6** by the new method are compared with that of applying [1]. The comparison shows that the accuracy of the new method is similar to that of [1], which requires a calibration target such as a checkerboard pattern.

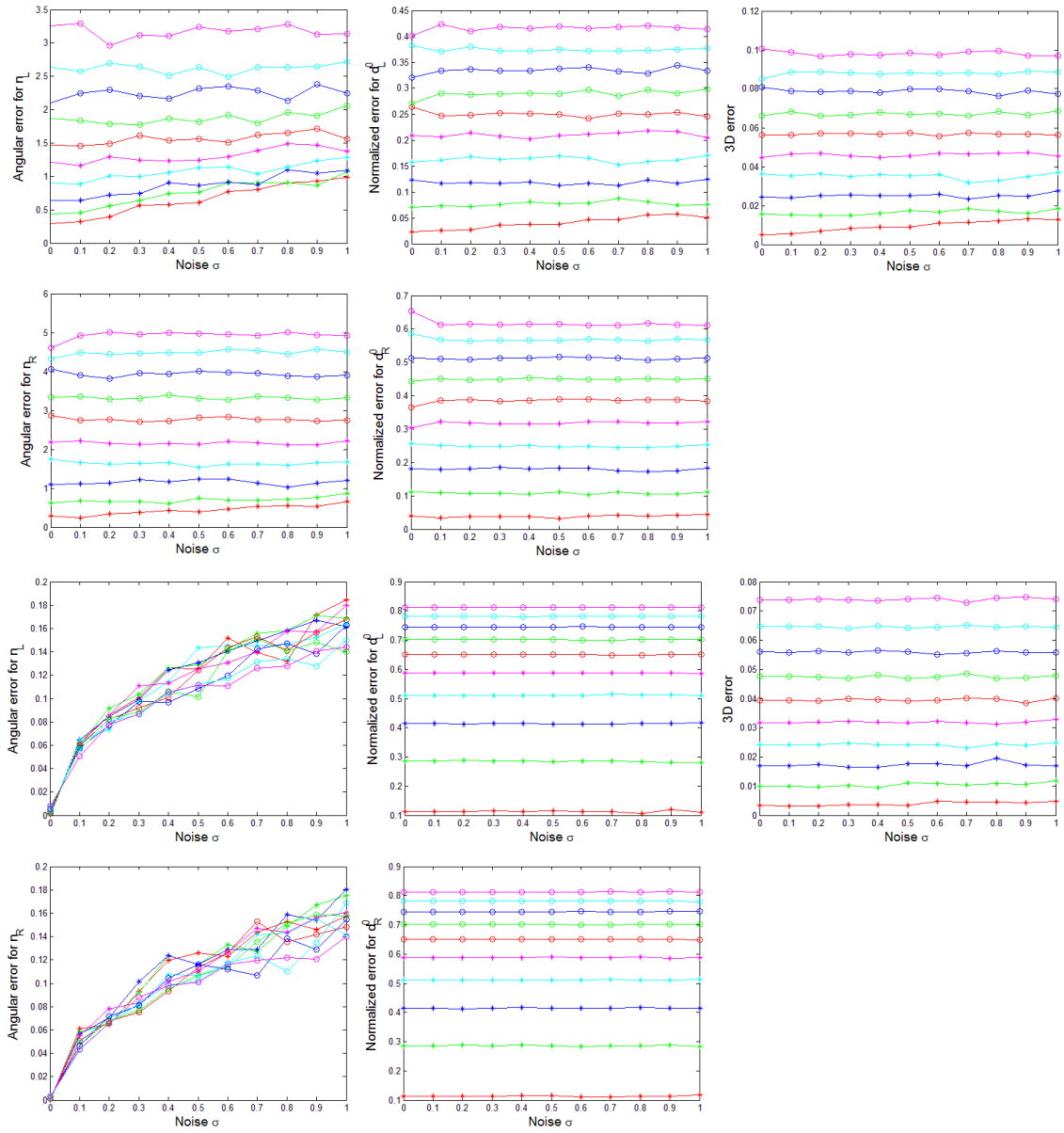


Figure 5.7: Experimental results for **Case 4.**

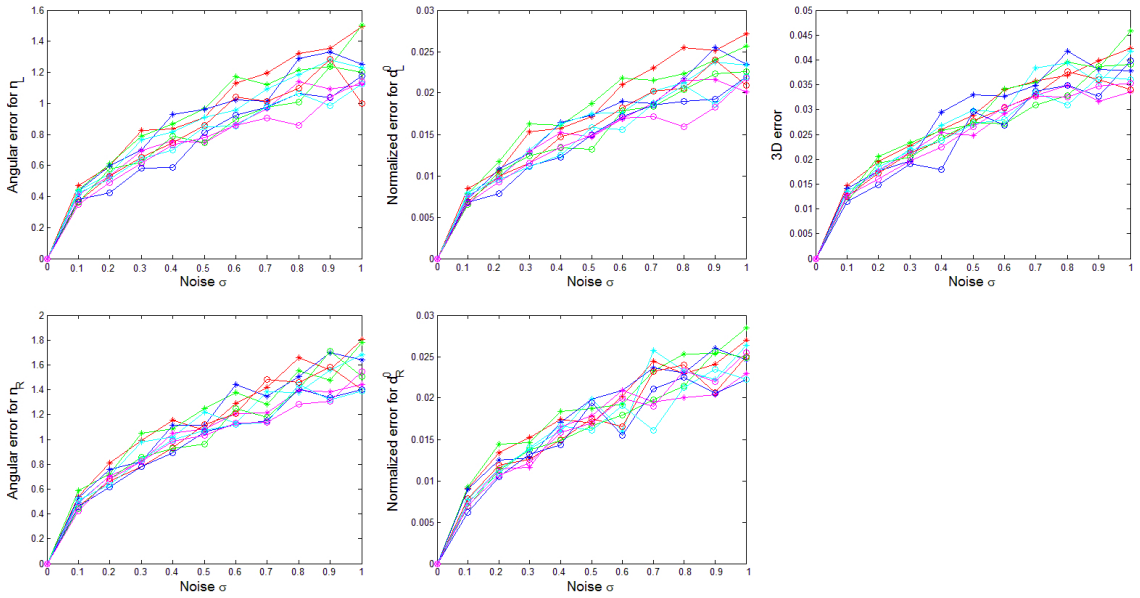


Figure 5.8: Experimental results for **Case 5.**

5.1.4.2 Simulations with Outliers

Besides noise, outliers are added to the correspondences and repeat the experiments. The experiments are designed as follows to demonstrate that the new method is robust to outliers. The datasets generated in Sec. 5.1.4.1 are re-used where each one has $M = 50$ pairs of correspondences. However, some of the correspondences are changed to be outliers and the number of outliers varies from 0% to 85%. The noise variance is set to be 0.5 pixels and the experiments are repeated for all the 6 cases described in Section 5.1.4.1.

RANSAC is used in the implementation to handle outliers. In particular, 4 corresponding pairs are used in each RANSAC trial, and the maximum number of trials is set to be 500. **Case 2** is used as an example to demonstrate the implementation.

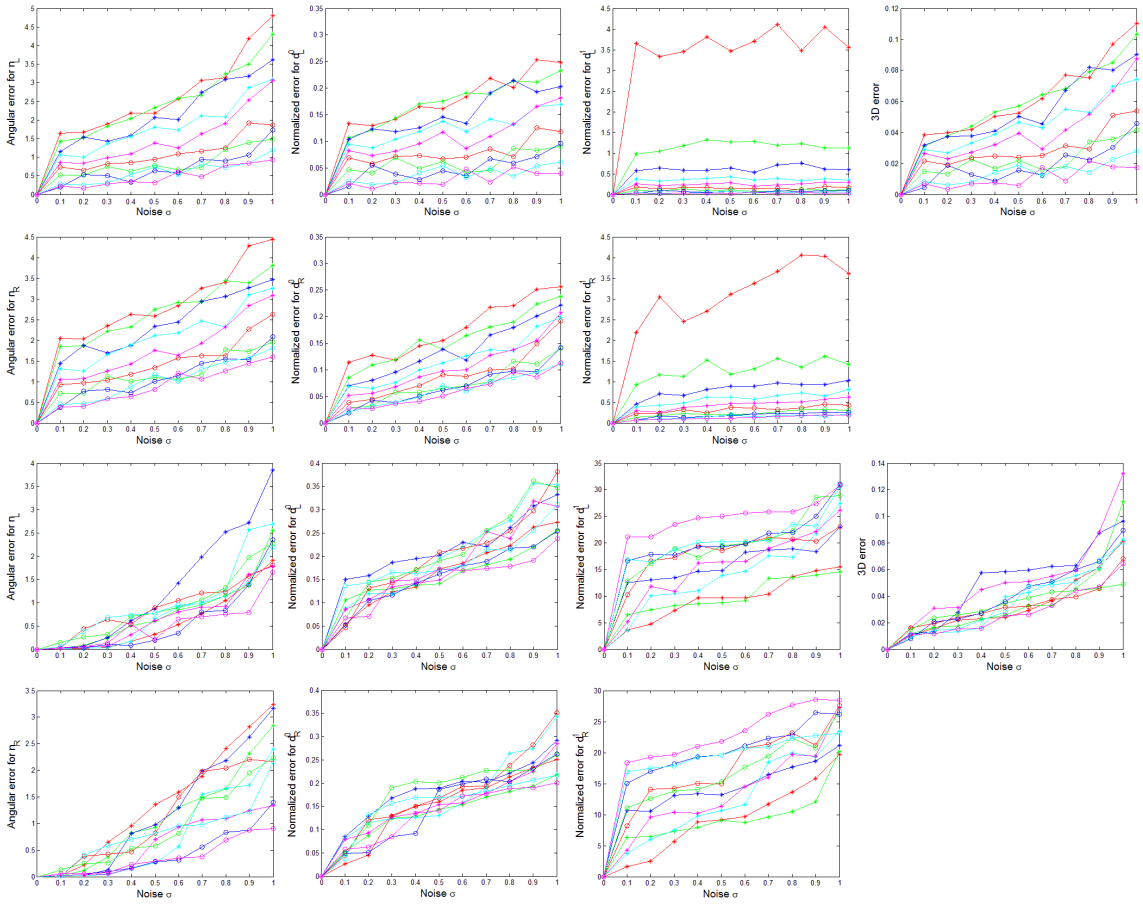


Figure 5.9: Experimental results for **Case 6.**

In this case, the glass thickness is known and the unknowns are d_L^0, d_R^0. That is, two pairs of correspondences are sufficient to solve the linear equation Eq. 5.1. However, since the correspondences are corrupted by Gaussian noise, the solution of $Ax = b$ using two pairs is not robust. In the implementation, RANSAC randomly selects four pairs of correspondences to solve the linear equation $Ax = b$, and then checks how many other correspondences can be fit into this linear equation. The solution that fits the most number of correspondences is selected. The result for **Case 2** is

shown in Figure 5.10. The curves for the other cases have a similar shape. The new method can handle up to 80% of outliers. The probability of selecting four pairs of correspondences to be all inliers at 80% and 85% is analyzed to explain the result. When the outliers is 80%, there are 10 pairs of correspondences that are inliers among a total of 50 pairs. The probability for RANSAC to select four pairs to be all inliers is $C_{10}^4/C_{50}^4 = 9.1185 \times 10^{-4}$. When there are 85% of outliers which means there are only 7 pairs of inliers, the probability becomes $C_7^4/C_{50}^4 = 1.5198 \times 10^{-4}$, which is 6 times smaller than when it is 80%. One can see that the probability at 85% is very small, which explains the sudden jump in the error curve. Nonetheless, the number of outliers in the established correspondences is usually fewer than 80% in reality.

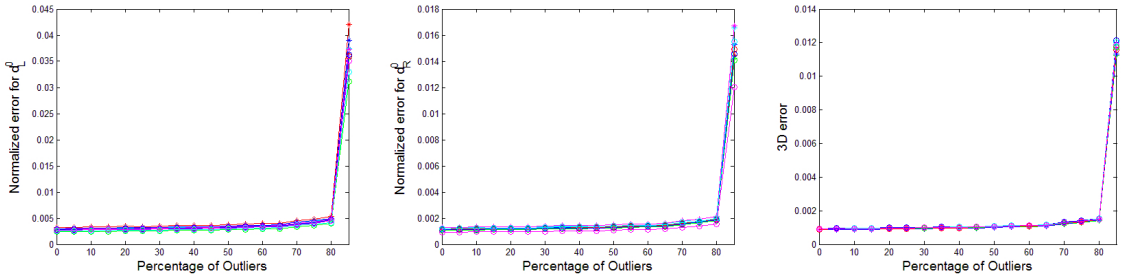
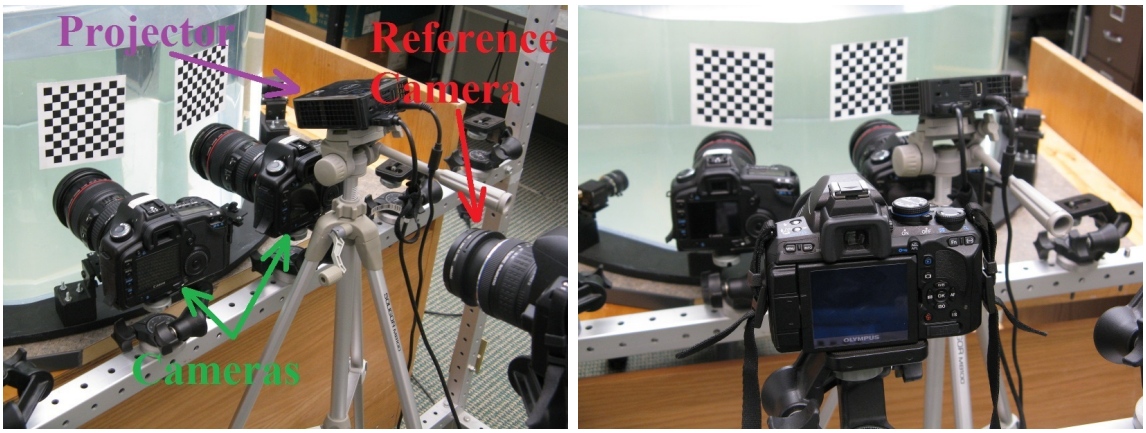


Figure 5.10: Experimental results for **Case 2** with ourliers in the input.

5.1.4.3 Real Data

Besides the simulated experiments, real experiments are carried out in a lab environment and the setup is shown in Figure 5.11. Two images are taken from different angles to show the details of the setup. In particular, two Canon 5D cameras (indi-



(a) (b)

Figure 5.11: Setup for the real experiments.

cated by the green arrows) are placed in front of a plexiglass tank and each camera has its own refractive interface. A projector, which is indicated by the purple arrow, is used to project gray code structured light patterns in order to establish dense correspondences between the cameras. In the experiments, SURF feature matches are established to test the new methods, where the gray code correspondences are used for the final dense 3D reconstruction in order to visually inspect the quality of the 3D model. The resolution of the cameras is 4368 $\times$ 2912 and of the projector is 1024 $\times$ 768. A reference camera (indicated by the red arrow) is used to obtain the ground truth for the refractive axis (n_L, n_R) and the distance from the two Canon cameras to the refractive interface (d_L^0, d_R^0). The reference camera focuses on both checkerboard patterns. Therefore, the transformation between the coordinate systems of the checkerboard and the reference camera can be computed through calibration. After that, once the three cameras are calibrated in air, the transformation between the

coordinate systems of the checkerboard and the two Canon cameras can be computed. As a result, the ground truth for the refractive axis and the distance can be obtained. The thickness of the glass (d_L^1, d_R^1) is measured by a ruler. The results are compared with the ground truth to measure the accuracy.

Three datasets are captured under the same stereo camera configuration. In particular, the three datasets are named "Plane", "Cave" and "Boat". SURF correspondences are established for each dataset and the parameters are estimated from each set of correspondences. Once all the parameters (n, d) are estimated, the 3D reconstruction can be performed from gray code correspondences. The results of the "Plane" scene are shown in Figure 5.12. From left to right, Figure 5.12 shows the

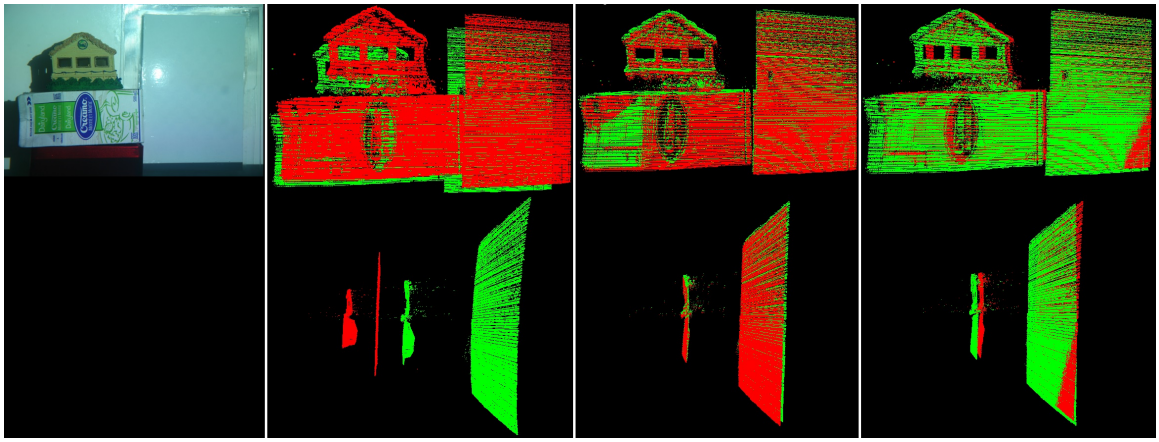


Figure 5.12: Results of the "Plane" scene. From left to right: captured image, 3D reconstruction results for the case when refraction is not accommodated, results of **Case 5** and **6** using the presented method. The two rows shows the 3D reconstruction results in two different views. The ground truth is shown in red for comparison.

captured image, the 3D reconstruction results when refraction is not accounted for, the results of **Case 5** and **6** by applying the new method. The "Plane" scene includes a piece of laminated paper pasted on a glass to show the impact of the refraction effects. Since the ground truth is measured for all the parameters (n, d), the "ground truth" of the 3D reconstruction can be obtained by back projecting the established pixel correspondences. The ground truth is shown in red for comparison. To demonstrate that the measurement of ground truth is accurate, two points are marked on the milk carton. The distance between these two points from 3D reconstruction using the measured ground truth is 99.0mm, and the distance measured using a ruler is 98.7mm. The above two numbers indicate that the measurement of the ground truth for the parameters is very accurate. From the second column of Figure 5.12, one can see that the objects are larger in 3D reconstruction when the refraction effects are ignored which is the case in reality. In particular, the distance between the two points marked on the milk carton is 149.8mm when refraction is not accounted for, which means that the scene is assumed to be placed in air. One can see that the error is quite large compared to the one measured by a ruler. Moreover, the reconstruction of the laminated paper is not flat anymore. It demonstrates that the error can be significant when the refraction effects are not accommodated in 3D reconstruction. The results for the datasets "Cave" and "Boat" are shown in Figure 5.13. The top row is the result for the "Cave" scene and the bottom row for the "Boat" scene. From left to right, the figure shows the captured image, result without accounting for refraction, results of **Case 5** and **Case 6**. The ground truth is shown in red. By visual

comparison with the ground truth, the results are the best in **Case 5** and worst in **Case 6**, which is consistent with the conclusion in the simulated experiments.

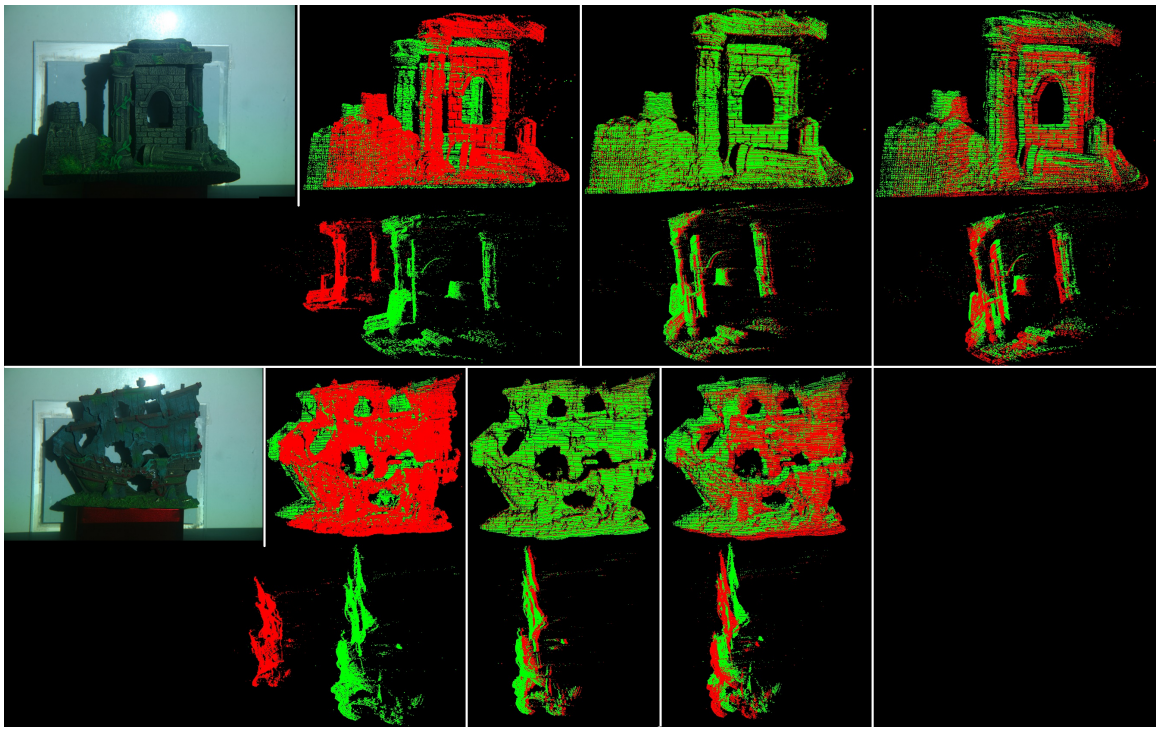


Figure 5.13: Top row: results of the "Cave" dataset. Bottom row: results of the "Boat" dataset. The ground truth is shown in red for comparison.

Besides the visual comparison of the 3D reconstruction results between the ground truth and the new method, the results of the housing parameters for the "Cave" scene are shown in Table 5.1. The unit for the angular error is degree and for thickness is mm. Some cells are filled in with "N/A" because it is not estimated in that case. For example, n_L and n_R are assumed to be known in **Case 1, 2** and **3**. "D" represents the 3D reconstruction error with unit in mm. The results indicate that most of the

parameters are well estimated except the glass thickness where large errors still exist. Therefore, it is strongly recommended that researchers should measure the thickness of the refractive interface when building the underwater camera systems. Moreover, a checkerboard pattern is placed in the water and the images are taken by both cameras in order to apply the method proposed in [1]. Comparing with [1], the new method is comparable in **Case 4**, while it is worse in **Case 6**. Nonetheless, [1] requires the 3D geometry of the scene to be known while the new method does not.

5.1.5 Limitation

When solving Eq. 5.7 which has the same form as $Ax = b$, an important assumption is that A has to be full rank. Moreover, any two rays r_L^i and r_L^j cannot be parallel. Similar assumption is applied to r_R^i. The second assumption infers that any two layers cannot be the same material. The following two solutions are proposed to address such a limitation. (1) If there are layers of the same material, the sum of the thicknesses for the same material layers can be computed. However, the thickness for each layer cannot be obtained separately. (2) If the layer material is not known, the rank of A can be computed. After that, the sum of the thicknesses for the layers of the same materials can be computed.

	angular error for n_L	angular error for n_R	d_L^0	d_R^0	d_L^1	d_R^1	D
GT	0	0	123.35	186.00	5.6	5.6	0
Case 1	N/A	N/A	128.93	196.11	N/A	N/A	1.85
Case 2	N/A	N/A	122.43	187.31	N/A	N/A	1.06
Case 3	N/A	N/A	126.35	191.2	55.36	68.19	1.77
Case 4	1.9866	1.3979	128.86	195.74	N/A	N/A	4.63
Case 4 ([1])	0.5321	0.5145	134.65	202.36	N/A	N/A	4.37
Case 5	1.1588	1.3268	125.54	189.75	N/A	N/A	1.67
Case 6	4.2682	5.842	180.47	273.59	69.23	81.97	16.22
Case 6 ([1])	0.8135	0.7754	151.23	233.62	95.52	89.97	8.20

Table 5.1: Comparison between the estimated parameters by the new method and the ground truth. "GT" denotes the ground truth.

5.2 Single Camera Housing Calibration

In the previous section, a method that estimate the housing parameters for stereo cameras is presented. The method requires searching the normal search space to estimate the refractive normal, which can be inefficient. In this section, a new method that utilizes triple wavelength dispersion to estimate housing parameters more efficiently using a single camera is presented.

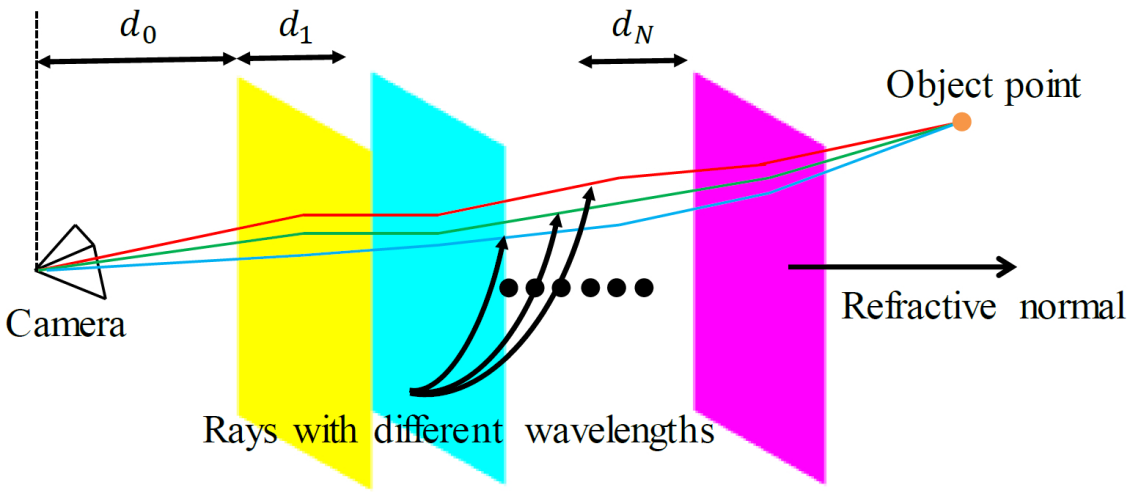


Figure 5.14: Refractive model.

Figure 5.14 shows the refractive model that is used in this method. A pinhole perspective camera is viewing the object through multiple flat refractive layers where all layers are parallel to each other, which means that all layers have the same refractive interface normal n. The number of layers is N and each layer has a thickness of d_i. The distance from the camera center to the first refractive interface is denoted as

the "interface distance", d_0. Assume that the object is able to emit three different wavelengths of light (red, green and blue), then the dispersion can be observed due to different refractive indices for different wavelengths. The goal is to compute both the refractive normal n and the interface distance d_0.

Let's focus on a specific refraction case which is the most common and practical one. In particular, assume that there are two refractions which are air $\rightarrow$ glass $\rightarrow$ water. This is the most common scenario because a camera is always placed inside a watertight housing before deployed underwater. Therefore, the camera is viewing the object in water through a piece of glass. In the lab environment, the water is always put inside a tank which is made of glass. The new method assumes that the thickness of the glass is known because it can be easily measured.

5.2.1 Normal Computation

The method of normal estimation is the same as the one described in [90]. In particular, when observed by the camera, an object point reaches different pixel locations in the camera through different rays due to different wavelengths. Denote v_a, v_b as the two rays with different wavelengths that travel in air and reach the camera, then these two rays must lie on the same plane as the refractive normal n. As a result, a constraint for n can be written as follows.

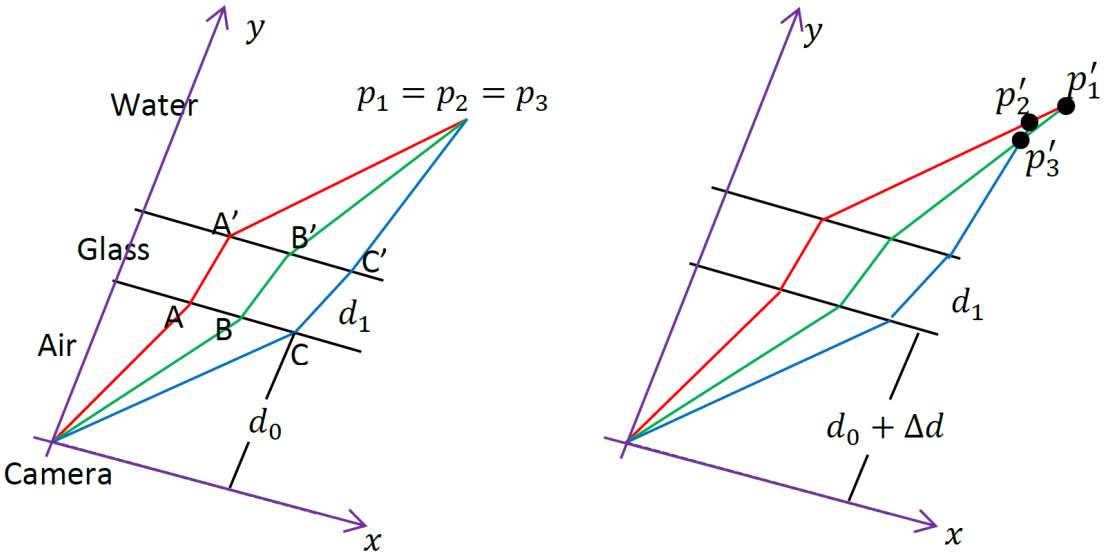
$$(v_a \times v_b)^T n = 0. \tag{5.9}$$

Due to different wavelengths, $v_a \neq v_b$ and the above constraint describes that n is on the same plane formed by v_a and v_b. Assume that there are multiple object points

described above, then by stacking multiple linear equations yields a linear system. Thus, the refractive normal n can be computed by solving this linear system.

5.2.2 Interface Distance Computation

The most important novelty of the new method is in estimating the interface distance d_0. In particular, a mathematical proof and derivation is provided to show that there is a closed-form solution to d_0 when the interface normal n is known based on the assumption that triple wavelength dispersion is observed. Because of this finding, the new method does not require any calibration target.



(a) (b)

Figure 5.15: Computing the interface distance d_0. See text for details.

Figure 5.15 depicts a typical case of triple wavelength dispersion. In particular,

due to the dispersion of the red, green and blue colors, an object point is observed at three different pixel locations in the captured images. The direction of the three rays in air can be computed once the camera intrinsic parameters are known. Since the refractive interface normal n can be obtained by the method described in Section 5.2.1, the direction of the rays traveling in glass and in water can be computed as well. When d_0 is the ground truth, the three rays in water intersect at three coincident points p_1, p_2 and p_3. With reference to Figure 5.15(b), in the following, a theorem is presented to prove that when d_0 is changed to $d_0 + \Delta d$ with any $\Delta d \neq 0$, the three rays in water will intersect at three non-coincident points p_1', p_2' and p_3', and vice versa. Using this important finding, then d_0 corresponds to the point at which the three points p_1, p_2 and p_3 are coincident. It is noteworthy that using triple wavelength dispersion is critically important in the new method and cannot be replaced using two wavelengths such as the method in [90]. The reason is as follows. Assume that there are only two wavelengths (red and blue), and hence there are two rays in water. One can see that with any d_0, these two rays in water will always intersect at one point. Because no calibration target is used, it cannot determine which d_0 corresponds to the ground truth point in water. In contrast, the new method can determine that the correct d_0, which corresponds to the point where p_1, p_2 and p_3 are coincident. In general, the new method can be extended to more than 3 wavelengths.

Let's denote the directions of the three rays in air as v_a^r, v_a^g, v_a^b, where the subscript "a" represents air and the superscript color, namely, red(r), green(g) and blue(b). Using this notation, the directions of rays are denoted as v_g^r, v_g^g, v_g^b in glass, and

v_w^r, v_w^g, v_w^b in water.

Lemma 2. *$v_a^r, v_a^g, v_a^b, v_g^r, v_g^g, v_g^b, v_w^r, v_w^g, v_w^b$ are always on the same plane even if $\Delta d \neq 0$ as shown in Figure 5.15(b).*

Proof. According to Snell's Law, all the incoming rays and the refracted rays are always on the same plane as the refractive normal. Moreover, Section 5.2.1 demonstrates that v_a^r, v_a^g, v_a^b are always on the same plane because they are due to the dispersion effects. In other words, all the 9 rays in Figure 5.15(a) are on the same plane. Notice that once the interface normal n is known, v_g^r, v_g^g, v_g^b and v_w^r, v_w^g, v_w^b are also fixed because they depend on v_a^r, v_a^g, v_a^b and n only. Therefore, even if d_0 is changed to $d_0 + \Delta d$, the ray direction does not change. As a result, the 9 rays are always on the same plane. □

Now that all the rays are on the same plane, the problem can be simplified from 3D to 2D. In particular, assume that the z plane is aligned with the plane that is formed by the above 9 rays. As shown in Figure 5.15(a), the camera center is selected as the origin of the new coordinate system. The y axis is perpendicular to the refractive interface, and the x axis is set so that all the 9 rays are at the first quadrant. Denote the slope of the 9 rays as $m_a^r, m_a^g, m_a^b, m_g^r, m_g^g, m_g^b, m_w^r, m_w^g, m_w^b$.

Theorem 2. *In Fig. 5.15(b), $p_1' \neq p_2', p_1' \neq p_3'$ and $p_2' \neq p_3' \Leftrightarrow \Delta d \neq 0$.*

Proof. The direction of $\Leftarrow$ is proved first. The three rays in air pass the origin,

therefore their equations are written as follows.

$$\begin{aligned} y &= m_a^r x \\ y &= m_a^g x \\ y &= m_a^b x \end{aligned} \tag{5.10}$$

These three rays intersect the inner glass interface at three different points whose coordinates are $\left(\frac{d_0}{m_a^r}, d_0\right)$, $\left(\frac{d_0}{m_a^g}, d_0\right)$ and $\left(\frac{d_0}{m_a^b}, d_0\right)$. Similar to that, The equation of those three rays inside the glass can be written as follows.

$$\begin{aligned} y &= m_g^r x + \left(d_0 - d_0 \frac{m_g^r}{m_a^r}\right) \\ y &= m_g^g x + \left(d_0 - d_0 \frac{m_g^g}{m_a^g}\right) \\ y &= m_g^b x + \left(d_0 - d_0 \frac{m_g^b}{m_a^b}\right) \end{aligned} \tag{5.11}$$

These three rays intersect the outer glass at three points where the coordinate are $\left(\frac{d_1}{m_g^r} + \frac{d_0}{m_a^r}, d_0 + d_1\right)$, $\left(\frac{d_1}{m_g^g} + \frac{d_0}{m_a^g}, d_0 + d_1\right)$ and $\left(\frac{d_1}{m_g^b} + \frac{d_0}{m_a^b}, d_0 + d_1\right)$. Given these three points, the equation for the three rays in water can be written as follows.

$$\begin{aligned} y &= m_w^r x + \left(d_0 + d_1 - d_1 \frac{m_w^r}{m_g^r} - d_0 \frac{m_w^r}{m_a^r}\right) \\ y &= m_w^g x + \left(d_0 + d_1 - d_1 \frac{m_w^g}{m_g^g} - d_0 \frac{m_w^g}{m_a^g}\right) \\ y &= m_w^b x + \left(d_0 + d_1 - d_1 \frac{m_w^b}{m_g^b} - d_0 \frac{m_w^b}{m_a^b}\right) \end{aligned} \tag{5.12}$$

Assume that p_1 is the intersection between the red and green ray, p_2 the red and blue ray and p_3 the green and blue ray. The coordinate for p_1, p_2 and p_3 can now be computed. In particular, let's focus on the x component of these three points only

and they are written as follows.

$$
\begin{aligned}
p_1(x) &= \frac{\frac{d_1 m_w^r}{m_g^r} - \frac{d_1 m_w^g}{m_g^g} + \frac{d_0 m_w^r}{m_a^r} - \frac{d_0 m_w^g}{m_a^g}}{m_w^r - m_w^g} \\
p_2(x) &= \frac{\frac{d_1 m_w^r}{m_g^r} - \frac{d_1 m_w^b}{m_g^b} + \frac{d_0 m_w^r}{m_a^r} - \frac{d_0 m_w^b}{m_a^b}}{m_w^r - m_w^b} \\
p_3(x) &= \frac{\frac{d_1 m_w^g}{m_g^g} - \frac{d_1 m_w^b}{m_g^b} + \frac{d_0 m_w^g}{m_a^g} - \frac{d_0 m_w^b}{m_a^b}}{m_w^g - m_w^b}
\end{aligned}
\tag{5.13}
$$

From $p_1 = p_2 = p_3$ one can infer that $p_1(x) = p_2(x) = p_3(x)$.

The slope of all rays in Figure 5.15(b) is the same as that of in Figure 5.15(a). Following the above calculation, the three rays in air have the same equation as shown in Eq. 5.10. They intersect the inner glass interface at three points with coordinate $\left(\frac{d_0+\Delta d}{m_a^r}, d_0 + \Delta d\right)$, $\left(\frac{d_0+\Delta d}{m_a^g}, d_0 + \Delta d\right)$ and $\left(\frac{d_0+\Delta d}{m_a^b}, d_0 + \Delta d\right)$. With that, the equation of the three rays inside the glass can be written as follows.

$$
\begin{aligned}
y &= m_g^r x + \left(d_0 + \Delta d - (d_0 + \Delta d)\frac{m_g^r}{m_a^r}\right) \\
y &= m_g^g x + \left(d_0 + \Delta d - (d_0 + \Delta d)\frac{m_g^g}{m_a^g}\right) \\
y &= m_g^b x + \left(d_0 + \Delta d - (d_0 + \Delta d)\frac{m_g^b}{m_a^b}\right)
\end{aligned}
\tag{5.14}
$$

These three rays intersect the outer glass at three points where the coordinate are $\left(\frac{d_1}{m_g^r} + \frac{d_0+\Delta d}{m_a^r}, d_0 + \Delta d + d_1\right)$, $\left(\frac{d_1}{m_g^g} + \frac{d_0+\Delta d}{m_a^g}, d_0 + \Delta d + d_1\right)$ and $\left(\frac{d_1}{m_g^b} + \frac{d_0+\Delta d}{m_a^b}, d_0 + \Delta d + d_1\right)$. Given these three points, the equation for the three rays in water can be written as follows.

$$
\begin{aligned}
y &= m_w^r x + \left(d_0 + \Delta d + d_1 - d_1\frac{m_w^r}{m_g^r} - (d_0 + \Delta d)\frac{m_w^r}{m_a^r}\right) \\
y &= m_w^g x + \left(d_0 + \Delta d + d_1 - d_1\frac{m_w^g}{m_g^g} - (d_0 + \Delta d)\frac{m_w^g}{m_a^g}\right) \\
y &= m_w^b x + \left(d_0 + \Delta d + d_1 - d_1\frac{m_w^b}{m_g^b} - (d_0 + \Delta d)\frac{m_w^b}{m_a^b}\right)
\end{aligned}
\tag{5.15}
$$

With the above equation, $p_1'(x), p_2'(x), p_3'(x)$ can be written as follows.

$$\begin{aligned} p_1'(x) &= p_1(x) + \frac{\frac{\Delta dm_w^r}{m_a^r} - \frac{\Delta dm_w^g}{m_a^g}}{m_w^r - m_w^g} \\ p_2'(x) &= p_2(x) + \frac{\frac{\Delta dm_w^r}{m_a^r} - \frac{\Delta dm_w^b}{m_a^b}}{m_w^r - m_w^b} \\ p_3'(x) &= p_3(x) + \frac{\frac{\Delta dm_w^g}{m_a^g} - \frac{\Delta dm_w^b}{m_a^b}}{m_w^g - m_w^b} \end{aligned} \tag{5.16}$$

Because $p_1(x) = p_2(x)$, one can see that in order to prove $p_1'(x) \neq p_2'(x)$, it only needs to prove that

$$\left(\frac{m_w^r}{m_a^r} - \frac{m_w^g}{m_a^g}\right)(m_w^r - m_w^b) \neq \left(\frac{m_w^r}{m_a^r} - \frac{m_w^b}{m_a^b}\right)(m_w^r - m_w^g). \tag{5.17}$$

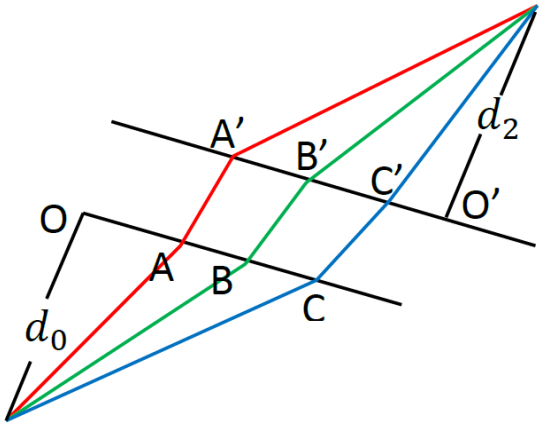


Figure 5.16: Diagram for derivation of Eq. 5.18. See text for more details.

.

Denote the length of line segment OA in Figure 5.16 as l_a^r. Similar to that, the length of OB, OC is denoted as l_a^g, l_a^b. Since m_a^r denotes the slope of a ray, it can be written as $m_a^r = \frac{d_0}{l_a^r}, m_a^g = \frac{d_0}{l_a^g}$ and $m_a^b = \frac{d_0}{l_a^b}$. Denote the length of $O'A', O'B', O'C'$ as

l_w^r, l_w^g, l_w^b, and then $m_w^r = \frac{d_2}{l_w^r}, m_w^g = \frac{d_2}{l_w^g}, m_w^b = \frac{d_2}{l_w^b}$. Substitute these into Eq. 5.17, the following equation can be derived.

$$\left(\frac{d_2 l_a^r}{d_0 l_w^r} - \frac{d_2 l_a^g}{d_0 l_w^g}\right)\left(\frac{d_2}{l_w^r} - \frac{d_2}{l_w^b}\right) \neq \left(\frac{d_2 l_a^r}{d_0 l_w^r} - \frac{d_2 l_a^b}{d_0 l_w^b}\right)\left(\frac{d_2}{l_w^r} - \frac{d_2}{l_w^g}\right)$$

$$\Rightarrow \left(\frac{l_a^r}{l_w^r} - \frac{l_a^g}{l_w^g}\right)\left(\frac{1}{l_w^r} - \frac{1}{l_w^b}\right) \neq \left(\frac{l_a^r}{l_w^r} - \frac{l_a^b}{l_w^b}\right)\left(\frac{1}{l_w^r} - \frac{1}{l_w^g}\right)$$

$$\Rightarrow \left(\frac{l_a^r l_w^g - l_a^g l_w^r}{l_w^r l_w^g}\right)\left(\frac{l_w^b - l_w^r}{l_w^r l_w^b}\right) \neq \left(\frac{l_a^r l_w^b - l_a^b l_w^r}{l_w^r l_w^b}\right)\left(\frac{l_w^g - l_w^r}{l_w^r l_w^g}\right)$$

$$\Rightarrow (l_a^r l_w^g - l_a^g l_w^r)(l_w^b - l_w^r) \neq (l_a^r l_w^b - l_a^b l_w^r)(l_w^g - l_w^r)$$

$$\Rightarrow l_w^r(l_a^g - l_a^b) + l_w^b(l_a^r - l_a^g) + l_w^g(l_a^b - l_a^r) \neq 0$$

$$\Rightarrow l_w^r(l_a^g - l_a^b + l_a^r - l_a^r) + l_w^b(l_a^r - l_a^g) + l_w^g(l_a^b - l_a^r) \neq 0$$

$$\Rightarrow l_w^r(l_a^g - l_a^r) + l_w^r(l_a^r - l_a^b) + l_w^g(l_a^b - l_a^r) + l_w^b(l_a^r - l_a^g) \neq 0$$

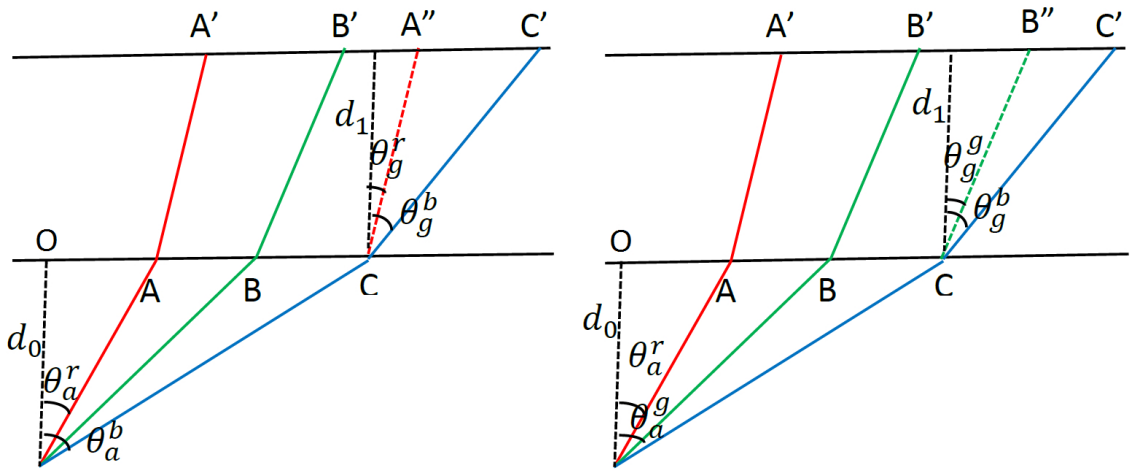
$$\Rightarrow (l_a^r - l_a^g)(l_w^b - l_w^r) \neq (l_a^r - l_a^b)(l_w^g - l_w^r)$$

Together with Figure 5.16, the above equation can be written as Eq. 5.18 where $|AC|$ denotes length of the line segment AC.

$$\frac{|A'C'|}{|AC|} \neq \frac{|B'C'|}{|BC|} \tag{5.18}$$

Denote the angle between the y axis and the red ray in air as θ_a^r, then similar notation such as $\theta_a^g, \theta_a^b, \theta_g^r, \theta_g^g, \theta_g^b$ can be used. With this notation, Eq. 5.18 can be further expanded. In Figure 5.17(a), line CA" is parallel to line AA', and the following derivation can be achieved.

$$\frac{|A'C'|}{|AC|} = \frac{|A'A"| + |A"C'|}{|AC|} = 1 + \frac{|A"C'|}{|AC|} = 1 + \frac{d_1(\tan\theta_g^b - \tan\theta_g^r)}{d_0\,(\tan\theta_a^b - \tan\theta_a^r)} \tag{5.19}$$



(a) (b)

Figure 5.17: Extension for Eq. 5.18.

Moreover, the line CB" in Figure 5.17(b) is parallel to BB', and hence the following equation can be derived.

$$\frac{|B'C'|}{|BC|} = \frac{|B'B"| + |B"C'|}{|BC|} = 1 + \frac{|B"C'|}{|BC|} = 1 + \frac{d_1(\tan\theta_g^g - \tan\theta_g^r)}{d_0\,(\tan\theta_a^g - \tan\theta_a^r)} \tag{5.20}$$

By combining Eq. 5.19, Eq. 5.20 and together with Eq. 5.18, the following equation is derived.

$$\begin{gathered}\frac{|A'C'|}{|AC|} \neq \frac{|B'C'|}{|BC|} \Rightarrow \frac{\tan\theta_g^r - \tan\theta_g^b}{\tan\theta_a^r - \tan\theta_a^b} \neq \frac{\tan\theta_g^r - \tan\theta_g^g}{\tan\theta_a^r - \tan\theta_a^g} \\ \Rightarrow \frac{\frac{\sin\theta_g^r}{\sqrt{1-(\sin\theta_g^r)^2}} - \frac{\sin\theta_g^b}{\sqrt{1-(\sin\theta_g^b)^2}}}{\frac{\mu_g^r\sin\theta_g^r}{\sqrt{1-(\mu_g^r\sin\theta_g^r)^2}} - \frac{\mu_g^b\sin\theta_g^b}{\sqrt{1-(\mu_g^b\sin\theta_g^b)^2}}} \neq \frac{\frac{\sin\theta_g^r}{\sqrt{1-(\sin\theta_g^r)^2}} - \frac{\sin\theta_g^g}{\sqrt{1-(\sin\theta_g^g)^2}}}{\frac{\mu_g^r\sin\theta_g^r}{\sqrt{1-(\mu_g^r\sin\theta_g^r)^2}} - \frac{\mu_g^g\sin\theta_g^g}{\sqrt{1-(\mu_g^g\sin\theta_g^g)^2}}}\end{gathered} \tag{5.21}$$

One can see that the left-hand-side of Eq. 5.21 is a function of μ_g^r and μ_g^b. Let's denote it as $f(\mu_g^r, \mu_g^b)$. The right-hand-side can be denoted as $g(\mu_g^r, \mu_g^g)$. Eq. 5.21 can be proved by contradiction. Assume that $f(\mu_g^r, \mu_g^b) = g(\mu_g^r, \mu_g^g)$, one can see that

this is possible only when both sides are equal to the same constant. If $f(\mu_g^r, \mu_g^b)$ is a constant, it implies that μ_g^r and μ_g^b are related, i.e. they are not independent which is a contradiction.

In order to prove $p_1'(x) \neq p_3'(x)$, it only needs to prove that $\frac{\frac{m_w^r}{m_a^r} - \frac{m_w^g}{m_a^g}}{m_w^r - m_w^g} \neq \frac{\frac{m_w^g}{m_a^g} - \frac{m_w^b}{m_a^b}}{m_w^g - m_w^b}$. Similar to the above, it can be simplified to be Eq. 5.22. The proof of this inequality is very similar to that of Eq. 5.18.

$$\frac{|A'C'|}{|AC|} \neq \frac{|A'B'|}{|AB|} \tag{5.22}$$

Finally, to prove that $p_2'(x) \neq p_3'(x)$, it needs to prove that $\frac{\frac{m_w^r}{m_a^r} - \frac{m_w^b}{m_a^b}}{m_w^r - m_w^b} \neq \frac{\frac{m_w^g}{m_a^g} - \frac{m_w^b}{m_a^b}}{m_w^g - m_w^b}$. It can be simplified to be the same as Eq. 5.18.

To conclude, for any $\Delta d \neq 0 \Rightarrow p_1' \neq p_2', p_2' \neq p_3'$ and $p_1' \neq p_3'$.

Now let's prove that $p_1' \neq p_2' \Rightarrow \Delta d \neq 0$, $p_2' \neq p_3' \Rightarrow \Delta d \neq 0$, and $p_1' \neq p_3' \Rightarrow \Delta d \neq 0$. The proof for $p_1' \neq p_2' \Rightarrow \Delta d \neq 0$ is provided, and the other two cases are very similar.

Let's rewrite Eq. 5.16 as follows:

$$\begin{aligned} \Delta d &= \frac{\big(p_1'(x) - p_1(x)\big)(m_w^r - m_w^g)}{\frac{m_w^r}{m_a^r} - \frac{m_w^g}{m_a^g}} \\ \Delta d &= \frac{\big(p_2'(x) - p_2(x)\big)(m_w^r - m_w^b)}{\frac{m_w^r}{m_a^r} - \frac{m_w^b}{m_a^b}} \end{aligned} \tag{5.23}$$

It is known that $p_1(x) = p_2(x)$, and now that $p_1'(x) \neq p_2'(x)$, one can infer that at least one of $p_1'(x) - p_1(x) \neq 0$ or $p_2'(x) - p_2(x) \neq 0$ is true. Hence, it can be concluded that $\Delta d \neq 0$, because $m_w^r - m_w^g \neq 0, m_w^r - m_w^b \neq 0$. □

Denote the distance between p_1' and p_2' as d_{12}'. In the following, Theorem 3 proves that it varies linearly with Δd.

Theorem 3. *The distance d'_{ij} where $i \neq j, i = \{1,2\}, j = \{2,3\}$, varies linearly with Δd.*

Proof. It is sufficient to focus on the x component of $d'_{12}, d'_{13}, d'_{23}$ because the y component is a linear function to x. From Eq. 5.14, the following equations can be derived.

$$\begin{aligned}
p'_1(x) - p'_2(x) &= p_1(x) - p_2(x) + \Delta d \left(\frac{\frac{m^r_w}{m^r_a} - \frac{m^g_w}{m^g_a}}{m^r_w - m^g_w} - \frac{\frac{m^r_w}{m^r_a} - \frac{m^b_w}{m^b_a}}{m^r_w - m^b_w} \right) \\
p'_1(x) - p'_3(x) &= p_1(x) - p_3(x) + \Delta d \left(\frac{\frac{m^r_w}{m^r_a} - \frac{m^g_w}{m^g_a}}{m^r_w - m^g_w} - \frac{\frac{m^g_w}{m^g_a} - \frac{m^b_w}{m^b_a}}{m^g_w - m^b_w} \right) \\
p'_2(x) - p'_3(x) &= p_2(x) - p_3(x) + \Delta d \left(\frac{\frac{m^r_w}{m^r_a} - \frac{m^b_w}{m^b_a}}{m^r_w - m^b_w} - \frac{\frac{m^g_w}{m^g_a} - \frac{m^b_w}{m^b_a}}{m^g_w - m^b_w} \right)
\end{aligned} \tag{5.24}$$

Because $p_1(x) = p_2(x) = p_3(x)$ and from Theorem 2 it is known that $\frac{\frac{m^r_w}{m^r_a} - \frac{m^g_w}{m^g_a}}{m^r_w - m^g_w} \neq \frac{\frac{m^r_w}{m^r_a} - \frac{m^b_w}{m^b_a}}{m^r_w - m^b_w}, \frac{\frac{m^r_w}{m^r_a} - \frac{m^g_w}{m^g_a}}{m^r_w - m^g_w} \neq \frac{\frac{m^g_w}{m^g_a} - \frac{m^b_w}{m^b_a}}{m^g_w - m^b_w}, \frac{\frac{m^r_w}{m^r_a} - \frac{m^b_w}{m^b_a}}{m^r_w - m^b_w} \neq \frac{\frac{m^g_w}{m^g_a} - \frac{m^b_w}{m^b_a}}{m^g_w - m^b_w}$, it can be concluded that $p'_1(x) - p'_2(x) \neq 0, p'_1(x) - p'_3(x) \neq 0, p'_2(x) - p'_3(x) \neq$ and varies linearly with Δd. □

Theorem 4. *There is a closed-form solution to d_0.*

Proof. From Eq. 5.13 together with $p_1(x) = p_2(x)$, one can derive Eq. 5.25, which concludes that d_0 has a closed-form solution.

$$d_0 = \frac{\left(\frac{d_1 m^r_w}{m^r_g} - \frac{d_1 m^g_w}{m^g_g} \right) (m^r_w - m^b_w) - \left(\frac{d_1 m^r_w}{m^r_g} - \frac{d_1 m^b_w}{m^b_g} \right) (m^r_w - m^g_w)}{\left(\frac{m^r_w}{m^r_a} - \frac{m^b_w}{m^b_a} \right) (m^r_w - m^g_w) - \left(\frac{m^r_w}{m^r_a} - \frac{m^g_w}{m^g_a} \right) (m^r_w - m^b_w)} \tag{5.25}$$

□

One can see that d_0 is easy to compute once the interface normal n is estimated.

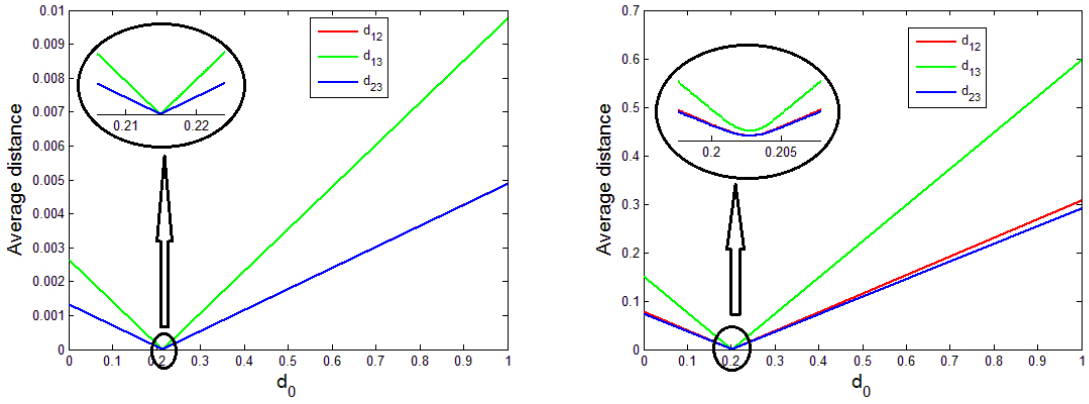
5.2.3 3D Reconstruction

Because an object point is observed at three different pixel locations due to dispersion, these three pixel locations can be back projected to compute the 3D coordinate of this object point once the interface normal n and distance d_0 is known.

5.2.4 Implementation Details

The new method depends on triple wavelength dispersion, which means that the object needs to emit three different kinds of colors (red, green and blue). The implementation to achieve this is simple and very practical. An LED projector (TI LightCrafter) is used to project three colors to illuminate the object. In this case, the object changes its original color to red, green and blue, to create the required triple wavelength dispersion.

When the observed pixel location is corrupted by noise, the three points p_1, p_2, p_3 may not be the same, in which case, d_0 is computed by minimizing the distances among these three points, which vary linearly with Δd as proved in Theorem 3, d_0 can be easily found. Figure 5.18 demonstrates the procedure. In particular, two datasets in the simulated experiment are selected where one is noise free (5.18(a)) and the other one corrupted with Gaussian noise (5.18(b)). The x axis is the value for d_0 where y is averaged d_{12}, d_{13}, d_{23} through all the object points. One can see that the distance is monotonic in d_0 even with Gaussian noise. In the case when it is noise free, all three distance are zero. Notice that the 3D object points and housing parameters are randomly generated in the simulated experiments, therefore



(a) (b)

Figure 5.18: Plot of d_{12}, d_{13}, d_{23} when the observed pixel is (a) Noise free (b). Corrupted by Gaussian noise with variance $\sigma = 0.5$ pixels.

the ground truth for d_0 is different for different datasets.

After the interface normal n and the distance d_0 are computed, a nonlinear refinement process is applied to further optimize the results. In particular, all the object points are reprojected to the image using all three wavelengths with the reprojection error as the cost function. The Matlab function $fmincon$ is applied in this step.

In the implementation of the 3D reconstruction step, the final 3D point is the barycenter of p_1, p_2 and p_3 if they do not intersect at the same point due to noise.

The Matlab code is run on a desktop PC with Intel Core i7. Solving the refractive normal takes about 1 second, the interface distance about 10 seconds, and the nonlinear optimization less than 2 minutes.

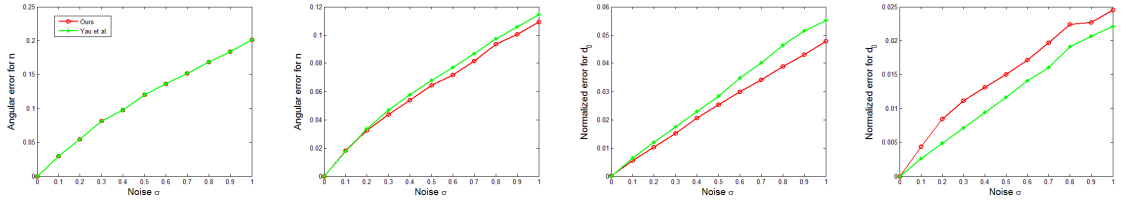
5.2.5 Experimental Results

Simulations and real experiments are designed to demonstrate the accuracy and robustness of the new method. In the experiments, it is assumed that the camera is contained in a housing equipment, and hence the light path from the camera to the object is air $\rightarrow$ glass $\rightarrow$ water, which is the most common scenario for the lab environment and for camera systems deployed underwater. The refractive index for air is 1.0 for all three wavelengths. It is 1.343 for red light in water, 1.337 for green and 1.332 for blue. It is 1.516 for red light in water, 1.502 for green and 1.488 for blue.

5.2.5.1 Simulations

In the simulated experiment, the focal length of the camera is set to be 5600 pixels. The resolution of the image is 5472×3648 with the principal point at the center of the image. This parameters is based on the Canon 6D camera used in the real experiments. It is assumed that there is no distortion in the image. In the experimental setup, the refractive normal n for each camera is randomly generated. However, the angle between the refractive normal and the camera optical axis is within a range of $[10°, 15°]$. Similarly, the interface distance d_0 is also randomly generated to be within $0.2 \sim 0.25$ units. Notice that for camera systems that require to be deployed thousands of meters undersea, the glass is usually very thick. But for other systems designed for shallow water, the glass is much thinner. Therefore, the glass thickness d_1 is a random number between 0.02 and 0.5. 100 object points are randomly generated and each is placed at a random distance that is around 1.5 units. Each object point

is observed at three different pixels and the pixel locations are corrupted by Gaussian noise (variance σ^2 pixels) and 100 trials are performed for each noise setting. The results are evaluated as follows. Suppose the refractive normal recovered by the new method is $\hat{n}$ and the ground truth is n, then the angle between them is measured in degrees and called "angular error." For interface distance d_0, the normalized error is computed by $\frac{|\hat{d_0}-d_0|}{d_0}$, where $|\cdot|$ denotes the absolute value. One can see that all the measures indicate errors, which means that a lower value in the curves indicates a better result.



(a) (b) (c) (d)

Figure 5.19: Results for the simulated experiments. (a, b) Angular error before and after nonlinear refinement. (c, d) Normalized error for d_0 before and after nonlinear refinement.

Figure 5.19 shows the results of the simulated experiments. The results are compared with that of the method [90] that is known to be the current best underwater camera calibration method. The implementation of [90] requires that the 3D geometry of the object to be known. To achieve that, it is assumed that the 3D coordinates of the object points are known in the world coordinate system. One can see that this

is impossible for an arbitrary 3D object. From the results one can see that comparing with [90], the new method can achieve similar accuracy even though it does not require the 3D geometry of the object.

5.2.5.2 Importance of Calibration

The importance of the camera housing calibration is demonstrated by showing the impact of the housing parameters to the final 3D reconstruction. The same configuration and parameters as the previously mentioned simulated experiments are used. Using the observed pixel locations with triple wavelength dispersion along with the ground truth housing parameters, these 3D object points can be computed without any error. However, once the housing parameters are changed, errors appear in the computed 3D object points. Denote the ground truth coordinates of each object point as P, and the coordinates computed by the changed parameters as $\hat{P}$. The distance between these two points is obtained and averaged for all the 3D points, which is denoted as D. One can see that D is the error when compared with the ground truth.

Figure 5.20 shows the error D. In particular, the x axis denotes the normalized error for d_0 in percentage, and the y axis denotes the angular error of the interface normal. The error D is computed when the angular error is less than 10.5° and the error for d_0 is smaller than 10%. This figure is color coded where blue indicates a smaller error and red a larger error. The color bar is also attached to the right of this figure. The figure shows that when the interface normal is 10° off from the ground

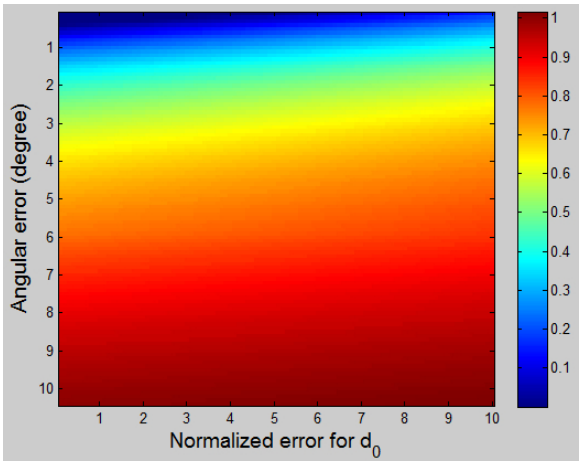
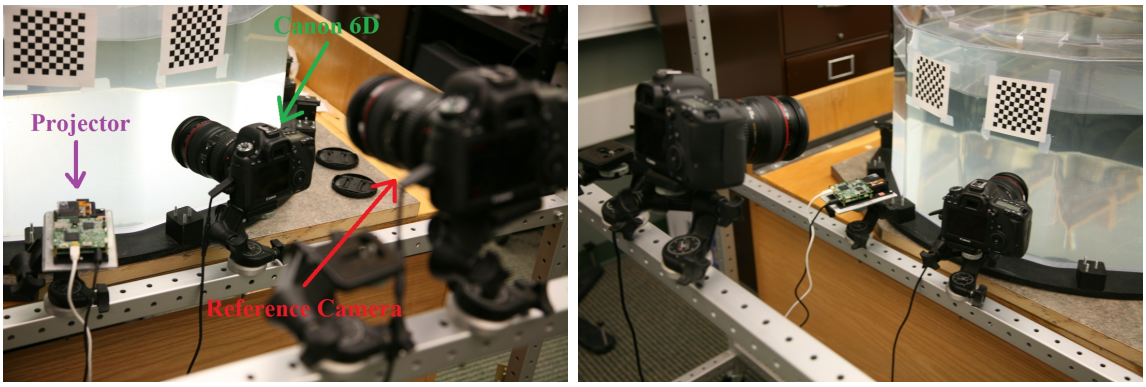


Figure 5.20: Impact of the housing parameters to the final 3D reconstruction results. See text for details.

truth and d_0 is 10% off, the error D is around 1 unit. This error is very large, because each of the 3D points in the ground truth is about 1.5 unit away from the camera. In other words, the housing parameters are critically important to the 3D reconstruction result.

5.2.5.3 Real Data

Real experiments are carried out in a lab environment and the setup is shown in Figure 5.21. A Canon 6D camera (indicated by the green arrow), with a resolution of 5472×3648, is placed in front of a plexiglass tank. A projector, which is indicated by the purple arrow, is used to project the red, green and blue lights onto the object. A reference camera (indicated by the red arrow), which is another Canon 6D camera, is used to obtain the ground truth of the refractive normal n and the interface distance d_0. In particular, the reference camera focuses on the checkerboard pattern which



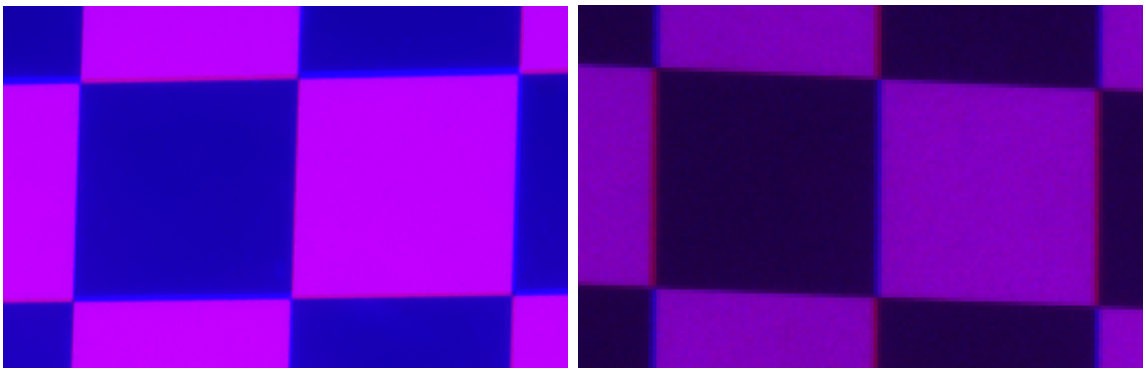
(a) (b)

Figure 5.21: Setup for the real experiments.

is pasted on the interface. Therefore, the transformation between the coordinate system of the checkerboard and that of the reference camera can be computed through calibration. Once the two cameras are calibrated in air, the transformation between the coordinate systems of the checkerboard and of the cameras facing the interface can be computed. As a result, the ground truth of n and of d_0 can be obtained, and hence the results can be evaluated against them. The thickness of the glass is measured by a ruler. In addition to using a Canon 6D camera, the same experiment is performed using a Point Grey Blackfly mono camera with a resolution of 2448 $\times$ 2048. Since the method presented in [90] requires a light box as a calibration target while the new method use an arbitrary scene, it is compare with [90].

To demonstrate that the dispersion effect can be observed by both types of cameras, a checkerboard pattern is placed in water and two images are captured when it is illuminated by the red and blue lights from the projector. If the dispersion were not observed, then the edge of the grids would align perfectly with each other in these

two images. The blue image is superimposed onto the red image and the results are shown in Fig. 5.22. One can see that the edges do not align well which indicates that the dispersion between red and blue can be observed.



(a) (b)

Figure 5.22: The dispersion effect observed by (a) Canon 6D camera and (b) Blackfly camera.

The procedures of the experiment can be described as follows. (1) The lens chromatic aberration is first corrected in air using the same method presented in [90]. (2) Red, green and blue patterns are projected onto an arbitrary scene and for each pattern, one image is captured. (3) SIFT features are detected and matched using the three images. The refractive normal and the interface distance are estimated using the detected SIFT matches. (4) Gray code [80] patterns are projected using red, green and blue in order to establish dense correspondences. There are three scenes for each type of camera. The results of using the Canon 6D camera are shown in Figure 5.23, while the results of using the Blackfly mono camera are shown in Figure 5.24.

In particular, the left image is captured under ambient light. The ones captured

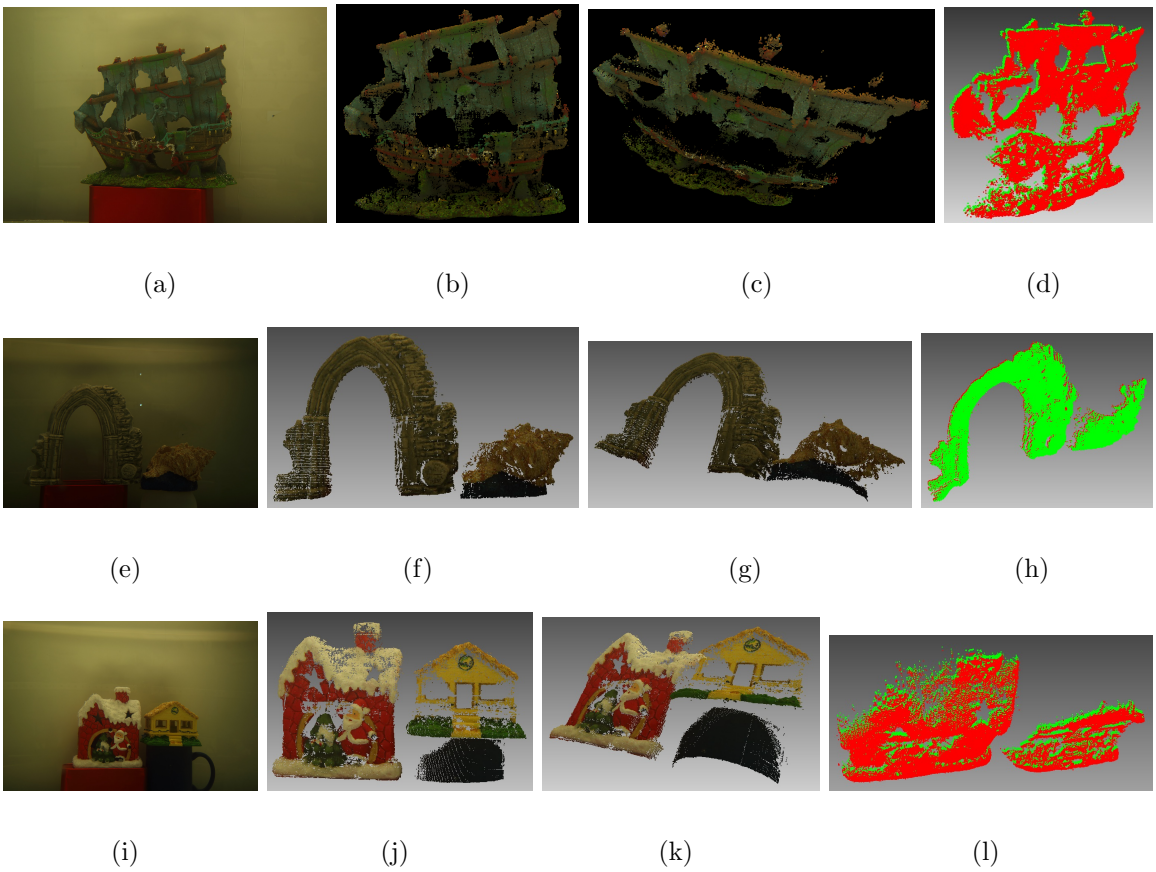
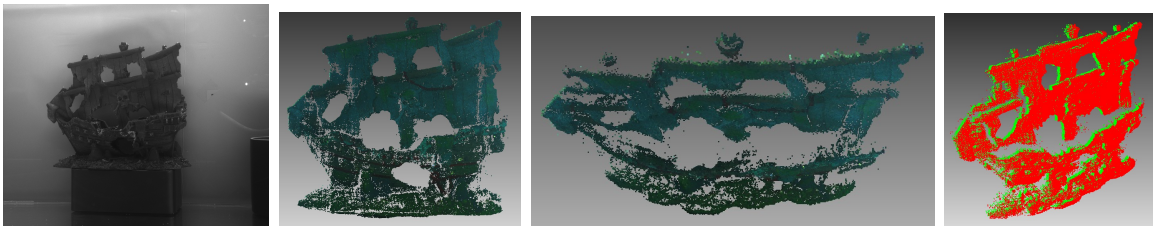
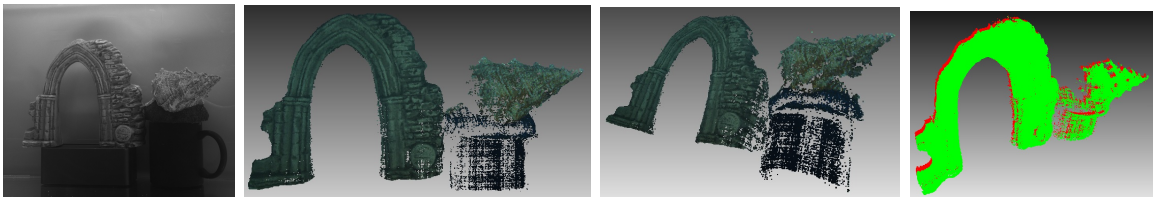


Figure 5.23: Experimental results using the Canon 6D camera. Left: Images captured under ambient light. Middle two: Reconstructed 3D point cloud from two viewpoints. Right: Comparison of 3D point cloud using the ground truth (red) and the estimated parameters(green) by the new method.

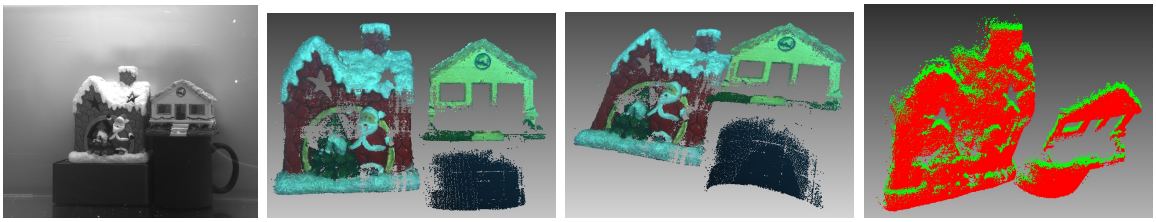
by the Blackfly mono camera have no color. The middle two are the reconstructed 3D point clouds using the dense correspondences established from gray code and the housing parameters estimated by the new method. One can see that many details are



(a) (b) (c) (d)



(e) (f) (g) (h)



(i) (j) (k) (l)

Figure 5.24: Experimental results using the Point Grey Blackfly camera. Left: Images captured under ambient light. Middle two: Reconstructed 3D point cloud from two viewpoints. Right: Comparison of 3D point cloud using the ground truth (red) and the estimated parameters(green) by the new method.

preserved in the 3D model. It is worthy to note that even though the image taken by the Blackfly camera has no color, the images captured can be combined when the scene is illuminated under the red, green and blue light, to form a color image. Even though the color in the combined image is a little different comparing to its original

color as captured by the Canon camera, it is close. The method presented in [18] is used to compare the 3D point clouds of the new method and that of the ground truth. That is, in the right image, the 3D point cloud that is reconstructed using the estimated parameters by the new method is shown in green, and by using the ground truth parameters is in red. One can see that the result of the new method is very close to the ground truth. The evaluation of the result for these two scenes is shown in Table 5.2. The error for the parameters is very close to the one that is reported in [90] by using their calibration target, which is consistent with the simulated results. It also indicates that the new method is practical. Besides the parameters, the averaged distance between the 3D points computed by the new method and the ground truth is also computed, and is denoted as "3D err" in the table. The value of the 3D error is very small, which indicates that the new method can generate good results. The real experiments also demonstrate that a black/white camera can be used because the dispersion effects can be observed. Moreover, by combining the three images captured when the scene is illuminated by red, green and blue pattern, the color information can be obtained for every pixel.

Handling outlier. The new method can handle outliers easily. In particular, a threshold is set when detecting SIFT matches using the three images. A SIFT match is regarded as outlier if the difference of pixel location in these three images is larger than a threshould. For the Canon 6D camera, the threshold is 10 pixels and 5 for the Blackly camera. The reason is that a SIFT match must be from the same physical point, even though it is observed at different pixel locations due to dispersion, the

	Err in d_0	Err in n (deg)	3D err (mm)
Canon + Boat	1.733%	0.835	1.773
Canon + Coral	1.654%	0.985	1.812
Canon + House	1.813%	0.907	1.828
Blackfly + Boat	2.189%	1.248	2.648
Blackfly + Coral	2.318%	1.093	2.393
Blackfly + House	2.276%	1.447	2.729

Table 5.2: Evaluation of the new method against the ground truth.

difference of pixel location should not be too large.

5.2.6 Discussion and Limitation

Most existing methods can be grouped into two categories. The methods in the first category require at least two cameras viewing the same scene, and estimate parameters from a set of feature matches. Two representative methods are [81, 45]. The methods in the second category perform single camera calibration with a calibration target. Two representative methods are [1, 90].

The new method is compared with some recent methods in Table 5.3 and highlight the advantages. First, comparing with the methods in the first category, estimate the relative camera pose is not required which largely reduces the number of parameters to be optimized. Moreover, the new method can use SIFT matches from the entire field of view of the camera because only one camera is required. Last but not least,

	Initialization	Refinement	Shortcomings
Category 1: [81, 45]	Manual guess [81], Initialization for the housing parameters is not shown [45]	Nonlinear optimization	Need to estimate relative camera pose, SIFT matches come from the regions viewed by both cameras only, and the results using real data are not compared to the ground truth.
Category 2: [1, 90]	Solving linear equations	Nonlinear optimization	Require calibration object.
The new method	Solving linear equations and closed-form solution	Nonlinear optimization	See text for details.

Table 5.3: Comparison of different methods.

the initialization is more accurate than using manual guess in [81]. The nonlinear optimization is more efficient compared to that in [81]. The method in [45] claims that it can solve three different problems. However, the initialization for the housing parameters is not shown. As well, in their simulations, the results of the estimated the housing parameters are not shown or evaluated.

The new method is a major extension based on [90]. It produces results with similar accuracy comparing to that in [90], which means that it is more accurate

than that in [1]. Comparing to [90], the major advantage of the new method is that calibration target is not required. In particular, the novelty is to exploit triple wavelength dispersion, which not only requires no calibration target but also provides a closed-form solution to the interface distance. The new method of projecting red, green and blue patterns to the object is also unique, which removes the requirement for an object of known geometry to emit light with three different wavelengths. Such an implementation is obviously better than the custom-built light box [90]. As well, the new method can perform 3D reconstruction from a single camera where no previous methods can.

There is a limitation of this method, which is demonstrated in Fig. 5.25. In particular, when the ray connecting the camera center and the object point is perpendicular to the refractive interface, then this ray undergoes no refraction at all. In this case, there is no dispersion observed which means that the new method does not apply. However, notice that there is only one point in the entire field of view (FOV) of the camera where this would happen. In Figure 5.25, assume that the dispersion of all the points within the green circle is less than one pixel in the image, and it can be demonstrated that this green circle is very small comparing to the FOV of the camera. The configuration of the simulated experiments is used and assume that the z component of all the object points is 1.2 units. In this case, the FOV of the camera is about 0.9×0.6 units. The radius of the green circle is 0.042 unit, which means that there is about 1.03% of the FOV of the camera where the dispersion is less than one pixel. Obviously this area is very small.

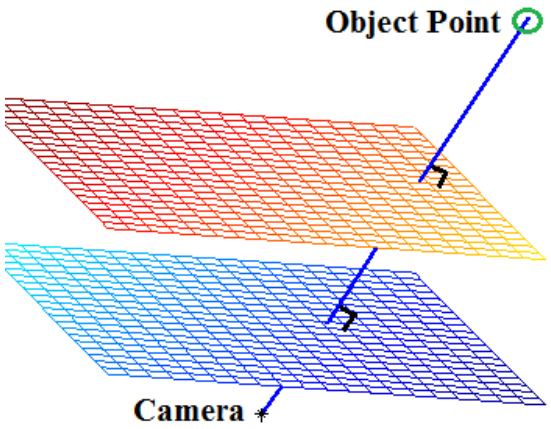


Figure 5.25: Demonstration of the limitation of the presented method.

Chapter 6

Undersea Camera System

In this chapter, the design and development of an undersea camera system is presented. The goal of this system is to provide a 3D model of the undersea habitat in a long-term continuous manner. The most important feature of this system is the use of multiple cameras and multiple projectors, which is able to provide accurate 3D models with an accuracy of a millimeter. By introducing projectors in the system, many different structured light methods can be used for different tasks. There are two main advantages comparing this system with the existing underwater imaging systems that are described in Section 2.3. First, this system can provide continuous monitoring of the undersea habitat. Second, this system has a low hardware cost. Comparing to existing deployed camera systems, the advantage of this system is that it can provide accurate 3D models and provides opportunities for future development of innovative algorithms for undersea research. In our system, we do not consider the poor-visibility conditions such as the ones mentioned in [36, 60].

The goal of this system is to provide a 3D model of the undersea habitat from 2D image. To achieve that, the system needs an image capture module and a control module. The image capture module is in charge of taking images from different views while the control module is in charge of streaming and collecting data from each camera.

6.1 Image Capture Module

The most important feature of this system is the use of multiple projectors and multiple cameras to build a 3D model of the undersea environment using 8 2D images of projected patterns with a structured light method. The structured light method is chosen because of its high accuracy, which can typically limit the error to within a millimeter. The method requires at least one projector and one camera. Multiple camera and multiple projectors are used so that the accuracy is even higher, and the observed volume is larger. Using projectors in the system is a critical design for future research opportunities. Because projectors can be used to project different structured light patterns for accurate 3D object reconstruction. There are many structured light methods that can be used in the developed system [28]. Some of the methods can be used to capture dynamic events, while others can achieve extremely high accuracy. In other words, this system is a framework that can be used to serve multiple purposes.

Figure 6.1 shows the diagram of the image capture module. There are 8 cameras and 3 projectors arranged in a full circle, and all the devices are aiming at a common point. The devices are arranged so that the system can observe as much information

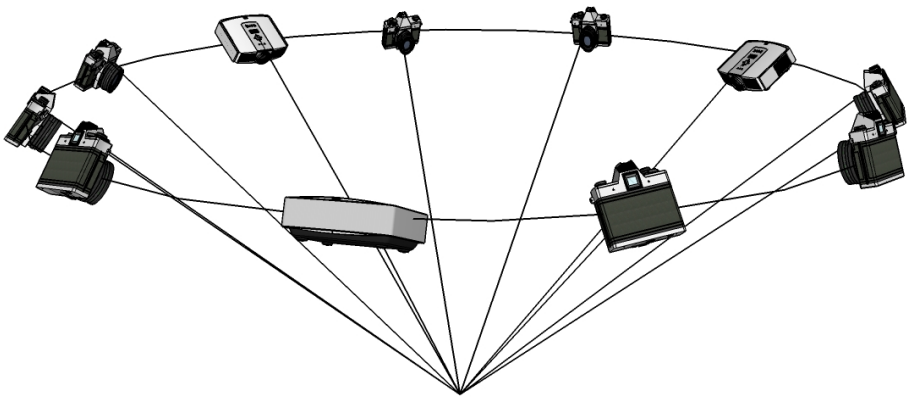


Figure 6.1: Diagram of the image capture module.

of the habitat as possible. Each camera used in the system is a Point Grey Flea3 GigE color camera with 5M pixels, where the data can be streamed through the Ethernet. This camera is selected because it can be easily controlled through the Ethernet and the image buffer can be transferred quickly. Since each device is placed in a watertight housing, ideally the device should generate no or very low heat. It is the main reason that the it also is selected, which is a pico data projector that can operate without a fan. Another reason is that TI Lightcrafter can be controlled via the Ethernet for projecting structured light patterns, which is consistent with the camera. It can potentially achieve a very high frame rate at over 1000 fps.

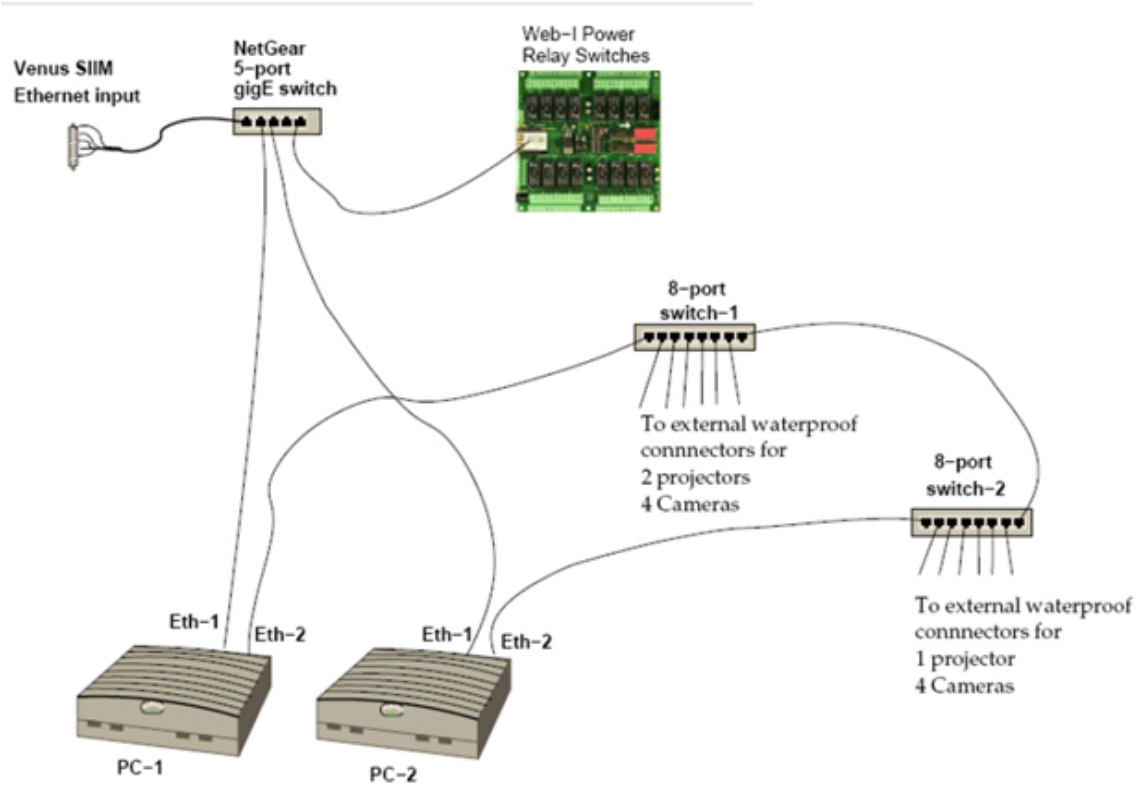


Figure 6.2: Diagram of the control module.

6.2 Control Module

Figure 6.2 shows the diagram of the control module. There are two PCs for redundancy in charge of controlling the image capture module. One of the PC serves as the main operating PC and the other one serves as a backup in case of hardware failure of the main one. In particular, the PC is running custom designed software that controls the TI Lightcrafters to project structured light patterns and the cameras to capture images simultaneously, as well as to collect buffered data from the cameras. In the design, the PC is carefully selected to satisfy several requirements.

First of all, the PC needs to communicate with the image capture module through the Gigabit Ethernet. It also needs to communicate with the server in order to upload the data. As well, the PC should be fast. In this case, a PC that has at least two Gigabit Ethernet ports is selected, one for each of the above functions so that the two separate communication channels will not interfere with each other. Another important requirement of the PC is that it must run without a fan. The cooling for the hardware such as the CPU should be air cooled and very efficient. As a result, the Intense PC [1] is selected because it fits all of the requirements. It is a fanless PC with an Intel i7 CPU, 8GB memory, two 1Gb Ethernet ports, and 500GB of solid state hard drive. In the diagram, one can see that one of the Ethernet ports of the PC (Eth-2) is connected to two 8-port Ethernet switches, which connect 8 cameras and 3 projectors together. Another Ethernet port (Eth-1) is connected to a 5-port switch which connects to an on-shore computer as well as to a Web-I power relay board. Using the Web-I relay board is another unique design of this system. In particular, the relay board is software programmable and has 16 ports acting as power switches for the 8 cameras, 3 projectors, 2 PCs and 2 8-port switches. With this design, the on-shore computer can be used to turn on or off any device in the system using a program. Moreover, once one of the PCs is turned on, programs can be used to turn on or off any device using the undersea PC, which gives more control on the image capture module. For example, one may not want the projector to be turned on at all time because the light will attract fish.

[1] http://www.fit-pc.com/web/products/intense-pc/

6.2.1 Housing

All the cameras and projectors are each mounted in a watertight housing equipment. The entire control module is mounted in a watertight housing equipment. The original housings of the PC's and the switches are removed before mounting in the housing. The left image in Figure 6.3 shows a camera and its housing before the camera is sealed inside. The housing equipment is tested to guarantee that it can resist a pressure of 100m depth of water. The right image shows a camera, a projector and their housings.

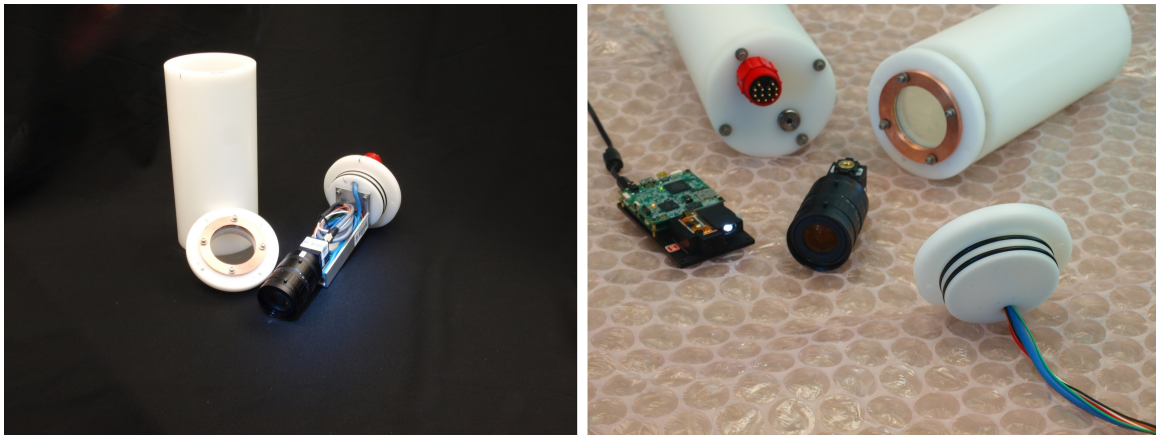


Figure 6.3: Left: A camera and its housing. Right: A camera, a projector and their housings.

Figure 6.4 shows the system before assembling. In particular, the 8 cameras and 3 projectors are put inside the housing and standing next to the connectors. The control module and its housing are also shown in the figure. In this figure, only one side of the control module can be seen. On this side, there is one PC, the relay board and

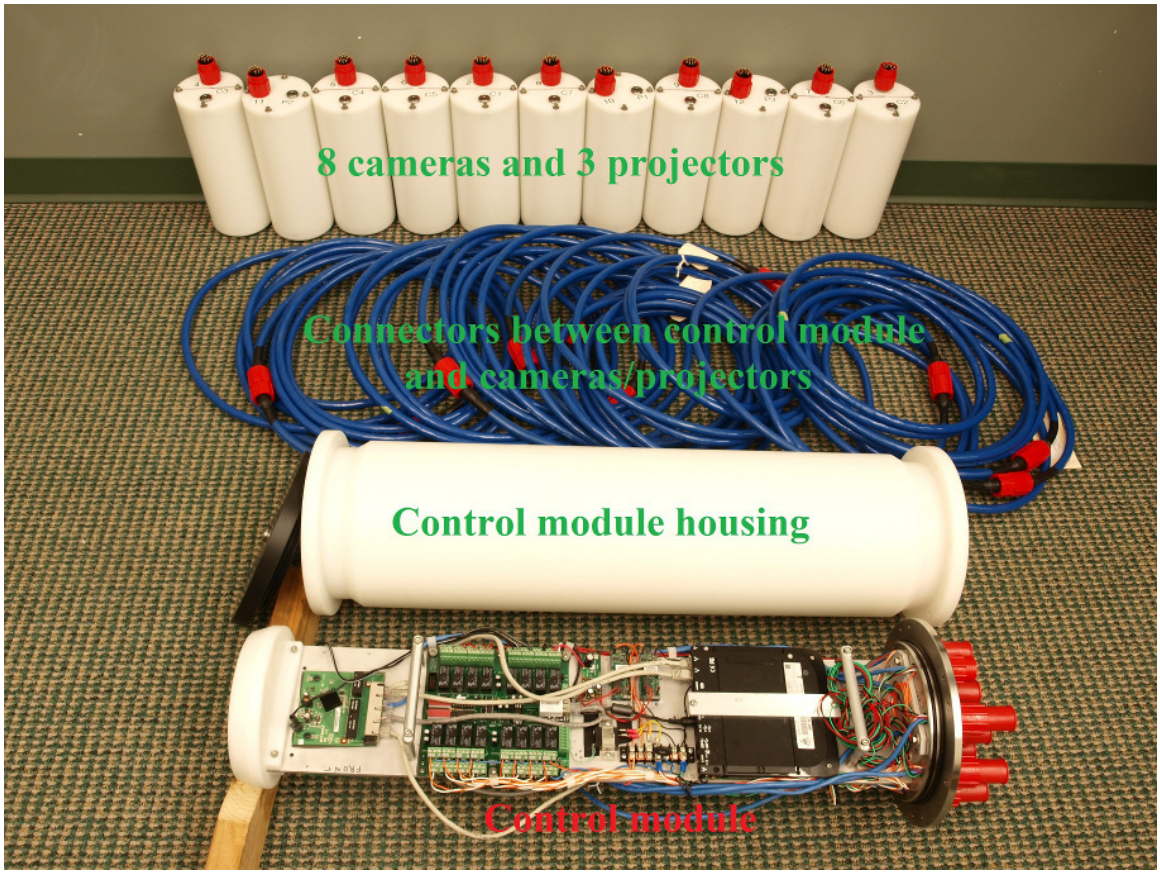


Figure 6.4: The system before its final assembly.

the 5-port switch attached. On the other side, the other PC, the two 8-port switches are attached. With this design, the space usage is best used. Figure 6.5 shows all the components of this system after they have been mounted on a supporting frame and is ready to be deployed undersea.

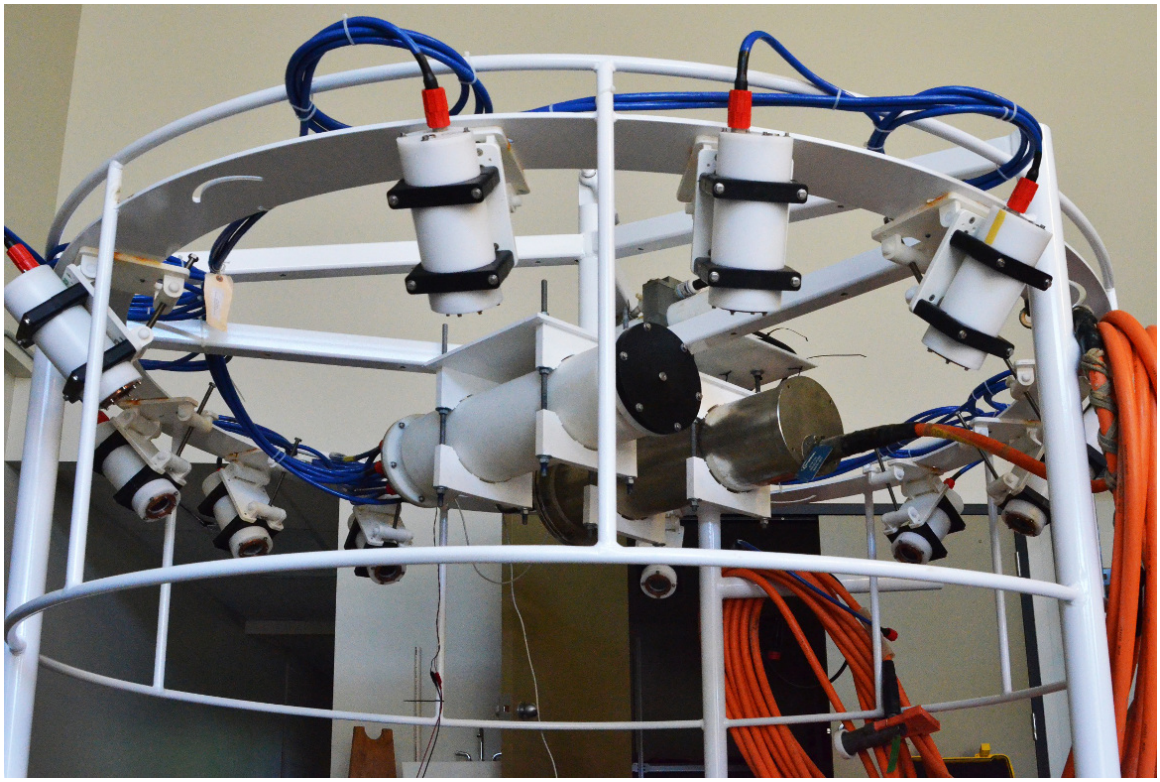


Figure 6.5: The system before it is deployed undersea.

6.3 Software Development

The software of this system includes two parts: system calibration and 3D reconstruction.

6.3.1 System Calibration

In order to obtain accurate 3D models for the underwater habitat, accurate system parameters are needed which are obtained by system calibration. In particular, there are two sets of parameters that are required for 3D reconstruction. The first includes

the camera intrinsic parameters and the relative pose of cameras. The second consists of the refractive interface normal and the distance from the camera center to the refractive interface of each camera. Both sets of parameters can be obtained when the system is built and before it is deployed undersea. The second set is used to accommodate for water refraction so that the 3D reconstruction result is more accurate.

6.3.2 3D Reconstruction

The structured light method is used for 3D reconstruction of the undersea habitat, to be more specific, the gray code method [28]. After 3D reconstruction, the captured images as well as the 3D model are uploaded to a data server instead of storing in the PC.

6.4 Results

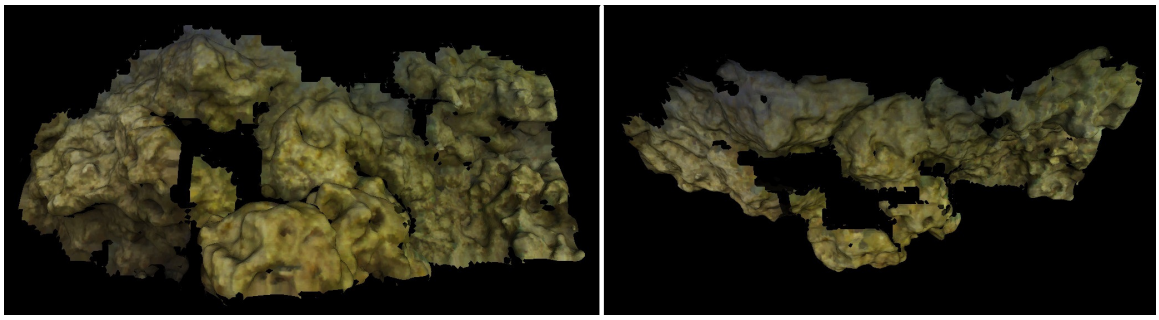


Figure 6.6: 3D reconstruction results of a coral reef in the lab: front view (left) and top view (right).

The prototype of this system was tested in the lab. In particular, Figure 6.6 shows the 3D reconstruction result of a coral reef placed in water using the prototype. The left image shows the result from the front view and right image from the top view. One can see that many surface details are captured in the result using the structured light method. Figure 6.7 shows the result using a passive multiview stereo method [27]. One can see that the structured light method is able to provide much better result than the result using the passive method [27].

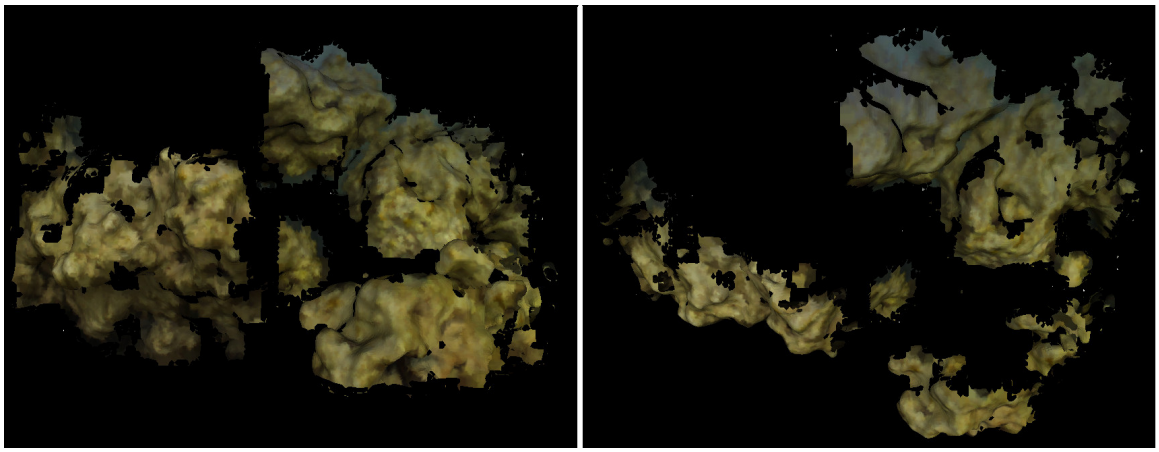
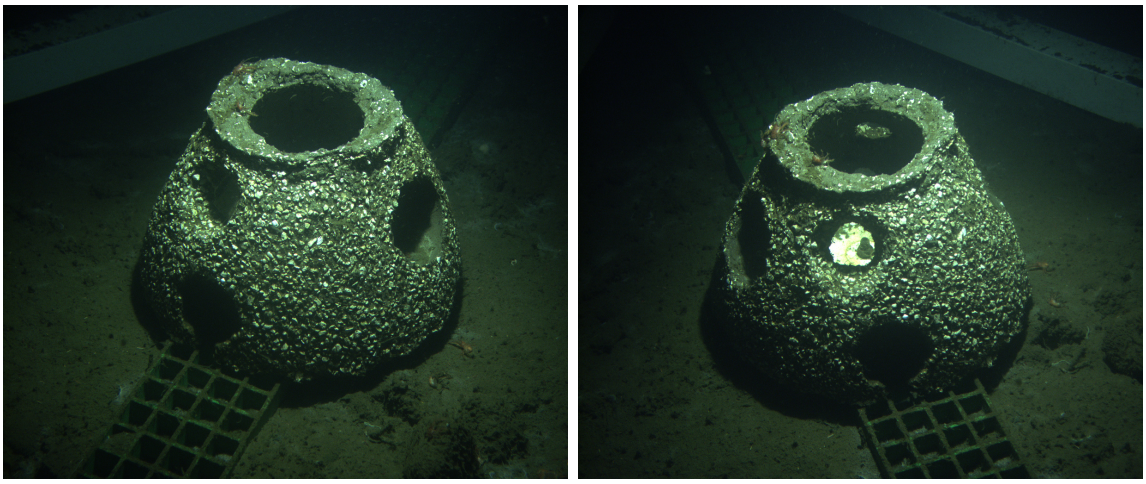


Figure 6.7: 3D reconstruction results using PMVS2 [27]. Left: front view. Right: Top view.

The system starts capturing images once it is deployed undersea. Figure 6.8(a) shows images captured in Sept. 2015 by two different cameras. Figure 6.9 compares the 3D reconstruction results of the structured light method with that of using the method presented in [27], using the dataset that was captured in Sept. 2015. It is obvious that structured light method generates more complete results since the result

produced by [27] contains only part of the object.



(a) (b)

Figure 6.8: Images captured by two different cameras.

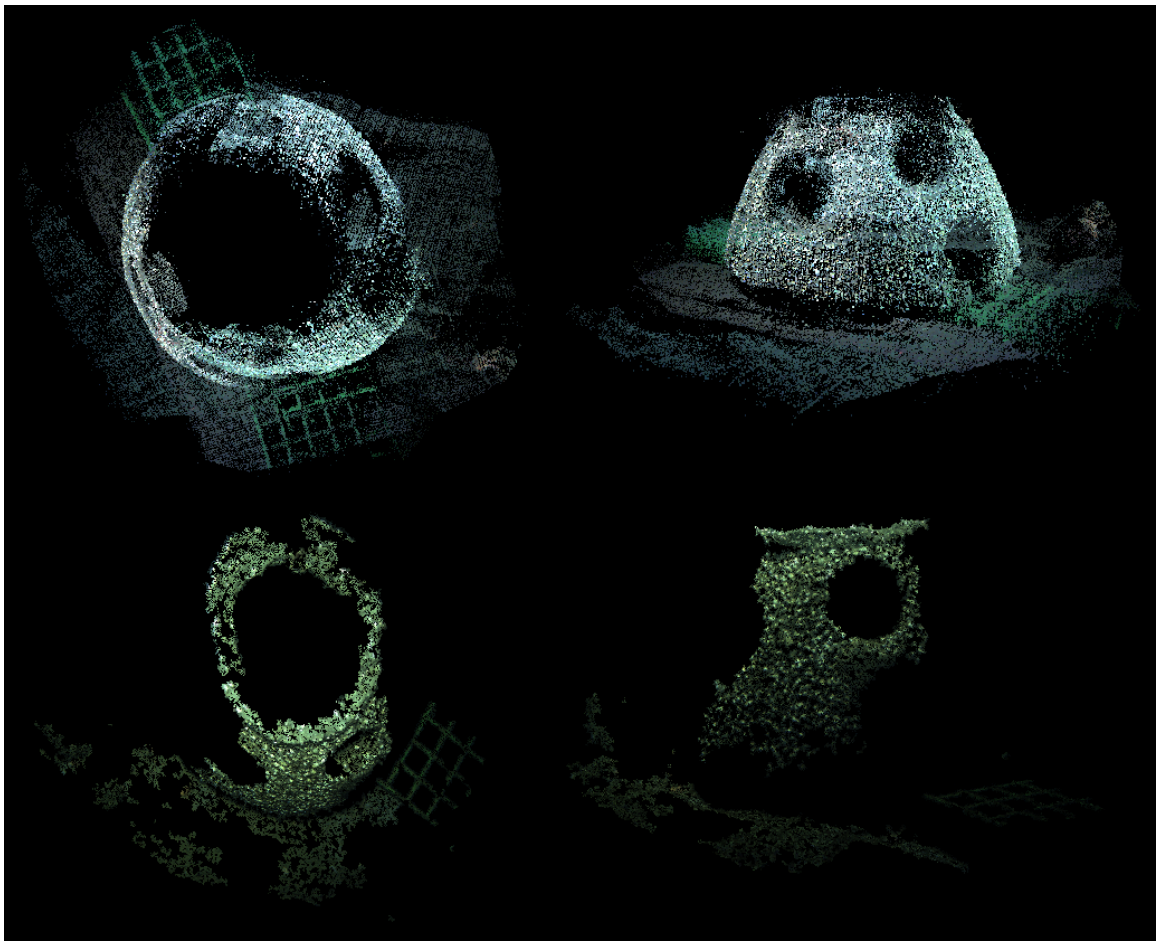


Figure 6.9: Top row: 3D reconstruction result by the structured light method. Bottom row: result by using the method in [27].

Chapter 7

Conclusion and Future Work

This thesis focuses on the topic of 3D object reconstruction, which has been a popular research topic in computer vision for many years. Several new methods have been developed in this thesis in two important areas, which are establishing correspondences and underwater camera calibration.

7.1 Contributions

In Chapter 3, a new structured light pattern is designed to recover dense depth map. The design of this pattern is based on an important finding that a Gaussian blurred De Bruijn pattern preserves the desirable windowed uniqueness property. With the newly designed pattern, dense correspondences can be established instead of sparse correspondences using a typical single-shot structured light pattern. The method can be applied to dynamic scene when a typical multi-shot structured light method

cannot. The experimental results demonstrate that the new method produces results with high accuracy.

A new structured light 3D reconstruction method is presented in Chapter 4. It can be applied to scenes in the presence of global illumination such as inter-reflection, subsurface scattering and severe projector defocus. Important properties that are embedded in the binary code patterns are used to determine the minimum stripe width that can eliminate the effects of projector defocus and subsurface scattering in the capture stage. After that, an iterative approach is designed to detect and correct errors due to inter-reflection. The experiments are designed to include real-word scenes with various characteristics, and the results demonstrate the accuracy of this method by quantitative evaluation, as well as robustness by qualitative evaluation. Both quantitative and qualitative evaluation show that the new method is better than two of the currently known best methods [35, 61].

In Chapter 5, a novel method to estimate the housing parameters for stereo cameras is presented first. Its novelty is based on an important observation that the layer thickness can be recovered once the refractive normal is known. Based on that, the solution to the layer thickness can be formulated into a set of linear equations. In terms of the refractive normal, either binary search or the dispersion constraint [90] can be used to solve it. Another novel method is presented by making use of triple wavelength dispersion. A closed-form solution is presented to compute the interface distance when the refractive normal is known. The result enables a new method for 3D reconstruction using a single camera. An extensive set of experiments are designed

to test the robustness of the new methods to noise and outliers. All the experimental results are evaluated against the ground truth, and most of them are compared with state-of-the-art methods [1, 90]. The experimental results demonstrate that the new methods provides very good results.

In Chapter 6, an undersea system that provides 3D reconstruction of the undersea habitat is presented. This system provides monitoring of the undersea habitat in a long term continuous manner. In particular, it can capture images in a pre-defined time interval and the collected data is uploaded to a data server where it is shared with many researchers.

7.2 Future Work

The future work of this thesis includes two major directions. First, the method presented in Chapter 3 requires the images captured by stereo cameras to be rectified first. However, the stereo rectification is still an open problem for underwater images. The reason is that the typical epipolar geometry only works for stereo cameras in air, but not underwater. In other word, this method cannot be applied to underwater images directly unless the underwater image rectification can be solved.

Another future direction is to improve the second part of Chapter 4 which is the iterative approach to detect and correct errors due to inter-reflection. The iterative approach is an online processing procedure which could be inefficient when the error is large. The best way to improve it is to design an offline procedure, or a new set of patterns that can accommodate the error.

Bibliography

[1] A. Agrawal, S. Ramalingam, Y. Taguchi, and V. Chari. A theory of multi-layer flat refractive geometry. In *Computer Vision and Pattern Recognition*, pages 3346–3353. IEEE, 2012. xiii, 5, 29, 33, 34, 35, 93, 95, 97, 106, 107, 131, 132, 133, 149

[2] C. Albitar, P. Graebling, and C. Doignon. Design of a monochromatic pattern for a robust structured light coding. In *Proceedings of the International Conference on Image Processing*, pages 529–532, 2007. 3, 10, 23

[3] D. G. Aliaga and Y. Xu. Photogeometric structured light: A self-calibrating and multi-viewpoint framework for accurate 3d modeling. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, AK, 2008. 3, 10, 11

[4] D. G. Aliaga. and Y. Xu. A self-calibrating method for photogeometric acquisition of 3d objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):747–754, 2010. 3, 10, 11

[5] R. A. Armstrong, H. Singh, J. Torres, R. S. Nemeth, A. Can, C. Roman, R. Eustice, L. Riggs, and G. Garcia-Moliner. Characterizing the deep insular shelf coral reef habitat of the hind bank marine conservation district (us virgin islands) using the seabed autonomous underwater vehicle. *Continental Shelf Research*, 26(2):194–205, February 2006. 6

[6] A. Baker. *Transcendental Number Theory*. Cambridge University Press, 1974. 43

[7] C. R. Barnes, M. Best, F. Johnson, L. Pautet, and B. Pirenne. Challenges, benefits, and opportunities in installing and operating cabled ocean observatories: Perspectives from neptune canada. *IEEE Journal of Oceanic Engineering*, 38, 2013. 38

[8] D. Bergmann. New approach for automatic surface reconstruction with coded light. In *Proceedings of Remote Sensing and Reconstruction for Three-Dimensional Objects and Scenes*, pages 2–9, San Diego, CA, 1995. 10, 11

[9] B. J. Boom, P. X. Huang, C. Spampinato, S. Palazzo, J. He, C. Beyan, E. Beauxis-Aussalet, J. Ossenbruggen, G. Nadarajan, Y. H. Chen-Burger, D. Giordano, L. Hardman, F.-P. Lin, and R. B. Fisher. Longterm underwater camera surveillance for monitoring and analysis of fish populations. In *Int. Workshop on Visual observation and Analysis of Animal and Insect Behavior*, 2012. 38

[10] J. Y. Bouguet. Camera calibration toolbox for matlab. 28

[11] C. Brauer-Burchardt, M. Heinze, I. Schmidt, P. Kuhmstedt, and G. Notni. Compact handheld fringe projection based underwater 3d-scanner. In *Underwater 3D Recording and Modeling*, pages 33–39, Piano di Sorrento, Italy, 2015. 38

[12] D. Brown. Close-range camera calibration. *PHOTOGRAMMETRIC ENGINEERING*, 37(8):855–866, 1971. 27

[13] M. S. Brown, P. Song, and T.-J. Cham. Image pre-conditioning for out-of-focus projector blur. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1956–1963, 2006. 54

[14] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–140, May 1990. 27

[15] D. Caspi, N. Kiryati, and J. Shamir. Range imaging with adaptive color structured light. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):470–480, 1998. 3, 10, 14

[16] Y.-J. Chang and T. Chen. Multi-view 3d reconstruction for scenes under the refractive plane with known vertical direction. In *International Conference on Computer Vision*, pages 351–358, 2011. xiii, 29, 30, 32

[17] V. Chari and P. Sturm. Multiple-view geometry of the refractive plane. In *British Machine Vision Conference, London, UK*, 2009. 29

[18] X. Chen and Y.-H. Yang. Two-view camera housing parameters calibration for multi-layer flat refractive interface. In *CVPR*, pages 524–531, 2014. 29, 130

[19] V. Couture, N. Martin, and S. Roy. Unstructured light scanning to overcome interreflections. In *Proceedings of the 2011 International Conference on Computer Vision*, pages 1895–1902, 2011. 26

[20] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63:542–567, 1996. 49

[21] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):296–302, Feb. 2005. 3, 10, 26

[22] D. Desjardins and P. Payeur. Dense stereo range sensing with marching pseudo-random patterns. In *Proceedings of the Fourth Canadian Conference on Computer and Robot Vision*, pages 216–226, 2007. xii, 3, 24, 25

[23] P. Fechteler and P. Eisert. Adaptive color classification for structured light systems. In *Proceedings of the 15th International Conference on Computer Vision and Pattern Recognition Workshop (CVPR2008)*, Anchorage, Alaska, USA, 27th June 2008. 21

[24] P. Fechteler and P. Eisert. Adaptive colour classification for structured light systems. *IET Journal on Computer Vision*, 3(2):49–59, June 2009. 21

[25] R. Ferreira, J. Costeira, and J. A. Santos. Stereo reconstruction of a submerged scene. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 102–109, 2005. 29

[26] H. Fredricksen. The lexicographically least debruijn cycle. *Journal of Combinatorial Theory*, 9(1):509–510, 1970. 19, 41

[27] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1362–1376, 2010. xviii, 144, 145, 146

[28] J. Geng. Structured-light 3d surface imaging: a tutorial. *Advances in Optics and Photonics*, 3(2):128–160, 2011. 136, 143

[29] D. C. Ghiglia and M. D. Pritt. *Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software*. John Wiley and Sons, Inc, 1998. 18

[30] L. Goddyn and P. Gvozdjak. Binary gray codes with long bit runs. *ELECTRONIC JOURNAL OF COMBINATORICS*, 10, 2003. 10, 13

[31] H. G. Greene, D. S. Stakes, D. L. Orange, J. P. Barry, and B. H. Robison. Application of a remotely operated vehicle in geologic mapping of monterey bay, california, usa. In *Proceedings of the American Academy of Underwater Sciences (13th annual Scientific Diving Symposium)*, 1993. 37

[32] P. M. Griffin, L. S. Narasimhan, and S. R. Yee. Generation of uniquely encoded light patterns for range data acquisition. *Pattern Recognition*, 25(6):609–616, 1992. 10, 22

[33] G. Griffiths. *Technology and Applications of Autonomous Underwater Vehicles*. CRC Press, 2002. 37

[34] J. Guhring. Dense 3-d surface acquisition by structured light using off-the-shelf components. In *Proceedings of Videometrics and Optical Methods for 3D Shape Measuring*, pages 220–231, 2001. 10, 11

[35] M. Gupta, A. Agrawal, A. Veeraraghavan, and S. G. Narasimhan. Structured light 3d scanning in the presence of global illumination. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 713–720, 2011. xv, 4, 10, 13, 54, 76, 77, 79, 80, 81, 82, 83, 84, 85, 148

[36] M. Gupta, S. G. Narasimhan, and Y. Schechner. On controlling light transport in poor visibility environments. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008. 39, 135

[37] M. Gupta, Y. Tian, S. Narasimhan, and L. Zhang. (de) focusing on global light transport for active scene recovery. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 2969–2976, 2009. 4

[38] M. Heesemann, T. Insua, M. Scherwath, S. K. Juniper, and K. Moran. Ocean networks canada: From geohazards research laboratories to smart ocean systems. *Oceanography*, 27:151–153, 2014. 38

[39] C. Hermans, Y. Francken, T. Cuypers, and P. Bekaert. Depth from sliding projections. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1865–1872, 2009. 4, 16

[40] https://en.wikipedia.org/wiki/Kinect. Kinect. 19, 27

[41] http://vision.middlebury.edu/stereo/. Middlebury stereo vision page. 2, 52

[42] P. S. Huang and S. Zhang. Fast three-step phase-shifting algorithm. *Applied Optics*, 45(21):5086–5091, 2005. 10, 17

[43] P. S. Huang, S. Zhang, and F.-P. Chiang. Trapezoidal phase-shifting method for 3-d shape measurement. *Optical Engineering*, 44(12), 2005. 10, 17

[44] I. Ishii, K. Yamamoto, K. Doi, and T. Tsuji. High-speed 3d image acquisition using coded structured light projection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 925–930, 2007. 10, 11

[45] A. Jordt-Sedlazeck and R. Koch. Refractive structure-from-motion on underwater images. In *ICCV*, pages 57–64, 2013. 34, 131, 132

[46] S. Juniper, M. Matabos, S. Mihaly, R. S. Ajayamohan, F. Gervais, and A. Bui. A year in barkley canyon: a time-series observatory study of mid-slope benthos

and habitat dynamics using the neptune canada network. *Deep Sea Research Part II: Topical Studies in Oceanography*, 92:114–123, 2013. 39

[47] L. Kang, L. Wu, and Y.-H. Yang. Experimental study of the influence of refraction on underwater three-dimensional reconstruction using the svp camera model. *Applied Optics*, 51(31):7591–7603, 2012. 29

[48] L. Kang, L. Wu, and Y.-H. Yang. Two-view underwater structure and motion for cameras under flat refractive interfaces. In *Proceedings of the 12th European Conference on Computer Vision*, pages 303–316, Berlin, Heidelberg, 2012. Springer-Verlag. 6, 33

[49] R. Kawahara, S. Nobuhara, and T. Matsuyama. A pixel-wise varifocal camera model for efficient forward projection and linear extrinsic calibration of underwater cameras with flat housings. In *ICCV Workshop*, pages 819–824, 2013. 36

[50] T. P. Koninckx and L. V. Gool. Real-time range acquisition by adaptive structured light. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):432–445, Mar. 2006. 3, 25

[51] T. P. Koninckx, P. Peers, P. Dutre, and L. V. Gool. Scene-adapted structured light. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 611–618, 2005. 25

[52] K. K. Ku, R. Bradbeer, and K. Lam. An underwater camera and instrumentation system for monitoring the undersea environment. In *Mechatronics and Machine Vision in Practice*, pages 167–179, 2008. 29

[53] J.-M. Lavest, G. Rives, and J.-T. Lapreste. Underwater camera calibration. In *European Conference on Computer Vision*, pages 654–668, 2000. 29

[54] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The Digital Michelangelo Project: 3D scanning of large statues. In *Proceedings of ACM SIGGRAPH 2000*, pages 131–144, July 2000. 1

[55] M. I. A. Lourakis and A. A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical report, 2004. 94

[56] J. L. Moigne and A. M. Waxman. Structured light patterns for robot mobility. *IEEE Journal of Robot Automation*, 4(5):541–548, 1988. 10, 22

[57] G. K. Moore. Satellite remote sensing of water turbidity. *Hydrological Sciences*, 25(4):407–421, 1980. 6

[58] R. A. Morano, C. Ozturk, R. Conn, S. Dubin, S. Zietz, and J. Nissanov. Structured light using pseudorandom codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):322–327, Mar. 1998. 24

[59] H. Morita, K. Yajima, and S. Sakata. Reconstruction of surfaces of 3-d objects by m-array pattern projection method. In *Proceedings of the Second International Conference on Computer Vision*, pages 468–473, 1988. 10, 22

[60] S. G. Narasimhan and S. Nayar. Structured light methods for underwater imaging: light stripe scanning and photometric stereo. In *Proceedings of 2005 MTS/IEEE OCEANS*, volume 3, pages 2610 – 2617, September 2005. 39, 135

[61] S. Nayar and M. Gupta. Diffuse structured light. pages 1–8, 2012. xv, 4, 10, 17, 76, 77, 79, 80, 81, 82, 83, 84, 85, 148

[62] S. Nayar, G. Krishnan, M. Grossberg, and R. Raskar. Fast separation of direct and global components of a scene using high frequency illumination. In *ACM Transactions on Graphics*, pages 935–944, 2006. 12

[63] H. Ochimizu, M. Imaki, S. Kameyama, T. Saito, S. Ishibashi, and H. Yoshida. Development of scanning laser sensor for underwater 3d imaging with the coaxial optics. In *Proceeding of SPIE*, 2014. 29

[64] M. O'Toole, J. Mather, and K. N. Kutulakos. 3d shape and indirect appearance by structured light transport. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3246–3253, 2014. 27

[65] J. Pages, C. Collewet, F. Chaumette, and J. Salvi. An approach to visual servoing based on coded light. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, pages 4118–4123, 2006. 24

[66] J. Pages, J. Salvi, C. Collewet, and J. Forest. Optimised de bruijn patterns for one-shot shape acquisition. *Image and Vision Computing*, 23(8):707–720, Aug. 2005. 10, 20, 21

[67] J. Pages, J. Salvi, and J. Forest. A new optimised de bruijn coding strategy for structured light patterns. In *Proceedings of the Pattern Recognition, 17th International Conference*, pages 284–287, 2004. 10, 20, 21

[68] J. Pages, J. Salvi, and C. Matabosch. Robust segmentation and decoding of a grid pattern for structured light. In *Pattern Recognition and Image Analysis, First Iberian Conference*, pages 689–696, 2003. xii, 10, 23, 24

[69] A. P. Pentland. A new sense for depth of field. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(4):523–531, 1987. 54

[70] O. Pizarro, R. Eustice, and H. Singh. Relative pose estimation for instrumented, calibrated imaging platforms. In *Proceedings of Digital Image Computing: Techniques and Applications*, pages 601–612, 2003. 28

[71] J. L. Posdamer and M. D. Altschuler. Surface measurement by space-encoded projected beam system. *Computer Graphics and Image Processing*, 18(1):1–17, 1982. 10, 11

[72] J. P. Queiroz-Neto, R. L. Carceroni, W. Barros, and M. Campos. Underwater stereo. In *XVII Brazilian Symposium on Computer Graphics and Image Processing*, pages 170–177, 2004. 29

[73] P. Ralph, R. Gademann, A. Larkum, and U. Schreiber. In situ underwater measurements of photosynthetic activity of coral zooxanthellae and other reef-dwelling dinoflagellate endosymbionts. *Marine Ecology Progress Series*, 180:139–147, May 1999. 6

[74] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 438–446, 2002. 3, 10, 13

[75] F. Ruskey. The combinatorial object server. 20, 41

[76] Z. Sakr, H. J. W. Spoelder, and A. Moica. Object recognition using pseudo-random color encoded structured light. In *Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference*, volume 3, pages 1237–1241, 2000. 1

[77] J. Salvi, J. Pages, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37:827–849, 2004. xii, 23

[78] A. San-Martin, S. Orejanera, C. Gallardo, M. Silva, J. Becerra, R. Reinoso, M. C. Chamy, K. Vergara, and J. Rovirosa. Steroids from the marine fungus geotrichum sp. *Journal of the Chilean Chemical Society*, 53:1377–1378, 2008. 6

[79] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002. 51

[80] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 195–202, Madison, WI, 2003. 2, 3, 52, 127

[81] A. Sedlazeck and R. Koch. Calibration of housing parameters for underwater stereo-camera rigs. In *Proc. BMVC*, pages 118.1–118.11, 2011. 32, 34, 131, 132

[82] S.Nayar, H. Peri, M. Grossberg, and P. Belhumeur. A Projection System with Radiometric Compensation for Screen Imperfections. In *ICCV Workshop on Projector-Camera Systems (PROCAMS)*, Oct 2003. 79

[83] Y. Swirski, Y. Y. Schechner, B. Herzberg, and S. Negahdaripour. Stereo from flickering caustics. In *Proceedings of International Conference on Computer Vision*, pages 205–212, 2009. 39

[84] Y. Swirski, Y. Y. Schechner, B. Herzberg, and S. Negahdaripour. Caustereo: Range from light in nature. *Applied Optics*, 50:89–101, 2011. 27

[85] R. Valkenburg and A. McIvor. Accurate 3d measurement using a structured light system. *Image and Vision Computing*, 16:99–110, 1996. 10, 11

[86] C. Wall, R. Rountree, C. Pomerleau, and F. Juanes. An exploration for deep-sea fish sounds off vancouver island from the neptune canada ocean observing system. *Francis Deep-Sea Research Part I: Oceanographic Research Papers*, 83, 2014. 38

[87] M. Weinmann, C. Schwartz, R. Ruiters, and R. Klein. A multi-camera, multi-projector super-resolution framework for structured light. In *Proceedings of International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 397–404, 2011. 10, 13

[88] J. Xu, N. Xi, C. Zhang, Q. Shi, and J. Gregory. Real-time 3d shape inspection system of automotive parts based on structured light pattern. *Optics and Laser Technology*, 43(1), 2010. 1

[89] Y. Xu and D. G. Aliaga. An adaptive correspondence algorithm for modeling scenes with strong interreflections. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):465–480, 2009. 4, 10, 12, 13

[90] T. Yau, M. Gong, and Y.-H. Yang. Underwater camera calibration using wavelength triangulation. In *CVPR*, 2013. xiii, 5, 35, 36, 109, 111, 123, 124, 126, 127, 130, 131, 132, 133, 148, 149

[91] M. Young, E. Beeson, J. Davis, S. Rusinkiewicz, and R. Ramamoorthi. Viewpoint-coded structured light. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007. 10, 26

[92] L. Zhang, B. Curless, and S. M. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *Proceedings of the 1st International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 24–36, Padova, Italy, 2002. xii, xiii, 10, 20, 21, 41

[93] L. Zhang, B. Curless, and S. M. Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 367–374, June 2003. 10, 26

[94] L. Zhang and S. K. Nayar. Projection Defocus Analysis for Scene Capture and Image Display. *ACM Transactions on Graphics*, Jul 2006. 4

[95] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: High-resolution capture for modeling and animation. In *ACM Annual Conference on Computer Graphics*, pages 548–558, August 2004. 10, 26

[96] S. Zhang. Recent progresses on real-time 3-d shape measurement using digital fringe projection techniques. *Optical Laser Engineering*, 48(2):149–158, 2007. 18

[97] S. Zhang and P. Huang. High-resolution, real-time 3d shape acquisition. In *Proceedings of on Computer Vision and Pattern Recognition Workshop*, pages 28–, 2004. 10, 18, 19

[98] S. Zhang and S. T. Yau. High-resolution, real-time absolute 3-d coordinate measurement based on the phase shifting method. *Optics Express*, 14(7):2664–2649, 2007. 10, 18, 54

[99] S. Zhang and S. T. Yau. High-speed three-dimensional shape measurement system using a modified two-plus-one phase-shifting algorithm. *Optical Engineering*, 46(11), 2007. 10, 17, 19

[100] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of International Conference on Computer Vision*, pages 666–673, 1999. 27, 28