Light Transport Acquisition and 3D Reconstruction in the Presence of Light Refraction

by

Yiming Qian

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

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

3D reconstruction is an important topic in both computer vision and computer graphics. Many techniques have been proposed for objects with Lambertian reflectance. It assumes that the reflected light from the object surface is uniformly distributed in all directions. However, light interacts with real-world objects in complex manners, *e.g.* refraction, scattering and specular reflection. By ignoring these effects, traditional methods, when applied directly, produce large errors. For example, due to light refraction, a transparent surface appears differently when observed from different viewpoints. Thus the traditional color/texture correspondence-based methods cannot be used. This dissertation presents novel hardware setups and software designs for 3D reconstruction in the presence of light refraction.

I start with capturing the light transport characteristics, *i.e.* the environment matte, of objects that are either refractive or reflective, or both. The proposed approach can locate the contributing light sources at the pixel level and render photo-realistic images of the object under novel illumination background.

Then I propose to exploit the light transport for reconstructing 3D shape of transparent and refractive objects. In particular, a novel imaging setup is built to capture the light rays before and after refraction. By introducing a novel normal consistency constraint that encodes the light refraction effect, I design an optimization procedure, which jointly reconstructs the 3D positions and normals of the object, as well as the refractive index.

I also present a new method to recovering 3D dynamic fluid surfaces by leveraging light refraction. Two cameras are used to capture the distortion of a random pattern through the wavy fluid surface. After estimating the correspondence between the captured image and the original pattern, I develop a refraction-based optimization framework for recovering the 3D shape and the refractive index of the fluid surface.

Finally, I consider the imaging scenario of viewing an underwater scene through a water surface. By explicitly accounting for light refraction at the water surface, I present a novel approach for simultaneously recovering the 3D shape of both wavy water surface and the moving underwater scene.

Preface

All methods presented in this thesis are published at the top venues in computer vision.

The work [54] of Chapter 3 is published as: Y. Qian, M. Gong, and Y.-H. Yang. Frequency-based environment matting by compressive sensing. In Proceedings of the IEEE International Conference on Computer Vision, pages 3532–3540, 2015.

The work [53] of Chapter 4 is published as: Y. Qian, M. Gong, and Y.-H. Yang. 3d reconstruction of transparent objects with position-normal consistency. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4369–4377, 2016.

The work [55] of Chapter 5 is published as: Y. Qian, M. Gong, and Y.-H. Yang. Stereo-based 3D reconstruction of dynamic fluid surfaces by global optimization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1269-1278, 2017.

The work [56] of Chapter 6 is published as: Y. Qian, Y. Zheng, M. Gong, and Y.-H. Yang. Simultaneous 3D reconstruction for water surface and underwater scene. In Proceedings of European Conference on Computer Vision, 2018.

This thesis is a concatenation of the above four papers.

Acknowledgements

First of all, I would like to acknowledge the support and help of my research advisors: Prof. Herbert Yang and Prof. Minglun Gong. I appreciate all their time, insight and encouragement. I never forget the stimulating weekly research discussions and the productive paper revisions. This thesis would not be possible without their advice.

I also thank my PhD committee members: Prof. Martin Jagersand, Prof. Pierre Boulanger, Prof. Nilanjan Ray, Prof. Yasutaka Furukawa and Prof. Ioanis Nikolaidis. I am very lucky to collaborate with Prof. Yinqiang Zheng in my fourth year. I appreciate their time and suggestions on this thesis.

I would like to thank the Faculty of Graduate Studies and Research, and the Department of Computing Science. Thanks to all staffs who helped me in the past four years. I am also grateful to my labmates in Prof. Herbert Yang's group for their invaluable help in this thesis. They are Xida Chen, Yi Gu, Xiaoqiang Zhang, G.M. Mashrur E Elahi, Juehui Fan, Bernard Llanos, Qing Cai, Zhao Pei, Yu Xia and Jia Lin. I thank my friends: Steve Sutphen, Li He, Xiaolong Wang, Xuebin Qin and Jian Wang. They not only made my PhD life fun and interesting, but also provided technical assistance for the hardware setups presented in this thesis.

I acknowledge the funding sources that support my PhD work. My work is partially supported by Natural Sciences and Engineering Research Council of Canada, Alberta Innovates and the University of Alberta.

Lastly, I thank all my family members for their unconditional and never-ending love.

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Background and Contributions 3
1.2.1 Environment Matting 3
1.2.2 3D Reconstruction of Static Transparent Objects . . . 4
1.2.3 3D Reconstruction of Dynamic Fluid Surfaces 5
1.2.4 Simultaneous 3D Reconstruction of Water Surface and Underwater Scene 6
1.3 Organization 7

2 Related Work **8**
2.1 Environment Matting 8
2.2 3D Reconstruction of Static Transparent Objects 10
2.3 3D Reconstruction of Dynamic Fluid Surfaces 11
2.3.1 Known Flat Underwater Scene 11
2.3.2 Unknown Non-Flat Underwater Scene 13

3 Frequency-Based Environment Matting **15**
3.1 Prerequisites 15
3.1.1 Problem Formulation 15
3.2 Proposed Approach 19
3.2.1 Sparsity under Frequency-based Formulation 19
3.2.2 Reconstruction via Compressive Sensing 20
3.2.3 Augment with Phase Information 22
3.3 Experiments 26
3.3.1 Synthetic Object 26
3.3.2 Real Transparent Objects 31

4 Reconstructing Static Transparent Objects **36**
4.1 Acquisition Setup and Procedure 37
4.2 Position-Normal Consistency 38
4.2.1 Normals from Positions by PCA 41
4.2.2 Normals by Snell's Law 41
4.3 Optimize Depth Maps and Refractive Index 42
4.4 Experiments 43
4.4.1 Synthetic Object 44
4.4.2 Real Refractive Objects 46

5 Reconstructing Dynamic Fluid Surfaces **53**
5.1 Correspondence Acquisition and Matching 54
5.2 Stereo-Based Reconstruction 55
5.2.1 Normal Definitions 56
5.2.2 Objective Function 57

5.3 Optimizing Depths and Refractive Index 58
5.4 Experiments 60
5.4.1 Synthetic Data 60
5.4.2 Real Dynamic Water Surfaces 64

6 Jointly Reconstructing Water Surface and Underwater Scene 69
6.1 Multi-View Acquisition Setup 69
6.2 Multi-View Reconstruction Approach 71
6.2.1 Normal Consistency at Reference View 71
6.2.2 Normal Consistency at Side Views 73
6.2.3 Solution Method 73
6.3 Experiments 76
6.3.1 Synthetic Data 76
6.3.2 Real Data 80

7 Conclusion and Future Work 86
7.1 Summary 86
7.2 Limitations and Future Work 87
7.2.1 Environment Matting 87
7.2.2 Static Transparent Object Reconstruction 88
7.2.3 Dynamic Fluid Surface Reconstruction 89

References 92

List of Tables

3.1 Comparison with the method in [19] on real data. Note that capturing more images will not improve the results of [19] due to the hierarchical sampling scheme being used. My results are more visually appealing, while consuming less processing time. Since the scenes are captured at 2fps in the video mode, the number of minutes needed for acquisition are computed as $\#imgs/120$. 33

3.2 Comparisons among different environment matting methods, where s denotes the sparsity of a signal. The information about the previous methods is directly copied or estimated from the corresponding papers. Note that the extraction time of [48] includes data acquisition because of its adaptive scheme. . . . 34

4.1 Reconstruction errors of the “ornament” and “ball” objects. Several planar facets are manually labeled as shown in the images above. Each facet is fitted using RANSAC with the inlier threshold of $0.1mm$. (a) and (b) show the results of the front and back surfaces of the ornament, respectively. (c) and (d) show the results of the front and back surfaces of the ball, respectively. 48

4.2 Reconstruction time of the three real objects. As introduced in Section 4.3, a coarse-to-fine optimization procedure is used in my approach. At the coarse stage, I downsample the acquired correspondences to 1/4 of the original resolution to estimate the refractive index. At the fine stage, I reconstruct the final shape using the full resolution correspondences and the estimated refractive index. 52

5.1 Running time for the three real wave sequences. 68

6.1 Reconstruction errors of the synthetic Bunny scene and the Dragon scene. Here, for each scene, I list the average errors by considering all frames. 77

6.2 Average running time of the three real scenes. 85

List of Figures

1.1 A transparent prism observed at two different viewpoints. Notice that the same surface point of the prism has different colors at the two viewpoints, making the color appearance being an unreliable cue for correspondence-based 3D reconstruction of transparent objects. 2

3.1 A physical setup for capturing environment mattes. 16

3.2 (a) shows a transparent cylinder captured against a solid gray background. (b) shows the binary segmentation result. (c) shows the foreground color F of the object. The image looks black because $\mathbf{B} = 0$ (*i.e.* the monitor displays a black pattern and thus the illumination is turned off in the scene). (d) shows the estimated attenuation index ρ. 17

3.3 The object pixel C is contributed by two background pixels $\mathbf{B}(i)$ and $\mathbf{B}(j)$. By letting background pixels emit different frequency signals (*e.g.*, use pixel id as frequency value), the contributing sources can be obtained by analyzing the recorded signal in the frequency domain. 18

3.4 The Gini indices of the four objects used in this chapter. The resolution of the row-based and column-based pattern is set to 512×512 here for practical data acquisition. 20

3.5 A transparent goblet captured against four example patterns generated using Eq.(3.5) by setting $N = 1060, k = 2, \varphi_1 = 220°, \varphi_2 = 320°$. The top row shows two row-based patterns, whereas the bottom row shows two column-based patterns. The two columns show the captured images at the time instance $t = 5$ and $t = 15$, respectively. 23

3.6 The impact of phase region number k and measurement costs on both accuracy of the composited result (a) and matte extraction time (b). Note that log scale is used for both axes in (a). . . . 28

3.7 Comparison using synthetic data with ground truth (a). The result of [19] (b) is computed by capturing 40 images in the coarse level and 300 images in the fine level. It is blocky because of their square block assumption. My CS-based approach (c & d) uses 320 images and shows better performance in terms of both accuracy and matte extraction time. The red and blue boxes show zoom in views. 29

3.8 CS-based data acquisition on the Goblet object (under phase region $k = 2$). As the number of images increases, the result of CS-based approach improves. With 400 images, the result is visually comparable to the conventional DFT, which requires 2120 images. 30

3.9 Effectiveness of using phase for accelerating matte extraction for two real objects: Goblet and Saxophone. Both results show that setting $k = 4$ can lower the computation cost by several manifolds, especially when a large number of captured images (m) is used. The benefit of further increasing k to 8 is limited. 31

3.10 Handling dispersion. By processing the different color channels separately, my approach is able to render the rainbow phenomenon of the trophy. The greenish fringe around the checkerboard is due to chromatic lens aberration, which is not corrected in the experiment. 35

4.1 My data acquisition setup, where two cameras roughly face each other. Camera 1 is capturing data in this photo. Once Camera 1 is done, the monitor is moved to the other side of the object to serve as the light source for Camera 2. 37

4.2 My acquisition setup using a pair of cameras and one monitor as light source. Note that the monitor is moved to different positions during acquisition. 39

4.3 Reconstruction errors as a function of Gaussian noise level on the synthetic sphere under different refractive indices. 50 trials are performed under each setting. 44

4.4 Visual comparison between the ground truth and my results for the synthetic sphere under different noise levels. (a),(c),(e),(g) and (i) show the 3D point clouds colored with the PCA normals seen from three different viewpoints. Both the front and back points are plotted in the same coordinate system. The point clouds are colored with the corresponding PCA normal map. (b),(d),(f),(h) and (j) show the depth maps of the front and back surfaces. 45

4.5 Refractive index estimation for the synthetic sphere. Blue curves plot the reconstruction error Eq.(4.7) as a function of hypothesized refractive index. Red curves plot the corresponding objective function Eq.(4.3). Ground-truth indices are shown with vertical lines. 46

4.6 Reconstruction results of the “ornament” (top) and “ball” (bottom) objects; please refer to Table 4.1 for the photos. (a) and (c) show the 3D point clouds colored with the PCA normals seen from three different viewpoints. Both the front and the back points are plotted in the same coordinate. (b) and (d) show the depth maps of the front and the back surfaces. Note that some holes exist on the surface because no ray-ray correspondences are obtained for those regions. 49

4.7 Reconstruction results of the “bird” object. (a) shows the point cloud colored with the PCA normals seen from three viewpoints. (b) and (c) show the depth map of the front and the back surfaces. (d) shows the top view of the object. Because the positions of the three birds overlap, multiple refractions may happen between the camera and the light source. To avoid that, I cover the two smaller birds with tapes as shown in (e) and only reconstruct the larger bird for illustration. Note that the obtained back surface of the head of the larger bird is incomplete because the correspondences are not available in those complex regions. 50

4.8 Refractive index estimation for the "ornament" (a), "ball" (b) and "bird" (c) object. Blue curves plot the proposed reconstruction error term as a function of hypothesized refractive index. Green curves plot the corresponding objective function. The minimum of the blue curve is marked in red. 51

5.1 Acquisition setup (a) and the corresponding refraction stereo geometry (b). Note that Camera 3 in the left figure is for accuracy assessment only and not used during 3D reconstruction. . 54
5.2 Ray tracing geometrically to estimate the shape-based optical flow field. 59
5.3 Different error measures as a function of frame id for synthetic wave. (a) and (b) shows the error plots when the refractive index $\eta_l = 1.33$ is used in wave simulation. (c) and (d) shows the error curves when $\eta_l = 1.55$ is used in wave simulation. . 61
5.4 Evaluations on the *PCA* and *RightSnell* normals for synthetic wave. (a) and (b) show the error plots when the refractive index $\eta_l = 1.33$ is used in wave simulation. (c) and (d) show the error curves when $\eta_l = 1.55$ is used in wave simulation. Note that, for better visualization, the 10-base log scale is used for the vertical axes in (a) and (c). 62
5.5 Visual comparisons between the method in [45] and ours for an example frame when $\eta_l = 1.33$ is used for simulation. From top to bottom, it shows the *LeftSnell* normal map, the depth map and the point cloud colored with *LeftSnell* normals. Please see the supplemental videos [63] for the full video sequence as well as the captured images. 62
5.6 Average EPE Eq.(5.7) as a function of refractive index hypotheses using several example frames of synthetic data. The vertical dashed lines indicate the true indices. 63
5.7 Point clouds of adjacent frames of Wave 1 (top), Wave 2 (middle), Wave 3 (bottom). It shows that my results are visually temporal coherent. In this chapter, a point cloud is plotted based on its corresponding depth map and colored with *LeftSnell* normals. 64
5.8 View synthesis using an example frame of Wave 1 (top) and Wave 2 (bottom). The shading effects caused by reflection/caustics (red box) and motion blur effects (green box) can be observed in captured images (d). In (e), I compose the reconstructed 3D surface onto new scenes using the ray-tracing method as depicted in Fig. 5.2. 65
5.9 Quantitative evaluation on results generated under different constraints using Wave 1. The top figure plots average EPE as a function of frame ID. For better visualization, the 10-base log scale is used for the vertical axis. The bottom table shows the corresponding mean and standard deviation (stdev.) of EPE among all frames. 66
5.10 Quantitative evaluation on results generated under different constraints using Wave 1. The top figure plots MAE between binarized $\mathbf{I}_3$ and binarized $SBF(\mathbf{B}_3)$ as a function of frame ID. The bottom table shows the corresponding mean and standard deviation (stdev.) of MAE among all frames. Note that, here the evaluation results based on MAE are consistent with the ones based on average EPE as shown in Fig. 5.9. 67

5.11 Visual comparisons between the method of [45] and ours for an example frame of Wave 1 (top) and Wave 2 (bottom). Note that here I also impose a smoothness term in the algorithm of [45], *i.e.* Case 2, for fair comparisons. 68
5.12 Average EPE Eq.(5.7) as a function of refractive index hypotheses for real data. The vertical dashed line indicate the refractive index of water, *i.e.* 1.33. 68

6.1 Acquisition setup using a camera array (a) and the corresponding imaging model illustrated in 2D (b). The evaluation camera in (a) is for accuracy evaluation only and is not used for 3D shape recovery. 70
6.2 Discontinuity of underwater scene points. As indicated by the purple arrow, the red points are interlaced with the green points, although the red and green rays are each emitted from contiguous pixels. 75
6.3 Visual comparisons with GT on two example frames of the Bunny scene (left two columns) and the Dragon scene (right two columns). In each subfigure, I show the GT and my result in the top and bottom row, respectively. (a) shows the GT water surface depth and the estimated one. (b) shows the GT water surface colored with the GT normal map, and the computed one colored with the *Quadratic* normals. The *Snell* normals are not shown here because they are similar to the *Quadratic* normals. (c) shows the GT point set of the underwater scene and the recovered one, where each point is colored with its z-axis coordinate. The red boxes highlight an obvious different region of the underwater point clouds of two different frames; see text for details. 78
6.4 Different error measures as a function of the balancing parameter λ. 79
6.5 Different error measures as a function of the number of cameras used. 79
6.6 Reconstruction results of four example frames of my captured scenes. In each subfigure, I show the captured image of the reference camera (top), the point cloud of the water surface colored with the *Quadratic* normals (middle), the point cloud of the underwater scene colored with the z-axis coordinates (bottom). Note that the motion blur (green box) in the captured image may affect the reconstruction result (red box). 81
6.7 2D illustration of forward projection through the water surface. 82
6.8 View synthesis on two example frames (top and bottom) of Scene 3. From left to right, it shows the images captured using the evaluation camera, the synthesized images and the absolute difference maps between them. The effects of specular reflection (red box) and motion blur (green box) can be observed in the captured images. These effects cannot be synthesized, leading to higher differences in the corresponding areas. 84

7.1 Results of a failure case for a piece of myopia glass. Both front and back points are plotted in the same coordinate system. Since small normal perturbations on thin objects do not result in large ray correspondence changes, the estimated shape is quite noisy. Nevertheless, a slice through the center of the point cloud seen from the top (f) suggests that my approach properly models the thickness variation of the glasses. 90

Chapter 1

Introduction

1.1 Motivation

Although an image is a 2D array, we live in a 3D world [73]. 3D reconstruction refers to the process of inferring 3D shape of objects or scenes from 2D images, *i.e.* determining the 3D coordinates of surface points. As a core problem in both computer vision and computer graphics, 3D reconstruction has wide applications, *e.g.* digital heritage [40], medical diagnosis [79], object recognition [51].

Many techniques have been proposed for capturing the shape of opaque objects using either active [58], [59] or passive [15], [23] manners. However, existing methods usually assume that the object surface is Lambertian, by which the same amount of light is reflected from the surface in all directions. Such an assumption is violated for transparent objects, which, nevertheless, are commonly encountered in daily life, *e.g.* glasses, crystals, liquids. Transparent objects interact with light in complex manners including specular reflection, refraction and absorption. Therefore, 3D modeling methods tailored for Lambertian surfaces fail when they are used to recover the 3D shape of transparent objects.

Perceiving transparent objects is difficult, even for us — we often bump into clean glass doors, for example. Technically, the problem is challenging due to several reasons. Firstly, transparent surfaces do not have their own colors but acquire their appearance from surrounding backgrounds. Hence, traditional Lambertian-based 3D modeling methods, which rely on color con-

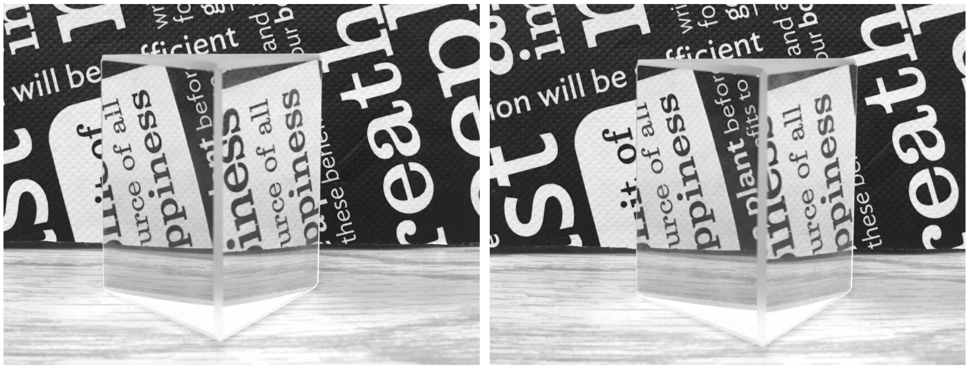


Figure 1.1: A transparent prism observed at two different viewpoints. Notice that the same surface point of the prism has different colors at the two viewpoints, making the color appearance being an unreliable cue for correspondence-based 3D reconstruction of transparent objects.

sistency across different views, cannot work for such view-dependent surfaces; see Fig. 1.1. Secondly, tracing the light path involved in light refraction in transparent surface reconstruction is non-trivial because of the non-linearity inherent in refraction. Even worse is that light refraction depends not only on the 3D shape but also on the medium's property, *i.e.* refractive index, which is usually unknown. Third, if the surface of interest is dynamic (*e.g.* water waves), the problem becomes harder because real-time data capture is required.

While 3D reconstruction of transparent surfaces has not been fully developed, there is much interest in using them if they are available. For example, if glasses and crystals are used in a cartoon, their real 3D models are essential to create a high-quality computer animation. On the other hand, the problem of accurately reconstructing 3D water surface have drawn much attention due to applications in oceanography and remote sensing.

Motivated by the limitations of existing methods and the demands for real-world applications, this dissertation focuses on 3D reconstruction of transparent surfaces, including static objects made of crystal or glass and dynamic fluid surfaces. I start by capturing the light transport characteristics of transparent objects. Then, instead of relying on the unreliable color appearance, I use the light transport as a key cue for 3D reconstruction. Specifically, compared to reflection, refraction is indeed an unique property for transparent surfaces and conveys important information of the 3D shape (For instance, according to Snell's law, light refraction paths are determined by surface normals). Hence, here the problem of shape recovery is tackled based on light refraction.

In the following sections, I present the background of the problems considered in this dissertation, as well as the contribution of my proposed methods.

1.2 Background and Contributions

1.2.1 Environment Matting

Environment matting is a technique introduced to render photo-realistic images of objects that are either refractive or reflective, or both. It focuses on modeling the object's light transport characteristics, *i.e.* the *environment matte*, and allows the object to be seamlessly composited into a new background. Typically, to obtain an environment matte, the object needs to be photographed in front of a series of pre-designed backdrops. How the object refracts and reflects light can then be inferred from the recorded images.

The concept of environment matting is first introduced in [82]. Since then, several methods [13], [19], [48], [71], [81] have been proposed to either simplify the data acquisition process or to improve the accuracy of the environment matte. The main task of environment matting is to decompose a *many-to-one* mapping, by which many background pixels are combined into one foreground pixel. Most existing methods decompose the mapping in the spatial domain, where a foreground pixel can be composited in an infinite number of ways. To handle the ambiguities, these methods use additional constraints and time-consuming non-linear optimization to estimate the mapping, whose physical correctness cannot be verified. A frequency-based approach is later proposed and is capable of finding the accurate contributing sources efficiently [81]. However, it requires a large number of captured images. On the other hand, extracting matte data from a single photo is possible if the object is perfectly specular transparent [13]. A recent spatial-domain method [19] also has low data acquisition requirement at the expense of high computational cost. Both of these two methods cannot provide high-accuracy environment mattes. Hence, there is a need for an effective algorithm that can accurately and quickly extract the matte data from a small number of images.

Motivated by the above observations, in this thesis (Chapter 3), I present a novel environment matting approach with the following objectives. First, my approach can accurately find the contributing sources at the pixel level. Second, my approach leverages on the recently developed theory of *Compressive Sensing (CS)* to reduce the complexity of data acquisition. Finally, I incorporate additional phase information into the frequency-based model, which further reduces the number of images required and significantly accelerates the process of environment matte extraction.

1.2.2 3D Reconstruction of Static Transparent Objects

Previous approaches of 3D transparent object reconstruction can be roughly classified into three groups [32], [33]: reflection-based, refraction-based, and intrusive methods. The first group attempts to reconstruct the objects by utilizing the specular highlights on the object surface [41], [44]. By analyzing only the surface reflection properties, such approaches can reconstruct transparent objects with complex and inhomogeneous interior. However, unlike opaque objects, only a small amount of light is reflected from the transparent object's surface. To measure the weak reflection, precise controlling and adjusting the light positions are usually required. The second group exploits the refraction characteristics of transparent objects. Many methods simplify the problem by considering only one-refraction events, either assuming that the surface facing away from the camera is planar [62] or that the object is thin [70]. As well, the refractive index is required to be known for surface normal estimation. Although the problem of two-refraction events has been investigated theoretically [38], the setup requires high precision movements of both the object and the light source, making the approach hard to use and the results difficult, if not impossible, to reproduce. Finally, intrusive methods either rely on special devices (*e.g.* diffuse coating [24]) or by immersing the object in special liquids [27], [30], [66], which are often impractical and may even damage the objects. Hence, there is a need for a practical approach that can accurately reconstruct transparent objects using a portable setup.

This thesis (Chapter 4) presents a new refraction-based approach for reconstructing homogeneous transparent objects, through which light is refracted twice. As is commonly done, inter-reflections within the object are assumed to be negligible. By introducing a novel normal consistency constraint, an optimization procedure is designed, which jointly reconstructs the 3D positions and normals at two refraction locations. The refractive index of the object can also be reliably estimated by minimizing a new reconstruction error metric. Further, my acquisition setup is simple and inexpensive, which consists of two cameras and one monitor, all of which do not require precise positioning.

1.2.3 3D Reconstruction of Dynamic Fluid Surfaces

In computer vision, the problem is usually solved via shape from refraction. Typically, a known background is placed beneath the fluid surface and 3D reconstruction is performed by analyzing pixel-point correspondences. That is, for each pixel, the corresponding location of the light source in the background is acquired. However, shape from pixel-point correspondence is known to have ambiguities: the 3D surface point can lie at any position along the camera ray that goes through the pixel. Recent methods resolve the ambiguities along two directions. Some methods [70], [74], [77], instead of using pixel-point correspondences, acquire *ray-ray* correspondences, *i.e.* the incident ray emitted from the background and the exit ray going to the camera, using special devices (*e.g.* Bokode [74], light field probes [70]). Alternatively, a number of methods [17], [45] propose to employ stereo/multiple cameras to capture the fluid surface, which basically utilize a cross-view normal consistency constraint: the normals computed using the pixel-point correspondences acquired from different views should be consistent. Nevertheless, for the above two groups, a common limitation is that they result in reliable normals only but not in depths. The final 3D points of the fluid surface are then obtained by normal integration. To get the boundary condition for integration, they either assume that the surface is flat at the boundary [17], [74] or the boundary is estimated using the noisy depths [45], [70].

To cope with the above limitations, this thesis (Chapter 5) presents an optimization-based approach to reconstruct a dynamic, homogeneous and transparent fluid surface, from which specular reflection is assumed to be negligible. My approach is based on pixel-point correspondences. By assuming that light is redirected only once through the fluid surface, I first use two perspective cameras to capture the distortion of a random pattern through the wavy surface. Hence, my acquisition system is easy to implement and requires no special optics. Compared to a conventional stereo-based method [45], the proposed approach can obtain both accurate, consistent depths and normals without the error-prone surface integration step. Specifically, rather than doing a point-by-point reconstruction, I formulate an optimization function, which exploits not only the cross-view normal consistency but also the single-view normal consistency constraints. By doing so, I jointly reconstruct both the depths and the normals. My method addresses the fundamental limitation of existing methods on surface integration without accurate boundary conditions. Besides, a new reconstruction error metric is designed to search the refractive index of liquid with very encouraging results.

1.2.4 Simultaneous 3D Reconstruction of Water Surface and Underwater Scene

As discussed in Section 1.2.3, existing methods for recovering dynamic water surfaces typically assume that the underwater scene is a known flat pattern, for which a checkerboard is commonly used [17], [45]. Meanwhile, most previous works reconstruct the underwater scene by assuming the interface between the scene and the imaging sensor is flat [3], [11], [21]. Therefore, the problems of reconstructing underwater scene and of reconstructing water surface are usually tackled separately in computer vision. Recently, Zhang *et al.* [76] make the first attempt to solve the two problems simultaneously using depth from defocus. Nevertheless, their approach assumes that the underwater scene is stationary and an image of the underwater scene with a flat water surface is available. Because of the assumptions of the flat water surface or the flat underwater scene, none of the above mentioned methods can be directly applied to solve the problem of jointly recovering the *wavy* water surface and the natural underwater *dynamic* scene. Indeed, the lack of any existing solution to the above problem forms the motivation of my work.

In this thesis (Chapter 6), I propose to employ multiple viewpoints to tackle such a problem. In particular, I construct a portable camera array to capture the images of the underwater scene distorted by the wavy water surface. My physical setup does not require any precise positioning and thus is easy to use. Following the conventional multi-view reconstruction framework for on-land objects, I first estimate the correspondences across different views. Then, based on the inter-view correspondences, I impose a normal consistency constraint across all camera views. Suppose that the light is refracted only once while passing through the water surface. I present a refraction-based optimization scheme that works in a frame-by-frame[1] fashion, and can handle the dynamic nature of both the water surface and the underwater scene. More specifically, my approach is able to return the 3D positions and the normals of a dynamic water surface, and the 3D points of a moving underwater scene simultaneously. Encouraging experimental results on both synthetic and real data are obtained.

1.3 Organization

The rest of this dissertation is organized as follows. Chapter 2 reviews previous works on light transport acquisition (*i.e.* environment matting) and 3D reconstruction for transparent objects. The four tasks discussed in Section 1.2 are introduced in Chapter 3, 4, 5, 6, respectively. For each problem, I first present the proposed approach, followed by the experimental results. Chapter 7 concludes the thesis and discusses future directions.

[1] In Chapter 6, a frame refers to the pictures captured from all cameras at the same time point.

Chapter 2

Related Work

2.1 Environment Matting

Zongker *et al*. [82] first formulate the problem and decompose the many-to-one mapping by assuming a foreground pixel is only contributed by a single rectangle area of the background. Chuang *et al*. [13] later propose two extensions: i) by sweeping different oriented Gaussian strips across the background to accommodate for sources that are not axis-aligned rectangles; ii) by extracting the environment mattes of colorless and pure specular objects using only one image. Wexler *et al*. [71] present a probabilistic model based method, which does not rely on predefined backdrops but requires enough sample images. It only works for thin transparent objects that do not introduce large optical distortion to the background. Inspired by image-based relighting, Peers and Dutr [48] use a set of wavelet basis images to obtain a visually pleasing result using a large number of sample images.

Inspired by the fact that a signal has a unique decomposition in the frequency domain, Zhu and Yang [81] propose a frequency-based approach which can find the accurate contributing sources, allowing the decomposition ambiguity to be alleviated. My proposed approach is built upon their frequency-based model, but uses CS to dramatically reduce the number of images required.

Note that using CS-based data acquisition for environment matting has been recently proposed by Duan *et al*. [19], [20]. My work differs from theirs in three main aspects: 1) Rather than solving the problem in the spatial domain, I work in the frequency domain, which helps to accurately locate the contributing sources; 2) The sparsity assumptions in CS are different. Their method assumes that the light transport vector is sparse and directly reconstructs it in the spatial domain, whereas I assume a foreground pixel is contributed by a sparse number of frequencies, not locations; 3) To reduce the computational cost, the hierarchical recovery scheme they proposed limits the contributions to a foreground pixel by only square blocks in the background. Hence, their composited results appear blurry and blocky. In contrast, my results are sharper and clearer because my approach can locate the contributing sources at the pixel level efficiently.

Compressive Sensing. Compressive sensing [9], [18] is an emerging field that provides a framework for reconstructing a sparse signal with far fewer measurements than the dimension of the signal. Instead of capturing the original N-dimensional signal x directly, to recover x, CS seeks to use $M < N$ linear measurements $y = Ax$, where A is an $M \times N$ measurement matrix, and x is an s-sparse signal, *i.e.* x contains at most $s \ll N$ nonzero elements. In CS, if the measurement matrix A satisfies the restricted isometry property (RIP) [8], then x can be stably recovered by solving the nonlinear optimization problem: $\min \|x\|_1$, s.t. $y = Ax$ with only $M = \mathcal{O}(s \log(N/s))$ measurements.

CS has facilitated the solving of many computer vision and graphics problems, either by helping in reformulating the problem using the sparsity constraint, *e.g.* face recognition [72], background subtraction [10], or by reducing the complexity of data acquisition, *e.g.* light transport acquisition [49], dual photography [61]. As mentioned before, CS is also incorporated into environment matting in the spatial domain [19]. Huang *et al.* [29] propose a CS-based solution for recovering data with both sparsity and dynamic group clustering priors. Note that the group clustering prior is also applicable to environment matting and has been utilized in [19] since the background pixels contributed to an object pixel appear in groups.

2.2 3D Reconstruction of Static Transparent Objects

Shape from reflection based methods utilize the specular property of the transparent surface. Such methods bypass the complex interactions of light with the object as it travels through the object by acquiring the linear reflectance field. Hence, inhomogeneous transparent objects can be reconstructed. Tarini *et al.* [65] acquire light reflections of mirror object against a number of known background patterns and then alternately optimize the depths and normals from reflective distortions. Morris and Kutulakos [44] reconstruct complex inhomogeneous objects by capturing exterior specular highlights on the object surface. Their approach requires delicate movements of light sources and imprecise movements often introduce errors to the results. Yeung *et al.* [75] introduce a low-cost solution by analyzing specular highlights, which can only obtain the normal map of an object. Recently, Liu *et al.* [41] apply a frequency-based approach to establish accurate reflective correspondences, but only sparse 3D points are obtained. A common issue of the above reflection-based approaches is that the reflection field is often corrupted by the indirect light transport within the object and various constraints are proposed to tackle it.

Refraction-based reconstruction methods rely on the refracted light, which is stronger than the reflected light for truly transparent objects and conveys unique characteristics of the objects.

Ben-Ezra and Nayar [6] develop a structure-from-motion based method for reconstructing the full 3D model of a transparent object, where the object is assumed to have a known parametric form. Wetzstein *et al.* [70] acquire the correspondences between the incident and exit rays, *i.e.* ray-ray correspondences, from a single image using light field probes and compute refraction positions through triangulation. However, their method assumes that the incident light is redirected once and thus, can only work for thin objects. Similarly, several methods [43], [62], [67] simplify the problem by focusing on only one-refraction events. In particular, they either assume that part of the surface information (*e.g.* normal) is known or that one of the refraction surfaces is known.

Kutulakos and Steger [38] categorize the refraction-based approaches based on the number of reflections or refractions involved, and discuss the feasibility of reconstruction of different cases. They show that at least three views are required to reconstruct a solid object where light is redirected twice. My approach presented in Chapter 4 also handles two-refraction cases, but differs from theirs in the following aspects: i) Their approach triangulates individual light paths separately to reconstruct the corresponding surface points, whereas I use an optimization procedure to solve all points in conjunction; ii) my approach reconstructs both refraction surfaces, whereas theirs only deals with a single surface; iii) I simultaneously recover both the 3D positions and normals of the refraction surface, whereas their approach computes the surface normals based on Snell's law in post-processing, which may not be consistent with the local shape; iv) I use only two cameras for data acquisition, while they use five, leading to more data to be captured; and v) precise object rotation and monitor translation are required in their setup and hence, applying their technique can be difficult. In contrast, my approach does not require precise positioning of the monitor, while the object is fixed during acquisition.

2.3 3D Reconstruction of Dynamic Fluid Surfaces

The following two subsections presents previous methods for recovering 3D dynamic fluid surfaces (*e.g.* water waves) when the underwater scene is a known flat pattern and when the underwater scene is a natural non-flat scene, respectively.

2.3.1 Known Flat Underwater Scene

The **single-view** based method was first introduced by Murase [47] in computer vision, where surface normals are recovered by capturing video with an orthographic camera of a flat background through wavy water. To eliminate the ambiguity in pixel-point correspondences, earlier efforts focus on proposing additional constraints, *e.g.* statistical appearance assumption of a fluid sequence [47], known average fluid height [34]. Recently, Shan *et al.* [62] improve Murase's method by solving all surface points at the same time under orthographic projection. However, their implementation requires a long exposure time (about 0.5 seconds) for each frame and thus is applicable to static objects only. By modeling the surface as a cubic B-spline, Liu *et al.* [42] introduce a parametric solution for reconstructing both mirror objects and transparent surfaces using pixel-point correspondences.

Ray-ray correspondence based methods are developed to avoid the ambiguity of pixel-point correspondences under a single-view setup. By placing a color screen at the focal length of a big lens, Zhang and Cox [77] associate each 2D source point of the background with a ray direction under orthographic projections. The incident rays are then easily obtained after getting pixel-point correspondences. Ye *et al.* [74] establish a similar setup by using a perspective camera. Wetzstein *et al.* [70] acquire ray-ray correspondences with light field probes [69]. Specifically, they replace the big lens with a lenslet array. A color pattern is then placed under the array, which encodes positional and angular correspondences using different color channels. All the above ray-ray correspondence based methods rely on special optics, which introduces many practical issues, *e.g.* calibrating the ray directions of background points [35] and making the setup waterproof [74]. In addition, as reported in their papers [70], [74], the surface positions obtained by intersecting the incident and exit rays are less accurate than that of the normals obtained by Snell's law. Furthermore, a surface integration algorithm is required to obtain the 3D shape from the normal information.

Another group of methods utilize **multiple viewpoints** to tackle the problem. Morris and Kutulakos [45] first propose using a stereo camera system to capture a dynamic fluid surface. By placing a checkerboard underneath the fluid surface, their approach can estimate both depths and normals based on pixel-point correspondences. Following their stereo setup, my approach presented in Chapter 5 not only inherits the advantage of easy implementation (*e.g.* no special devices required and can work under perspective projection) but also provides the following novel improvements: (1) In addition to cross-view normal consistency, my approach exploits a novel single-view normal consistency which takes local surface geometry into account; (2) Unlike their method which solves for each individual point independently, ours employs a global optimization scheme to recover all surface points simultaneously which results in higher accuracy in both depth and normal; (3) Since they compute depths and normals in separate steps, the surface obtained by mesh fitting based on the depth map and the one estimated via normal integration do not guarantee consistency. Typically, their normals are more accurate than the corresponding depths. Thus an additional surface integration from normals is required. In comparison, I simultaneously reconstruct depths and normals, which are both accurate and, most importantly, are consistent with each other; (4) I define a new error metric to recover the unknown refractive index without requiring to compute the complex inverses of pixel-point correspondences as in their method. It is noteworthy that the refraction stereo formulation has been extended to using a camera array [17], where the fluid surface is reconstructed by specular carving. However, the major limitations of [45] discussed above remain unsolved.

2.3.2 Unknown Non-Flat Underwater Scene

There are existing methods targeting at obtaining the 3D structure of underwater objects under a wavy surface. Alterman *et al.* [2] present a stochastic method for stereo triangulation through wavy water. However, their method can produce only a likelihood function of the object's 3D location. The dynamic water surface is also not estimated. More recently, Zhang *et al.* [76] treat such a task in monocular view and recover both the water surface and the underwater scene using a co-analysis of refractive distortion and defocus. Their method is limited in practical use. Firstly, to recover the shape of an underwater scene, an undistorted image captured through a flat water surface is required. However, such an image is very hard to obtain in real life, if not impossible. Secondly, the image plane of their camera has to be parallel with the flat water surface in their implementation, which is impractical to achieve. In contrast, my camera array-based setup presented in Chapter 6 can be positioned casually and is easy to implement. Thirdly, for the water surface, their method can return the normal information of each surface point only. The final shape is then obtained using surface integration, which is known to be prone to error in the absence of accurate boundary conditions. In comparison, my approach bypasses surface integration by jointly estimating the 3D positions and the normals of the water surface. Besides, the methods in [2] and [76] assume a still underwater scene, while both the water surface and the underwater scene can be dynamic in this thesis (Chapter 6). Hence, my proposed approach is applicable to a more general scenario.

Chapter 3

Frequency-Based Environment Matting

Extracting environment mattes using existing approaches often requires either thousands of captured images or a long processing time, or both. In this chapter, I present a novel approach to capturing and extracting the matte of a real scene effectively and efficiently. Grown out of the traditional frequency-based signal analysis, my approach can accurately locate contributing sources. By exploiting the recently developed compressive sensing theory, I simplify the data acquisition process of frequency-based environment matting. Incorporating phase information in a frequency signal into data acquisition further accelerates the matte extraction procedure. Compared with the state-of-the-art method, the proposed approach achieves superior performance on both synthetic and real data, while consuming only a fraction of the processing time.

3.1 Prerequisites

3.1.1 Problem Formulation

An environment matte describes how light is transferred from the environment through a transparent or reflective object to the camera. Figure 3.1 shows a typical data acquisition setup for environment matting, which consists of a camera and a monitor serving as light source. Following [19], [48], [81], the

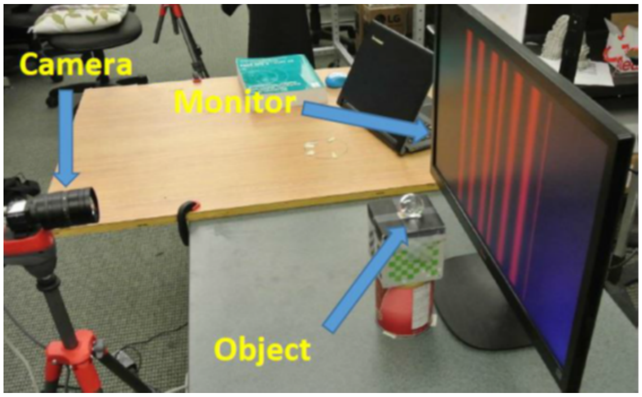


Figure 3.1: A physical setup for capturing environment mattes.

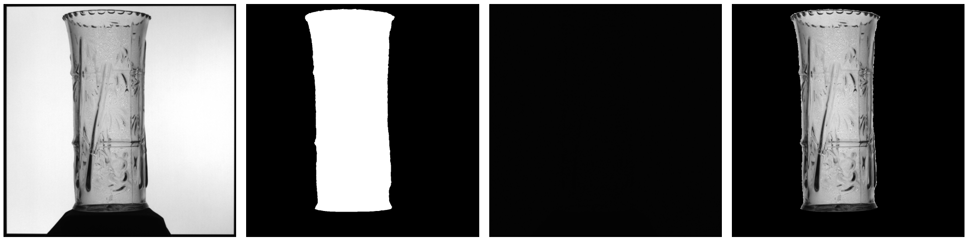
problem is usually modeled as

$$C = F + \rho \mathbf{W}\mathbf{B}, \tag{3.1}$$

where C is the intensity of a pixel in the composited image, and F the foreground object's color under the ambient illumination. $\mathbf{B}$ is an $n^2 \times 1$ vector representing the background image, and $\mathbf{W}$ the $1 \times n^2$ light transport vector describing the amount of contribution of light emitted from each background pixel to an object pixel, with the constraints $\|\mathbf{W}\|_1 = 1, \mathbf{W}_i \geq 0$. ρ is the light attenuation index which defines how light is attenuated by the object. In this way, each object pixel C is a combination of the foreground color F and the weighted contribution of the light emitted from the backdrop $\mathbf{B}$. Hence, the problem becomes: *given a number of captured images of an object against some known backdrops, how to extract the environment mattes: F, ρ and $\mathbf{W}$?*

Previous methods have shown that obtaining F and ρ under controlled environment is relatively easy [81], [82]. In particular, F can be obtained by displaying a pure black background because $C = F$ when there is no background contribution. ρ can be obtained by projecting a solid color background, where all entries in $\mathbf{B}$ have the same value b. Consider $\|\mathbf{W}\|_1 = 1$, Eq.(3.1) becomes $C = F + \rho b$, allowing ρ to be calculated after F is determined. Figure 3.2 gives some example outputs.

Thus the main task of environment matting is to recover the light transport vector $\mathbf{W}$ for each pixel. It is worth noting that I am only interested in the foreground object pixels, which are specified using a binary mask. The



(a) Object Image (b) Binary Mask (c) Foreground F (d) Attenuation Index ρ

Figure 3.2: (a) shows a transparent cylinder captured against a solid gray background. (b) shows the binary segmentation result. (c) shows the foreground color F of the object. The image looks black because $\mathbf{B} = 0$ (*i.e.* the monitor displays a black pattern and thus the illumination is turned off in the scene). (d) shows the estimated attenuation index ρ.

mask is obtained by capturing the scene with and without the object in front of 20 coarse-to-fine backdrops [82]. A pixel is considered an object pixel if the corresponding pixel colors of the object image and the reference image differ by more than a threshold in any of the 20 pairs. Two subsequent morphological operations, an opening followed by a closing operation with a 5×5 box structural element, are used to further refine the mask. A segmentation example is shown in Figure 3.2(b).

Frequency Analysis Model. In an effort to alleviate the ambiguity problem, Zhu and Yang [81] propose to estimate the matte in the frequency domain. The key idea is to utilize the following desirable properties of the *Discrete Fourier Transform (DFT)*:

1. Suppose a signal s_3 is a weighted combination of two other signals s_1 and s_2, *i.e.* $s_3 = w_1 s_1 + w_2 s_2$. Denote the frequency of s_1 and s_2 as f_1 and f_2, respectively, then s_3 is a signal with both f_1 and f_2;

2. Denote the complex vector S_3 as the DFT of s_3, I have $\text{mag}(S_3(f_1)) > 0$, $\text{mag}(S_3(f_2)) > 0$ and $\frac{\text{mag}(S_3(f_1))}{\text{mag}(S_3(f_2))} = \frac{w_1}{w_2}$, where $\text{mag}(\cdot)$ denotes the complex magnitude of a complex number;

As shown in Figure 3.3, by letting different pixels emit different frequency signals in the backdrop, I apply the DFT to the observed signal of each ob-

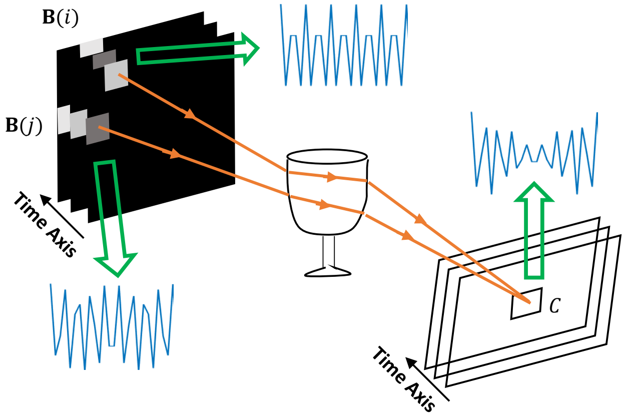


Figure 3.3: The object pixel C is contributed by two background pixels $\mathbf{B}(i)$ and $\mathbf{B}(j)$. By letting background pixels emit different frequency signals (*e.g.*, use pixel id as frequency value), the contributing sources can be obtained by analyzing the recorded signal in the frequency domain.

ject pixel and find the peaks of the frequency magnitude, which correspond to the contributing sources in the backdrop. The weights of these sources, *i.e.* the vector $\mathbf{W}$, can then be computed using the aforementioned property 2. However, for a backdrop with n^2 pixels ($n \approx 10^3$ for a conventional monitor), assigning each pixel a unique frequency requires at least $2 \times n^2$ images to be captured so that the frequency information can be recovered based on the Nyquist-Shannon Sampling Theorem. Capturing so many images is impractical and time-consuming.

To reduce the number of captured images, Zhu and Yang [81] split the data acquisition into two stages. Row-based patterns are first captured, where pixels in a row have the same frequency, then column-based patterns are captured, where pixels in a column share the same frequency. The final contributing sources can be jointly determined by row-based and column-based searching. While the number of images needed is reduced from $2 \times n^2$ to $4 \times n$, thousands of images are still needed to extract the matte at pixel level using a typical monitor.

3.2 Proposed Approach

In this section, I first present the sparsity of contributing sources under the frequency-based formulation. Then a CS-based reconstruction method is introduced to simplify the data acquisition process. Finally, I present a novel background design with phase incorporated to reduce the computational cost of L_1 minimization in CS.

3.2.1 Sparsity under Frequency-based Formulation

Unlike the previous work [19] that assumes the sparsity of the light transport vector $\mathbf{W}$ in Eq.(3.1), here I show the sparsity of contributing sources in my frequency-based pattern configuration. In particular, when row-based or column-based backdrops are displayed, an object pixel is only contributed by a few rows or columns. Hence, the corresponding DFT contains a small number of frequencies.

To quantitatively justify the sparsity of contributing frequencies in the recorded signal, I capture several objects under row-based and column-based frequency patterns. For row-based patterns, the intensity of each row in the temporal sequence is designed as

$$B(f,t) = \xi\Big(\cos\big(2\pi f \frac{t}{N}\big) + 1\Big), \tag{3.2}$$

where $1 \leq f \leq n$ is the row index of the background image, which also represents the frequency value at the fth row. ξ is set to 127.5 such that the range of pixel values is in $[0, 255]$. N represents the inverse of the sampling period, and I have $N \geq 2f_{max}$ according to the Nyquist-Shannon Sampling Theorem. In practice, I set $N = 2f_{max} + o$ and $o \in [10, 15]$ is an offset term. t is the time index (frame id) within the set $\{0, 1, \cdots, N-1\}$. The column-based pattern follows the same fashion.

After recording the object images under two kinds of backdrops, I apply the DFT to the received signals at each foreground pixel. For a single pixel under the row-based pattern, if the complex number at frequency f of the DFT is non-zero (the corresponding magnitude is non-zero), it means that the

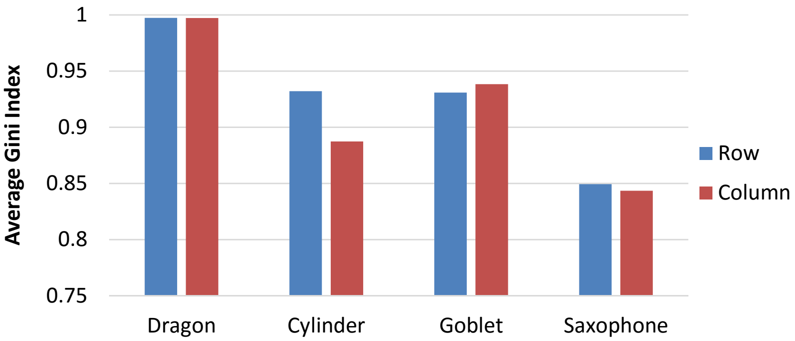


Figure 3.4: The Gini indices of the four objects used in this chapter. The resolution of the row-based and column-based pattern is set to 512 × 512 here for practical data acquisition.

received signal contains that frequency. Thus I conclude that the fth row in the background contributes to the pixel. Hence, the complex magnitudes of the DFT can be used to measure the sparsity of frequencies of each object pixel. I compute the Gini indices [31] (a widely-used sparsity metric in signal processing, with a higher value implying a sparser signal) of the magnitude vectors for all object pixels under both row-based and column-based patterns, and average them as the sparsity of rows and columns, respectively. As shown in Figure 3.4, the test objects have consistently high sparsity for both rows and columns. In the following subsections, I use the row-based pattern to illustrate my approach, since the column-based pattern is analyzed in the same fashion.

3.2.2 Reconstruction via Compressive Sensing

The existing frequency-based approach requires at least $4n$ (4096 when $n = 1024$) captured images for both row and column-based patterns. In contrast, the proposed CS-based approach utilizes the sparsity in frequencies to reduce the number of images required. To derive my CS-based method, I first consider the conventional DFT method for reconstructing frequency information. Given the recorded signal $\mathbf{C}$ and the computed ambient illumination F of a foreground pixel, I have $\mathbf{C} - F = \mathbf{DX}$, where $\mathbf{X}$ is an N-dimensional complex vector representing the frequency information of an object pixel and note that during searching non-zero frequencies, I am only interested in the sub-vector $\mathbf{X}(1 : f_{max})$. $\mathbf{D}$ is the inverse of the $N \times N$ discrete Fourier transform matrix.

Since each object pixel is contributed by only a few rows, *i.e.* frequencies, the CS theory can be used to reconstruct the sparse frequency information $\mathbf{X}$ by taking only $M < N$ measurements.

In practice, by randomly generating an M-dimensional permutation Ω of the set $\{0, 1, \cdots, N-1\}$ and displaying M backdrops pre-computed using Eq.(3.2) with frame ids from Ω, I solve the following L_1 minimization problem to reconstruct the frequency information $\mathbf{X}$:

$$\min \|\mathbf{X}\|_1, \text{ s.t. } \mathbf{C} - F = \mathbf{D}(\Omega, :)\mathbf{X}, \tag{3.3}$$

where $\mathbf{C}$ is an M-dimensional vector representing the recorded temporal signal of a foreground pixel, and $\mathbf{D}(\Omega, :)$ is the measurement matrix extracted from $\mathbf{D}$ by including only the rows with indices in Ω.

Besides sparsity, previous works [13], [19], [71] have shown the background regions that contribute to a foreground pixel can be clustered into several main groups. Since the signal frequencies correlate with pixel locations, such a group prior also exists in my frequency-based formulation. That is, most non-zero elements in $\mathbf{X}$ are neighbors and can be clustered into several local groups, which can help to improve the accuracy of L_1 minimization in Eq.(3.3). In practice, I apply the DGS tool [29] that can automatically handle the group clustering prior during optimization.

After $\mathbf{X}$ is obtained, a simple thresholding operation is performed to locate the contributing rows, *i.e.* the frequencies with non-zero magnitude. The threshold is set as $\max(\text{mag}(\mathbf{X}))/2$ in my implementation. Together with the contributing columns located in the same manner, the locations of contributing sources are thereby determined. The weight of the source at row r and column c is calculated as

$$\mathbf{W}(\text{ind}(r, c)) = \bar{\mathbf{W}}_{row}(r)\bar{\mathbf{W}}_{col}(c), \tag{3.4}$$

where $\text{ind}(\cdot, \cdot)$ returns the 1D index of the pixel located at the rth row and cth column in the background. $\mathbf{W}_{row}$ and $\mathbf{W}_{col}$ are computed from the frequency information $\mathbf{X}$ of row-based and column-based acquisitions, respectively, and are normalized before plugging into Eq.(3.4). Figure 3.7(c) is an example using the proposed CS-based frequency reconstruction, where $M = 160$ and $N = 2085$ are used for both row-based and column-based acquisitions.

3.2.3 Augment with Phase Information

The CS-based frequency search lowers the data acquisition requirement, but at the cost of a more expensive reconstruction process. Since the details of composition results depend on the resolution of the background pattern n, n needs to be large enough. When $n = 1024$, then the maximal frequency $f_{max} = 1024$ in the row-based and column-based patterns. Hence, the unknown vector $\mathbf{X}$ has a dimension of $N \geq 2f_{max} = 2048$. Solving such a large constrained minimization problem for all foreground pixels is time-consuming, *e.g.* extracting the environment matte of the object in Figure 3.7(c) takes over 26 minutes.

Background Pattern Design

To accelerate the process of solving L_1 minimization, I develop a new method that incorporates additional phase information to reduce the complexity of minimizing the L_1 norm. The core idea is to use both frequency and phase to identify the contributing sources. That is, for row-based patterns, I split the image into k horizontal regions. While different rows within the same region all have different frequencies, the corresponding rows in different regions have the same frequency but different phase values. This is achieved by assigning pixels in the fth row of the pth region the intensity of

$$B(f, t, \varphi_p) = \xi\Big(\cos\big(2\pi f \frac{t}{N} + \varphi_p\big) + 1\Big), \tag{3.5}$$

where $1 \leq f \leq \frac{n}{k}$ is the row index in the pth region, which also represents the corresponding frequency value. φ_p is a pre-designed phase value for the pth $(1 \leq p \leq k)$ region. The other notations are the same with the ones in Eq.(3.2). How to properly assign φ_p is discussed in Section 3.2.3. Figure 3.5 shows several example images of a goblet captured against my phase-augmented frequency-based patterns.

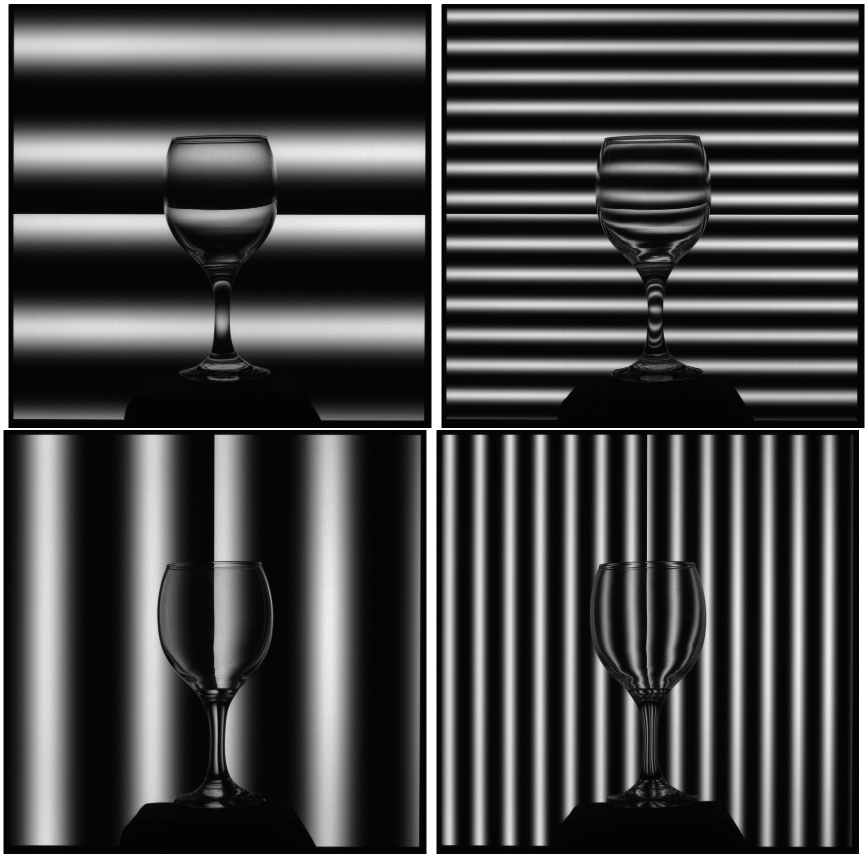


Figure 3.5: A transparent goblet captured against four example patterns generated using Eq.(3.5) by setting $N = 1060, k = 2, \varphi_1 = 220°, \varphi_2 = 320°$. The top row shows two row-based patterns, whereas the bottom row shows two column-based patterns. The two columns show the captured images at the time instance $t = 5$ and $t = 15$, respectively.

Adding to the background with k phases reduces the maximum frequency requirement from n to $\frac{n}{k}$, which subsequently reduces the dimension of $\mathbf{X}$ by k times and the computational cost of the L_1 minimization in Eq.(3.3). On the other hand, to determine the contributing sources, I need both frequency search and phase search. In practice, given the recorded temporal signal at a foreground pixel, I first determine the frequencies of the contributing sources by optimizing Eq.(3.3). Then, for a contributing frequency f, I compute its phase value to locate the region from which the frequency originates. Combining the phase and the frequency information gives us the row index in the background image.

Phase Acquisition and Inference

According to the theory of the DFT, given a set of phase candidates, the complex number $\mathbf{X}(f)$ is a weighted combination of different phase data:

$$\mathbf{X}(f) = \sum_{p=1}^{k} \alpha_p \left(\cos \varphi_p + j \sin \varphi_p \right) = R + jI, \tag{3.6}$$

where R and I are, respectively, the known real and imaginary part of $\mathbf{X}(f)$. If the frequency f comes from the pth region, I should have $\alpha_p > 0$ and vice versa. Therefore, if I know the coefficients α's, the contributing sources can be easily located. Considering the real and imaginary parts of Eq.(3.6) separately, I have two equalities. Hence, when $k = 2$, the two coefficients, α_1 and α_2, can be directly solved. When $k > 2$, additional equalities are required to compute the k coefficients.

To address the problem, I capture more frequency-based patterns under different phase settings. It is noteworthy that, regardless of the setting of the phase candidates $\{\varphi_1, \cdots, \varphi_k\}$, the complex number $\mathbf{X}(f)$ is non-zero as long as the fth row in some regions makes contribution to the object pixel. Furthermore, the coefficients α's are independent of the phase setting since they represent the amount of light from different regions. Therefore, to obtain k equalities for solving the k coefficients, I have to capture row-based patterns generated from Eq.(3.5) using $\frac{k}{2}$ different phase settings. It is worth noting that, due to the needs for additional phase setting, once $k > 2$, increasing k no longer reduces the number of background images needed. Nevertheless, the benefit of reducing the dimension of $\mathbf{X}$ remains.

Denote each phase setting as $\{\varphi_p^q : 1 \leq p \leq k\}$, where p is the region index and $1 \leq q \leq \frac{k}{2}$ the phase setting index. For each phase setting, by capturing the corresponding row-based patterns, I solve the optimization problem Eq.(3.3) to recover the frequencies. Then for each frequency, I have $\frac{k}{2}$ complex numbers: $\mathbf{X}^1(f) = R^1 + jI^1, \cdots, \mathbf{X}^{\frac{k}{2}}(f) = R^{\frac{k}{2}} + jI^{\frac{k}{2}}$, and they satisfy Eq.(3.6). Considering the real and imaginary parts separately, I have:

$$\begin{bmatrix} \cos\varphi_1^1 & \cos\varphi_2^1 & \cdots & \cos\varphi_k^1 \\ \vdots & \vdots & \ddots & \vdots \\ \cos\varphi_1^{\frac{k}{2}} & \cos\varphi_2^{\frac{k}{2}} & \cdots & \cos\varphi_k^{\frac{k}{2}} \\ \sin\varphi_1^1 & \sin\varphi_2^1 & \cdots & \sin\varphi_k^1 \\ \vdots & \vdots & \ddots & \vdots \\ \sin\varphi_1^{\frac{k}{2}} & \sin\varphi_2^{\frac{k}{2}} & \cdots & \sin\varphi_k^{\frac{k}{2}} \end{bmatrix} \times \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} R^1 \\ \vdots \\ R^{\frac{k}{2}} \\ I^1 \\ \vdots \\ I^{\frac{k}{2}} \end{bmatrix}. \tag{3.7}$$

For each frequency f, the corresponding coefficients $\{\alpha_1(f), \cdots, \alpha_k(f)\}$ can be obtained by solving the above linear system. I say that the fth row in the pth region (*i.e.* the row indexed at $r = f + \frac{n}{k}(p-1)$ in the background) makes contribution to the foreground pixel iff $\alpha_p(f) > 0$. The weight of the rth row is $\bar{\mathbf{W}}_{row}(r) = \frac{\alpha_p(f)}{\sum_{p,f}\alpha_p(f)}$.

In practice, because of measurement noise, the rth row is considered as a contributing row only if $\bar{\mathbf{W}}_{row}(r) > T$, where $T = \max(\bar{\mathbf{W}}_{row}(r))/2$ is used in all my implementation. By splitting the column-based pattern along the column direction and following the same phase settings, the column weights can be obtained in the same fashion, then the light transport vector $\mathbf{W}$ is computed using Eq.(3.4).

Construction of Phase Settings

Denote the linear system Eq.(3.7) as $\mathbf{Q}\alpha = \epsilon$, which could be indeterminate when $\mathbf{Q}$ is singular because of inappropriate phase settings. In addition, if similar degrees are used for neighboring phases, phase inference using Eq.(3.7) could locate undesired regions because of measurement noise.

To construct valid phase settings, three rules need to be followed: 1) The constructed $\mathbf{Q}$ is non-singular; 2) To make each region distinguishable by phase, there are no duplicated degrees in each phase setting; 3) The difference in phase values of adjacent regions should be large enough ($|\cos\varphi_p - \cos\varphi_{p+1}| > 0.5$ is used in my implementation). Note that the phase values are in the range $[0, 360)$. In my implementation, I randomly generate phase settings from $\{0, 20, \cdots, 340\}$ until the three rules are satisfied.

Considering $\alpha \geq 0$, solving $\mathbf{Q}\alpha = \epsilon$ is a classical *Non-negative Least Squares (NLS)* problem [39]. Here I propose to apply L_1 regularization to solve the linear system, which is more robust to noise than NLS. In particular, I compute the coefficients α's by solving

$$\min \|\alpha\|_1, \text{ s.t. } \mathbf{Q}\alpha = \epsilon, \alpha \geq 0. \tag{3.8}$$

3.3 Experiments

The proposed approach is tested using both synthetic and real transparent objects. The resolution of the background pattern is set to $n = 1024$ in all tests. Note that my CS-based data acquisition uses non-adaptive background patterns, which are generated and stored in advance. To prevent the interference caused by the bleeding effect of the monitor or other unknown light sources [81], the frequency range used is shifted up by 10Hz, *i.e.* $11 \leq f \leq 10 + \frac{n}{k}$. L_1 minimization is solved using the DGS tool [29] with group priors for Eq.(3.3) and without group priors for Eq.(3.8). Since the environment matte extraction process at each foreground pixel is independent, my parallel algorithm is implemented in MATLAB R2014b and accelerated on an 4-core PC with 3.4GHz Intel Core i7 CPU and 24GB RAM.

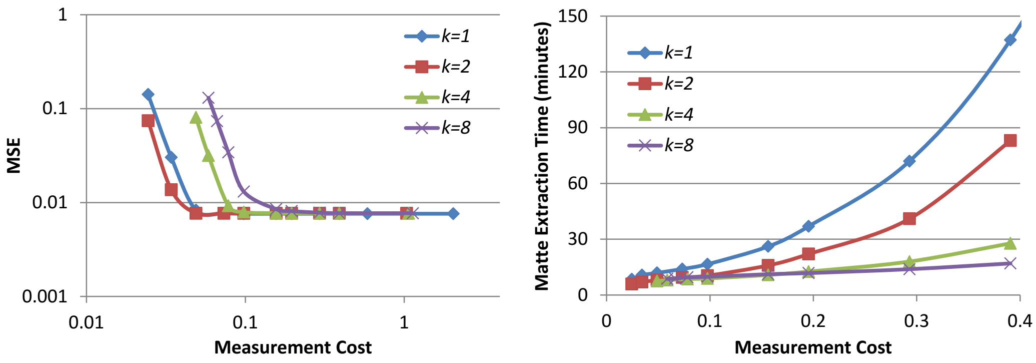
3.3.1 Synthetic Object

I start with quantitatively evaluate my approach using a complex synthetic model, the Stanford dragon, with my frequency-based backdrops texture mapped to the background. The data acquisition process is simulated using POV-Ray [50] and hence is free from measurement noise or lens imperfection. An image of the dragon in front of a checkerboard background is also rendered, which serves as the ground truth for measuring the *mean square errors* (MSE) of composited results.

Effectiveness of CS-based Phase-augmented Acquisition. I first evaluate the efficiency of the proposed data acquisition approach, which is usually quantified using the *measurement cost*, *i.e.* the ratio between the number of measurements and the number of unknowns. Here for each foreground pixel, I need to compute $\bar{\mathbf{W}}_{row}$ and $\bar{\mathbf{W}}_{col}$, which have a total of $2n$ unknowns. Denoting the total number of images used for both row and column based patterns as m, the measurement cost is defined as $\sigma = m/2n$.

As discussed in Section 3.1, the conventional frequency-based environment matting approach [81] requires $4n$ images. Hence, it has a measurement cost of 2. In my approach, with the additional phase search step, I only need to capture $2n$ images to reconstruct the frequency information using the DFT (*e.g.* in row-based acquisition, I have $k/2$ phase settings and for each phase setting I need to capture $2n/k$ images, thus n images are required). The measurement cost is therefore reduced to 1. Using CS-based acquisition can further lower the measurement cost dramatically without noticeably affecting the accuracy of the composited results. Figure 3.6(a) shows that the MSE remains to be low (< 0.01) when σ is set to about 0.1.

Figure 3.6(a) also shows the impact of the phase number k. As expected, without using phase ($k = 1$), a higher measurement cost is needed to achieve the same MSE than setting $k = 2$. Further increasing k results in more images needed to achieve a similar MSE. This is because using more phase regions corresponds to a fewer number of rows (columns) within the region. Since a foreground pixel is contributed by a fixed number of rows (columns), which often come from the same phase region, the frequency information becomes less sparse, which requires more samples for reconstruction according to the theory of CS.



(a) Accuracy

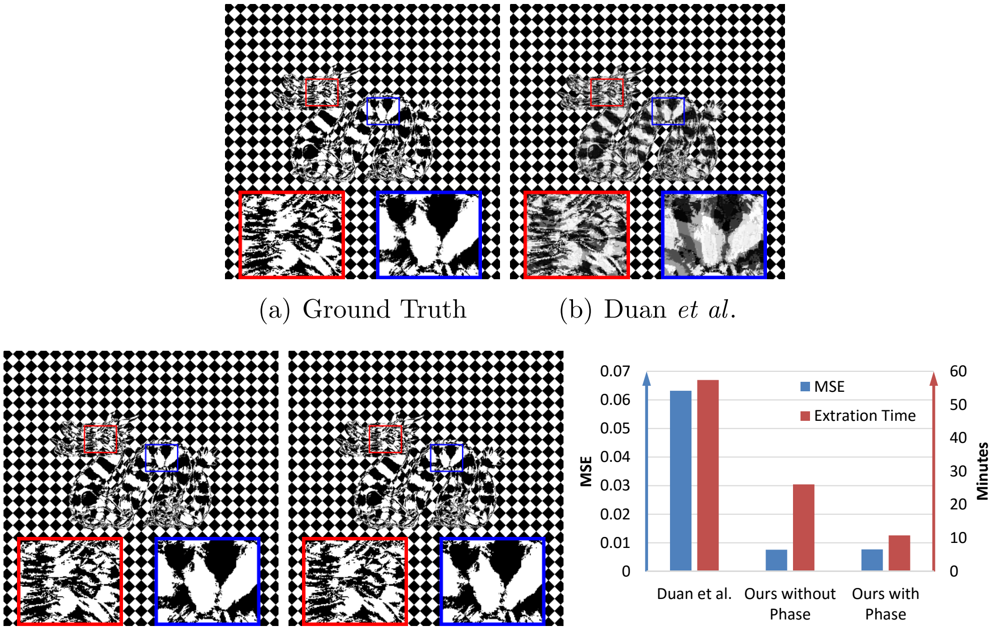
(b) Matte Extraction Time

Figure 3.6: The impact of phase region number k and measurement costs on both accuracy of the composited result (a) and matte extraction time (b). Note that log scale is used for both axes in (a).

Figure 3.6(b) further illustrates the impact of phases on the time of environment matte extraction. By splitting the pattern into more regions, the number of unknown frequencies decreases in Eq.(3.3), which accelerates the process of L_1 minimization, and thus speedups the whole process.

In summary, augmenting phase information can reduce the number of required images and accelerate the process of L_1 minimization. The phase region number k offers a tradeoff between the process of data acquisition and matte extraction. If the goal is to minimize the measurement cost while maintaining the accuracy of the composited results, $k = 2$ is the optimal setting. If the computational resource is limited, then a large k value should be used, which helps to reduce the cost of CS-based reconstruction.

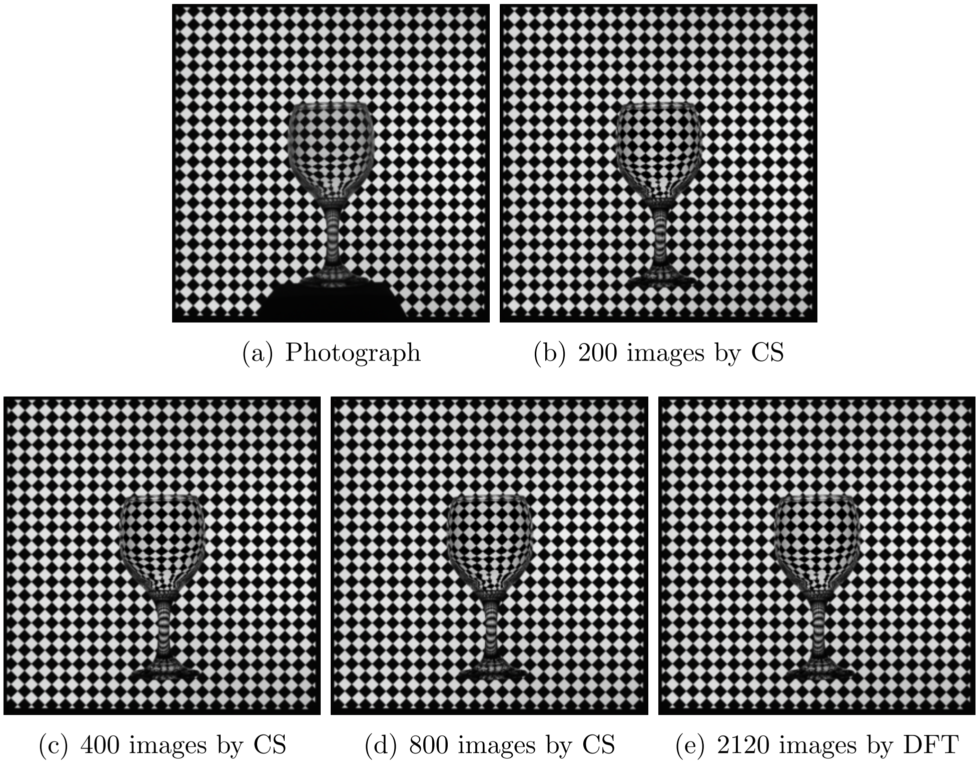
Comparisons with the CS in Spatial Domain. Finally, I compare my composited results with the latest CS-based environment matting method [19], which solves the problem in the spatial domain. The method of [19] is implemented by us based on their paper. To accelerate matte extraction, their method splits background pattern into square blocks and assumes a foreground pixel is contributed by these blocks. Thus their results appear blurry and blocky. As shown in Figure 3.7, my approach achieves superior performance in terms of both MSE and matte extraction time.



(a) Ground Truth (b) Duan *et al.*

(c) Ours without Phase (d) Ours with Phase (e) Comparison

Figure 3.7: Comparison using synthetic data with ground truth (a). The result of [19] (b) is computed by capturing 40 images in the coarse level and 300 images in the fine level. It is blocky because of their square block assumption. My CS-based approach (c & d) uses 320 images and shows better performance in terms of both accuracy and matte extraction time. The red and blue boxes show zoom in views.


(a) Photograph (b) 200 images by CS

(c) 400 images by CS (d) 800 images by CS (e) 2120 images by DFT

Figure 3.8: CS-based data acquisition on the Goblet object (under phase region $k = 2$). As the number of images increases, the result of CS-based approach improves. With 400 images, the result is visually comparable to the conventional DFT, which requires 2120 images.

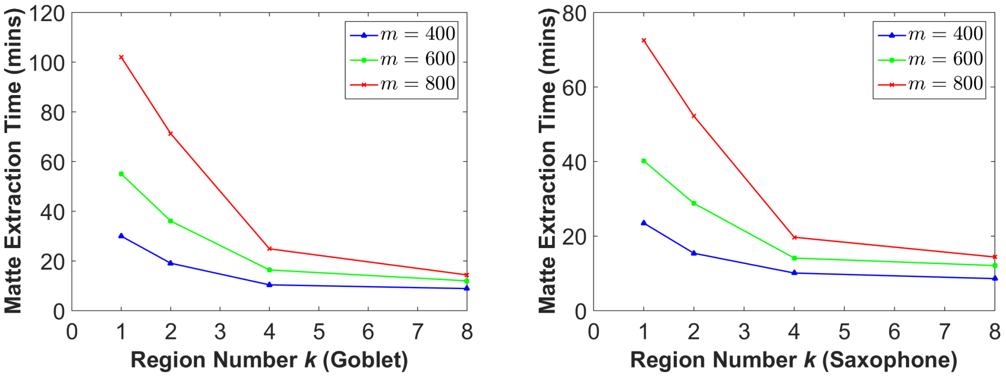


Figure 3.9: Effectiveness of using phase for accelerating matte extraction for two real objects: Goblet and Saxophone. Both results show that setting $k = 4$ can lower the computation cost by several manifolds, especially when a large number of captured images (m) is used. The benefit of further increasing k to 8 is limited.

3.3.2 Real Transparent Objects

Five objects, Goblet, Saxophone, Pie Pan, Trophy, and Cylinder, are used for testing the proposed approach on real captured data. Here I use an LG IPS monitor to display backdrops and a Point Grey Blackfly monochromatic camera to capture the scene. To automate the capture process, the patterns are displayed at 2fps, while the scene is captured in video mode at 6fps. As a result, three images are captured for each pattern and the middle one is used. This removes the needs for synchronizing between the monitor and the camera.

I first evaluate the effectiveness of CS-based data acquisition by comparing with the conventional DFT method. As shown in Figure 3.8, my approach achieves comparable results, while requiring only a fraction of sample images. The impact of the region number k is evaluated next. Figure 3.9 shows that, given the same number of captured images, the matte extraction process is accelerated as the region number k increases. Moreover, as illustrated in Table 3.1, setting a smaller k value ($e.g.$ $k = 2$) requires fewer images while maintains similar visual performance. Hence, the tests on real data further confirms that k offers a tradeoff between the data acquisition process and matte extraction.

Table 3.1 compares my approach with the spatial domain method in [19]. It shows that the proposed approach produces more realistic composited results while consuming only a fraction of processing time regardless of the setting of the region number. In the third scene, a goblet is laid on a glossy pie pan. The former object is highly refractive, whereas the latter reflects lights from a fairly broad area of the background. As a result, the contrast and sharpness of the two zoomed-in areas are noticeably different. My approach properly handles both areas, whereas Duan *et al.* gives blurry output due to the block assumption.

Table 3.2 further highlights the features of different environmental matting approaches. Although the state-of-the-art methods [13], [48] can handle multiple-region mapping and produce high quality visual effects, they require thousands of images. In addition, the time-consuming non-linear optimization in [13] depends on a number of parameters that can greatly affect the quality of mattes, while the adaptive data acquisition process in [48] takes hours and requires synchronization between the monitor and the camera. These limit their practical applications. For approaches with low data acquisition requirement, they require block assumption [19], [82] and thus cannot obtain visually pleasing results.

My approach locates contributing sources of the background at the pixel level and enjoys the following features: 1) Fast data acquisition and matte extraction process; 2) No camera/monitor synchronization or calibration needed; 3) Easy reproducibility with only two parameters, both of which are fixed in my experiments. These make my approach easy to use and can greatly facilitate follow-up applications, *e.g.* 3D reconstruction [38].

Note that I choose to use monochromatic camera in my experiment because the artifacts of Bayer mosaic can be eliminated. This helps to extract wavelength-dependent mattes, resulting proper handling of dispersion effects. As shown in Figure 3.10, by displaying patterns of different prime colors and performing environment matte extraction separately, I can render the dispersion effect of the object.

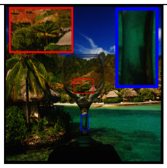
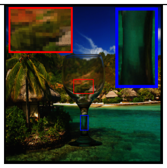
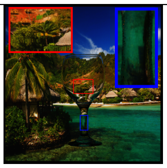
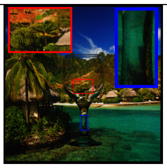
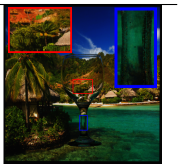
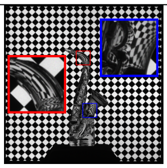
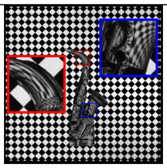
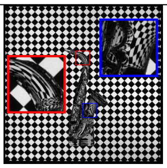
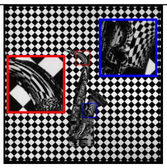
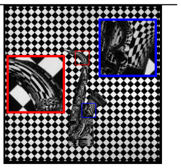
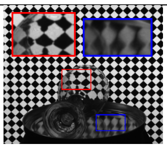
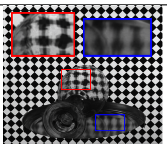
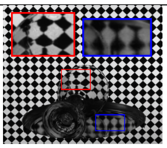
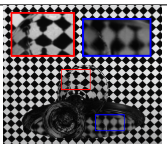
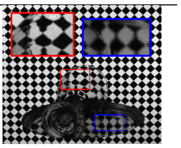
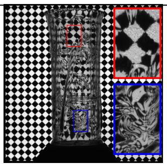
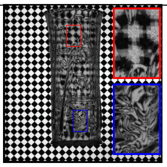
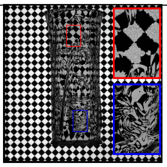
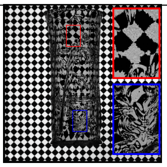
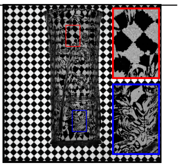
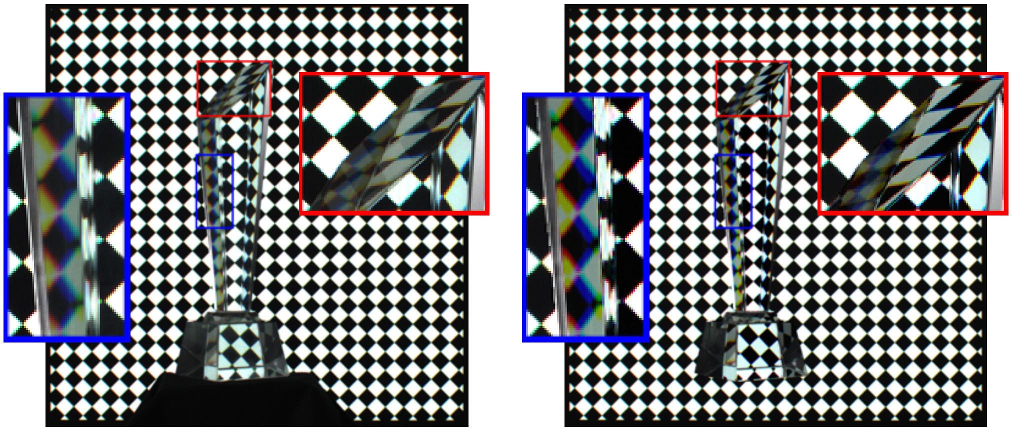
		Photograph	Duan *et al.*	$k = 2$	$k = 4$	$k = 8$
	# Img. (time)		340 (2.8)	400 (3.3)	600 (5)	800 (6.7)
Goblet	Composite Runtime		 128.6	 19.1	 16.4	 14.4
Saxophone	Composite Runtime		 145.5	 15.4	 14.1	 14.4
Pie Pan	Composite Runtime		 212.7	 24.0	 21.5	 20.4
Cylinder	Composite # imgs Runtime		 340 235.4	 400 55.5	 600 50.4	 800 48.7

Table 3.1: Comparison with the method in [19] on real data. Note that capturing more images will not improve the results of [19] due to the hierarchical sampling scheme being used. My results are more visually appealing, while consuming less processing time. Since the scenes are captured at 2fps in the video mode, the number of minutes needed for acquisition are computed as $\#imgs/120$.

Methods	# images when $n = 1024$	Runtime ($n = 1024$)	Remarks
Zongker *et al*. [82]	$\mathcal{O}(\log n)$, 20 images	20 mins when $n = 512$	Single-region mapping, block assumption
Chuang *et al*. [13]	$\mathcal{O}(n)$, 1800 images	Not available	Multi-region mapping, complex optimization
Real Time *et al*. [13]	1 image	2 mins	One-pixel mapping, colorless & pure specular object
Wavelet [48]	$\mathcal{O}(n)$, 2400 images	12 hours	Multi-region mapping, adaptive acquisition
Frequency [81]	$\mathcal{O}(n)$, 4096 images	5-10 mins	Multi-pixel mapping, slow acquisition
Duan *et al*. [19], [20]	$\mathcal{O}(s\log(n^2/s))$, 340 images	See Table 3.1	Multi-region mapping, block assumption
Ours	$\mathcal{O}(s\log(2n/s))$, 400 images	See Table 3.1	Multi-pixel mapping, fast acquisition & extraction

Table 3.2: Comparisons among different environment matting methods, where s denotes the sparsity of a signal. The information about the previous methods is directly copied or estimated from the corresponding papers. Note that the extraction time of [48] includes data acquisition because of its adaptive scheme.


(a) Photograph

(b) Composite

Figure 3.10: Handling dispersion. By processing the different color channels separately, my approach is able to render the rainbow phenomenon of the trophy. The greenish fringe around the checkerboard is due to chromatic lens aberration, which is not corrected in the experiment.

Chapter 4

Reconstructing Static Transparent Objects

Estimating the shape of transparent and refractive objects is one of the few open problems in 3D reconstruction. Different from opaque objects, light refraction is an important and unique property for transparent objects (*e.g.* glass, crystal). Based on Snell's law, the path of light refraction is determined by the normals of a transparent surface and thus conveys the cue for shape recovery of transparent objects. This observation motivates the work of this chapter.

Under the assumption that the light rays refract only twice when traveling through the object, I present the a novel approach to simultaneously reconstructing the 3D positions and normals of the object's surface at **both** refraction locations. Grown out of the environment matting methods in the last chapter, I first present a novel physical setup to capture the light paths. The acquisition setup requires only two cameras and one monitor, which serves as the light source. After acquiring the ray-ray correspondences between each camera and the monitor, I solve an optimization function that enforces a new position-normal consistency constraint. That is, the 3D positions of surface points shall agree with the normals required to refract the rays under Snell's law. Experimental results using both synthetic and real data demonstrate the robustness and accuracy of the proposed approach.

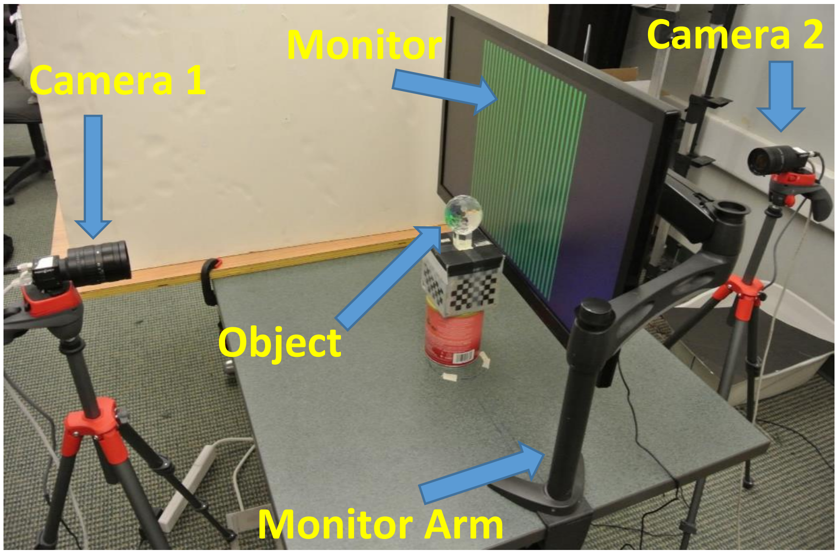


Figure 4.1: My data acquisition setup, where two cameras roughly face each other. Camera 1 is capturing data in this photo. Once Camera 1 is done, the monitor is moved to the other side of the object to serve as the light source for Camera 2.

4.1 Acquisition Setup and Procedure

My approach requires the acquisition of ray-ray correspondences before and after refraction. That is, for each observed ray refracted by the transparent object, I like to know the corresponding incident ray. As shown in Fig. 4.1, I use an LED monitor as the light source. Through displaying predesigned patterns on the monitor, the location of the emitting source for each captured ray can be found at pixel level accuracy. Adjusting the monitor location and repeating the process gives us two positions of the incident ray and hence, the ray direction can be determined. The same procedure is performed for the second camera, which observes the object in the opposite side of the first one.

Fig. 4.2 further illustrates the acquisition process in 2D. Two cameras are placed on the opposite sides of the object with their positions fixed during acquisition. For simplicity, I here refer the object surfaces on these two sides as the front and back surfaces, respectively. I first use Camera 1 to capture the front surface with the monitor positioned at plane m_1. The environment matting (EM) algorithm presented in Chapter 3 is applied to locate the contributing sources p_i on the monitor at pixel accuracy, which is achieved by projecting a set of frequency-based patterns. The monitor is then moved to plane m'_1 and the EM method is repeated. Connecting point p_i and p'_i gives us the incident ray direction $\overrightarrow{d_i^{in}}$ for the light source. The corresponding exit ray direction $\overrightarrow{d_i^{out}}$ is obtained from the intrinsic camera matrix, which is calibrated beforehand. I then capture the back surface using Camera 2 in a similar fashion with the monitor positioned at plane m_2 and m'_2.

Please note that precise monitor movement is not required in my setup. The monitor can be moved by any distance and its position can be easily calibrated by displaying a checkerboard patten [78]. It is noteworthy that instead of determining the incident ray using two monitor locations, light field probes [69] can also be used. However, I choose the monitor approach for two reasons: i) The light source locations can be determined at pixel-level accuracy and, ii) by displaying pattens with a primary color, my approach is robust to dispersion effects, whereas approaches relying on color-calibration are not.

So far, I have obtained the ray-ray correspondences w.r.t. the front and back surfaces using two cameras. In the subsections below, I present a novel reconstruction scheme that solves the following problem: *Given the dense ray-ray correspondences* $(p, \overrightarrow{d^{in}}) \Leftrightarrow (c, \overrightarrow{d^{out}})$ *of two cameras, how to compute the 3D positions and normals of the front and back surface points?*

4.2 Position-Normal Consistency

The seminal work [38] has shown that three or more views are required to reconstruct a single surface where the light path is redirected twice. Here I show that, by assuming the object surface is piecewise smooth, I can solve both the front and back surfaces using data captured from only two cameras. The key idea is that, for each reconstructed 3D surface point, its normal estimated based on its neighboring points should agree with the normal required for generating the observed light refraction effect.

I first explain how to measure position-normal consistency error for a given shape hypothesis. Here the object shape is represented using depth maps of

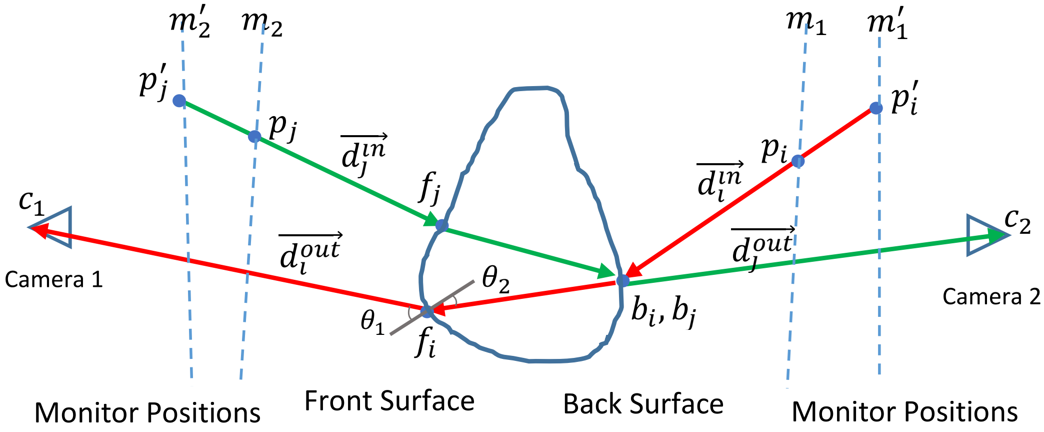


Figure 4.2: My acquisition setup using a pair of cameras and one monitor as light source. Note that the monitor is moved to different positions during acquisition.

the front and back surfaces, where the depth of a surface point is measured as its distance to the camera center along the camera's optical axis. Taking Camera 1 for example, as shown in Fig. 4.2, given a ray-ray correspondence $(p_i, \overrightarrow{d_i^{in}}) \Leftrightarrow (c_1, \overrightarrow{d_i^{out}})$ from Camera 1, the locations at which the incident ray $(p_i, \overrightarrow{d_i^{in}})$ and the exit ray $(c_1, \overrightarrow{d_i^{out}})$ meet the object are denoted as b_i and f_i, respectively. f_i can be computed from the corresponding depth map and how to compute b_i is discussed in Sec. 4.2.2. Connecting b_i and f_i gives us a hypothesis path that the ray travels through while inside the object. Hence, based on Snell's law, I can compute the normal at f_i, which is referred to as the Snell normal. Furthermore, using the 3D locations of nearby points of f_i, I can also estimate the normal of f_i using *Principal Component Analysis* (PCA) [57], which is referred to as the PCA normal. Ideally, the PCA normal and the Snell normal for the same point are the same. Hence, for the ith ray-ray correspondence, its position-normal consistency error is measured as:

$$E_{pnc}(i) = 1 - P(i) \cdot S(i), \tag{4.1}$$

where $P(i)$ (or $S(i)$) computes the PCA (or Snell) normal at the 3D location where the exit ray $(c, \overrightarrow{d_i^{out}})$ leaves the object. Note that the same definition also applied to ray-ray correspondences found by Camera 2.

In addition, based on the assumption that the object surface is piecewise smooth, I also want to minimize the depth variation in each depth map D. Hence, the second error term is defined as:

$$E_{so}(D) = \sum_{s \in D} \sum_{t \in \mathcal{N}(s)} (D(s) - D(t))^2, \tag{4.2}$$

where $\mathcal{N}(s)$ denotes the local neighborhood of pixel s in a given depth map D.

Combining both terms and summing over both front and back surfaces gives us the objective function:

$$\min_{D_f, D_b} \Big(\sum_{i \in \Omega} E_{pnc}(i) + \lambda \big(E_{so}(D_f) + E_{so}(D_b) \big) \Big), \tag{4.3}$$

where D_f and D_b are the depth maps for the front and the back surfaces, respectively, and Ω is the set containing all the ray-ray correspondences found by both cameras. Hence, Eq.(4.3) optimizes all the points in both depth maps at the same time using all available correspondence information. λ is a parameter balancing E_{pnc} and E_{so}.

Finally, here I provide an explanation to my new acquisition setup for transparent objects. As shown in Fig. 4.1, the two cameras used in our setup are facing each other: one observes the front surface, and the other the back surface. Compared to the conventional stereo setup for Lambertian opaque objects, the two cameras in my setup do not share a common field of view. There are mainly three reasons for such a specific design. Firstly, different from opaque objects, light interacts not only with the front surface but also with the back surface for a transparent object. Secondly, take Camera 1 for example, given the ray-ray correspondence $(p_i, \overrightarrow{d_i^{in}}) \Leftrightarrow (c_1, \overrightarrow{d_i^{out}})$, there are two unknowns to be solved — point f_i and point b_i — along the red light path shown in Fig. 4.2. However, there is only one normal consistency constraint if Camera 1 is used only. To make the problem solvable, one more constraint is required, which motivates using another camera to observe the back surface. Thirdly, to compute the *PCA* normal used in the normal consistency term Eq.(4.1) for both the front and back surfaces, the local neighborhood is needed to be searched for each surface point. By capturing the front and back surfaces using two cameras respectively, the pixel neighborhood relationship of the two cameras provides an easy access to the local neighborhood for each 3D surface point.

In the following subsections, I present how to compute the PCA and the Snell normals under the depth map hypotheses D_f and D_b. I use the front surface observed by Camera 1 to illustrate my approach, and the back surface is processed in the same fashion.

4.2.1 Normals from Positions by PCA

Given the positions of 3D points, previous work [57] has shown that the normal of each point can be estimated by performing a PCA operation, *i.e.*, analyzing the eigenvalues and eigenvectors of a covariance matrix assembled from neighboring points of the query point. Specifically, the covariance matrix $\mathcal{C}$ is constructed as follows:

$$\mathcal{C} = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} (f_j - f_i)(f_j - f_i)^T. \tag{4.4}$$

In my implementation, I use a 5 × 5 neighborhood of each pixel. The PCA normal is therefore the eigenvector of $\mathcal{C}$ with the minimal eigenvalue.

4.2.2 Normals by Snell's Law

As shown in Fig. 4.2, to obtain the Snell normal of f_i, the refractive index of the object, the interior ray path $\overrightarrow{b_i f_i}$ and the exit ray $\overrightarrow{f_i c_1} = \overrightarrow{d_i^{out}}$ are required. Here the refractive index is assumed to be known (how to handle objects with unknown refractive index will be discussed in Sec. 4.3). Since the front point position f_i is given, the interior ray direction $\overrightarrow{b_i f_i}$ can be obtained by locating the corresponding back point b_i. Note that b_i is observed by Camera 2 and on the line $(p_i, \overrightarrow{d_i^{in}})$. Hence, the problem of estimating the Snell normal of f_i is reduced to the problem of locating the first-order intersection between the back surface and the line $(p_i, \overrightarrow{d_i^{in}})$. A similar problem has been studied in image-based rendering, where the closest intersection between a ray and a disparity map needs to be computed. Here I apply the solution proposed in

[25], which converts the 3D intersection calculation problem into a problem of finding the zero crossing of a distance function.

After getting b_i, Snell's law is applied to estimate the Snell normal $S_1(i)$ for f_i. Denote η_1 and η_2 as the refractive index of air and the object, respectively, I have $\eta_1 \sin\theta_1 = \eta_2 \sin\theta_2$, where θ_1 and θ_2 are the angles between the normal and each of the light paths, as shown in Fig. 4.2. Let $\Delta\theta = \theta_1 - \theta_2 = \cos^{-1}(\overrightarrow{f_i c_1} \cdot \overrightarrow{b_i f_i})$, I have:

$$\theta_1 = \tan^{-1}\left(\frac{\eta \sin\Delta\theta}{\eta \cos\Delta\theta - 1}\right), \tag{4.5}$$

where $\eta = \eta_2/\eta_1$ is the relative index of refraction. The Snell normal is obtained by rotating $\overrightarrow{f_i c_1}$ by angle θ_1 on the plane spanned by $\overrightarrow{f_i c_1}$ and $\overrightarrow{b_i f_i}$, that is:

$$S_1(i) = \mathcal{R}(\theta_1, \overrightarrow{f_i c_1} \times \overrightarrow{b_i f_i})\overrightarrow{f_i c_1}, \tag{4.6}$$

where $\mathcal{R}(\theta, \vec{v})$ is the Rodrigues rotation matrix defined by θ and the rotation axis $\vec{v}$.

4.3 Optimize Depth Maps and Refractive Index

The aforementioned procedure returns the position-normal consistent model under a given refractive index hypothesis. For objects with unknown refractive indices, additional work needs to be done to estimate the proper index values. Similar to previous approaches [45], [62], my strategy here is to enumerate different refractive index values, evaluate the resulting models, and pick the best solution. However, unlike [45], [62], where the objective function to be optimized is directly used for evaluating the model quality, here a different reconstruction error metric is used.

As shown in Fig. 4.2, a given point b_i on the object surface may be involved in the ray-ray correspondence $(p_i, \overrightarrow{d_i^{in}}) \Leftrightarrow (c_1, \overrightarrow{d_i^{out}})$ from Camera 1 and the correspondence $(p_j, \overrightarrow{d_j^{in}}) \Leftrightarrow (c_2, \overrightarrow{d_j^{out}})$ from Camera 2. In the first ray path, b_i is the location where the incident ray enters the object, whereas in the second ray path, b_i is the location where the exit ray leaves. Two Snell normals can be computed from the two ray paths. Since the two Snell normals should be the same under the ground-truth model and the true refractive index value, their difference is a good measure of the reconstruction result. Hence, I define the reconstruction error for model D as:

$$RE(D) = \sum_{s \in \Psi} \left(1 - S_b(s) \cdot S_f(s)\right), \tag{4.7}$$

where $S_f(s)$ and $S_b(s)$ refer to the Snell normals computed using rays entering and exiting location s, respectively. Ψ is a set containing all locations on object surface that are involved in two ray-ray correspondences. It is worth noting that Eq.(4.3) only uses Snell normals computed using exit rays. Hence, Eq.(4.7) evaluate different errors as the objective function does.

Following [62], a coarse-to-fine optimization scheme is used for searching both the optimal refractive index and the optimal depth maps. In my implementation, I first downsample the obtained correspondences to 1/4 of the original resolution, enumerate the refractive index in the range of [1.2, 2.0] with increments of 0.05, and compute the optimal shape under each index value by minimizing Eq.(4.3). The relative index with the minimal reconstruction error as defined in Eq.(4.7) is then selected to compute the final model using the full resolution ray-ray correspondences.

Optimizing Eq.(4.3) is difficult because of the complex operations involved in the PCA and Snell normal computations. To avoid trivial local minima, I place a checkerboard in front of the front surface and the back surface, respectively. By calibrating the checkerboards, the depth searching ranges for D_f and D_b are obtained. Now Eq.(4.3) becomes a bounded constrained problem. I use the L-BFGS-B method [80] to solve Eq.(4.3) with numerical differentiation applied.

4.4 Experiments

The presented algorithm is tested on both synthetic and real data. The factor λ is fixed at 50 units in the synthetic data and 0.005 mm in the real experiments. Since the PCA and Snell normal calculations for different pixels can

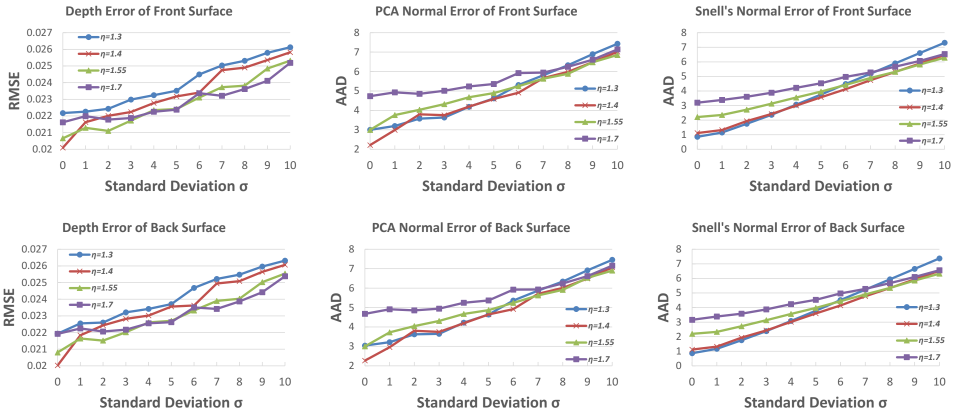


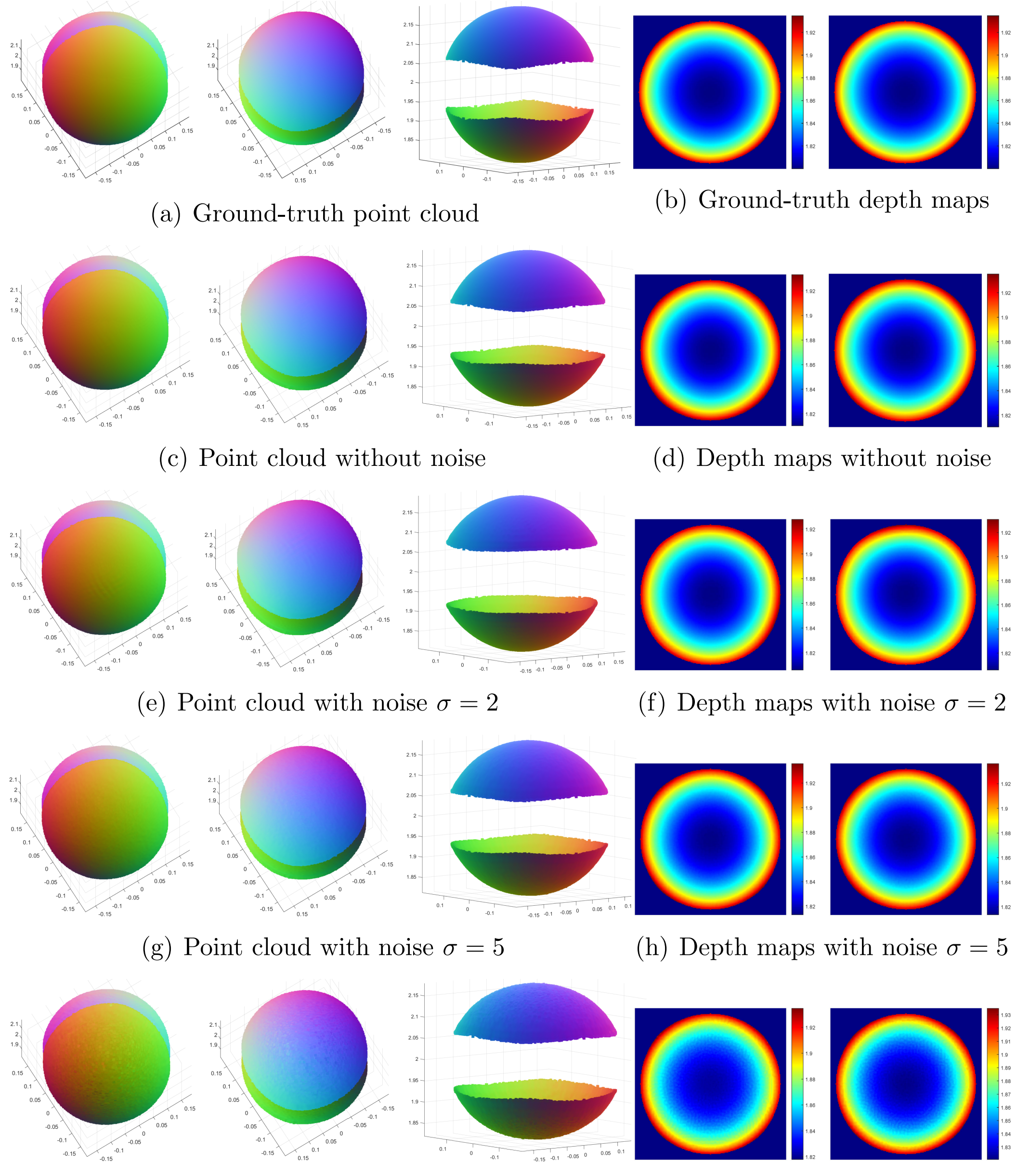
Figure 4.3: Reconstruction errors as a function of Gaussian noise level on the synthetic sphere under different refractive indices. 50 trials are performed under each setting.

be independently performed, they are computed in parallel. I implemented my parallel algorithm in MATLAB R2014b. Running on an 4-core PC with 3.4GHz Intel Core i7 CPU and 24GB RAM, the processing time needed for the models shown below varies between 1-2 hours.

4.4.1 Synthetic Object

I start with validating my approach on a synthetic sphere, where the ray-ray correspondences are generated by a ray-tracer. Specifically, the sphere is centered at $(0, 0, 2)$ with radius $= 0.2$. Two cameras are placed at $(0, 0, 0)$ and $(0, 0, 4)$. One observes the front surface and the other the back surface. By tracing along the poly-linear light paths, I can mathematically compute both the ground-truth positions and normals of the front and back surface points.

To evaluate the accuracy and robustness of my approach under different levels of data acquisition noise, I add zero-mean Gaussian noise to the obtained ray-ray correspondences. That is, for a given observed ray $(c, \overrightarrow{d^{out}})$, the corresponding light source locations under two monitor settings, p and p', are both corrupted with noise of standard deviation σ ($\sigma \leq 10$ pixels). The cameras are assumed to be calibrated, *i.e.*, their locations, orientations and internal parameters are not corrupted. I evaluate the reconstruction accuracy using three measures: the root mean square error (RMSE) between the ground-



(a) Ground-truth point cloud

(b) Ground-truth depth maps

(c) Point cloud without noise

(d) Depth maps without noise

(e) Point cloud with noise $\sigma = 2$

(f) Depth maps with noise $\sigma = 2$

(g) Point cloud with noise $\sigma = 5$

(h) Depth maps with noise $\sigma = 5$

(i) Point cloud with noise $\sigma = 8$

(j) Depth maps with noise $\sigma = 8$

Figure 4.4: Visual comparison between the ground truth and my results for the synthetic sphere under different noise levels. (a),(c),(e),(g) and (i) show the 3D point clouds colored with the PCA normals seen from three different viewpoints. Both the front and back points are plotted in the same coordinate system. The point clouds are colored with the corresponding PCA normal map. (b),(d),(f),(h) and (j) show the depth maps of the front and back surfaces.

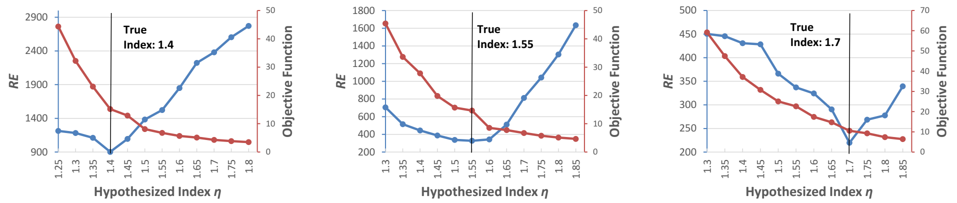


Figure 4.5: Refractive index estimation for the synthetic sphere. Blue curves plot the reconstruction error Eq.(4.7) as a function of hypothesized refractive index. Red curves plot the corresponding objective function Eq.(4.3). Ground-truth indices are shown with vertical lines.

truth depths and the estimated ones, the average angular difference (AAD) between the true normals and the reconstructed PCA normals, and the AAD between the true and the computed Snell normals. As shown in Fig. 4.3, my approach achieves high accuracy on both position and normal estimation and is robust to varying noise level. Fig. 4.4 visually compares the ground truth and the reconstructed results.

In addition to simultaneously reconstructing the 3D positions and normals, my approach can estimate the refractive index of the object. Here I evaluate the stability of refractive index estimation. By assigning different relative indices η, I capture the ray-ray correspondences using the ray-tracer. Then Gaussian noise with $\sigma = 5$ is added. Fig. 4.5 shows the variation of reconstruction error Eq.(4.7) with hypothesized refractive index. It shows that the index that corresponds to the minimum of the reconstruction error is close to the true index. This means that the proposed error term Eq.(4.7) can effectively estimate the refractive index. In comparison, directly using the objective function Eq.(4.3) cannot estimate the refractive index well.

4.4.2 Real Refractive Objects

Three transparent objects, a Swarovski ornament, a glass ball and a green bird, are used for evaluating the proposed approach on real captured data. The “ornament” and “ball” objects have apparent dispersion effects. To properly handle that, I use two Point Grey Blackfly monochromatic cameras so that artifacts of the Bayer mosaic can be avoided. An LG IPS monitor is used to display frequency-based patterns using a single color channel [54] with a resolution of 1024×1024. Calibration between the two cameras is challenging because they are facing each other (see Fig. 4.1). To address this difficulty, I place an additional camera between them and conduct pairwise camera calibrations. After calibration, the third camera is removed.

As shown in Table 4.1, the ornament and the glass ball each have many planar facets on its surfaces, resulting in complex light-object interaction and normal discontinuities. Fig. 4.6 shows my reconstruction results including the point clouds and depth maps, which are visually encouraging for both objects. More importantly, since my approach jointly optimizes the 3D positions and normals, the reconstructed normals are reconciled with the estimated point clouds.

Following [27], [38], to quantitatively assess the reconstruction accuracy, I manually label several facets shown in Table 4.1. For each facet, I fit a plane using the RANSAC algorithm [22]. Two measures are used to evaluate each facet: the AAD between the reconstructed normals and the fitted plane normal, as well as the mean distance error from the estimated 3D points to the plane. The quantitative measurements in Table 4.1 imply that the reconstructed normals and positions within each planar facet are consistent. This suggests that my approach can accurately reconstruct the piecewise planar structure without any prior knowledge of the shapes or parametric form assumptions.

Fig. 4.7 shows the reconstruction results of the bird. The model contains three largely separated parts. To avoid the inter-reflections between the three parts, I use tapes to block lights from the two smaller birds on the side when capturing the data. My results successfully captures the front and back shape of the bird in the center. Note that since the bird only transmit green light, approaches relying on light field probes won't work.

Fig. 4.8 shows the reconstruction error Eq.(4.7) under different hypothesized refractive indices. The estimated refractive index for "ornament" is 1.65, which agrees with the available report [64] stating that the refractive index of Swarovski crystal is between 1.5 and 1.7.

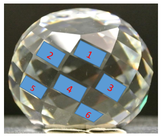
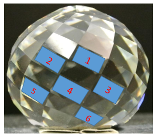
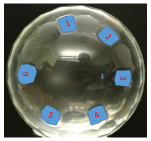
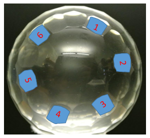
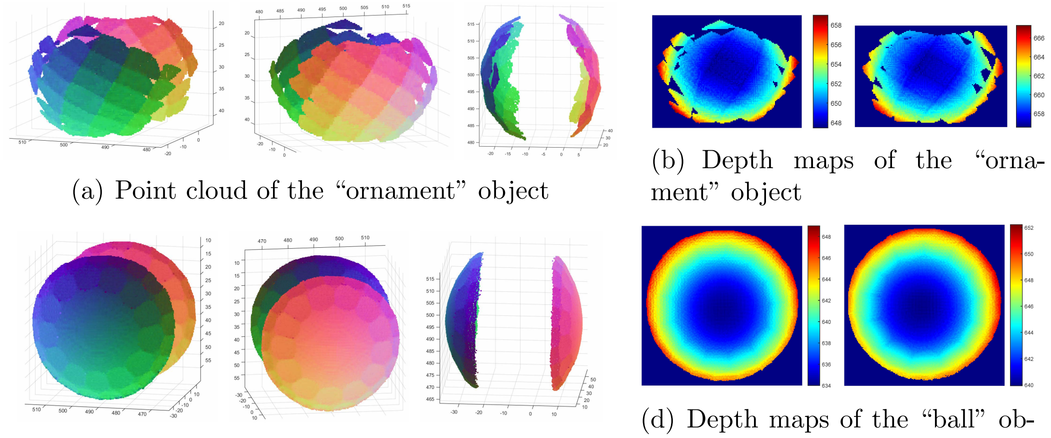
		Facet	1	2	3	4	5	6
(a)		Mean positional error (mm)	0.13	0.15	0.14	0.13	0.21	0.18
		AAD of PCA normals ($degree$)	3.28	5.10	4.77	3.32	6.50	5.10
		AAD of Snell normals ($degree$)	4.27	6.07	5.63	3.93	7.12	6.02
		RANSAC position inliers (%)	50.94	40.16	41.18	49.32	39.21	44.67
(b)		Mean positional error (mm)	0.13	0.11	0.13	0.12	0.15	0.15
		AAD of PCA normals ($degree$)	3.83	3.10	3.70	2.63	5.47	5.34
		AAD of Snell normals ($degree$)	4.52	4.08	4.72	3.42	6.21	6.09
		RANSAC position inliers (%)	45.13	52.88	45.29	57.64	43.95	39.66
(c)		Mean positional error (mm)	0.17	0.21	0.15	0.12	0.13	0.21
		AAD of PCA normals ($degree$)	6.26	7.08	5.85	3.02	3.86	7.69
		AAD of Snell normals ($degree$)	6.86	6.76	6.43	3.88	4.82	7.22
		RANSAC position inliers (%)	36.29	41.62	37.37	50.10	44.52	38.58
(d)		Mean positional error (mm)	0.18	0.18	0.12	0.07	0.06	0.07
		AAD of PCA normals ($degree$)	6.36	5.37	3.31	1.90	1.71	1.75
		AAD of Snell normals ($degree$)	7.01	6.09	4.30	2.99	2.85	2.60
		RANSAC position inliers (%)	37.41	40.77	48.60	69.10	77.23	70.12

Table 4.1: Reconstruction errors of the “ornament” and “ball” objects. Several planar facets are manually labeled as shown in the images above. Each facet is fitted using RANSAC with the inlier threshold of $0.1mm$. (a) and (b) show the results of the front and back surfaces of the ornament, respectively. (c) and (d) show the results of the front and back surfaces of the ball, respectively.



(a) Point cloud of the “ornament” object

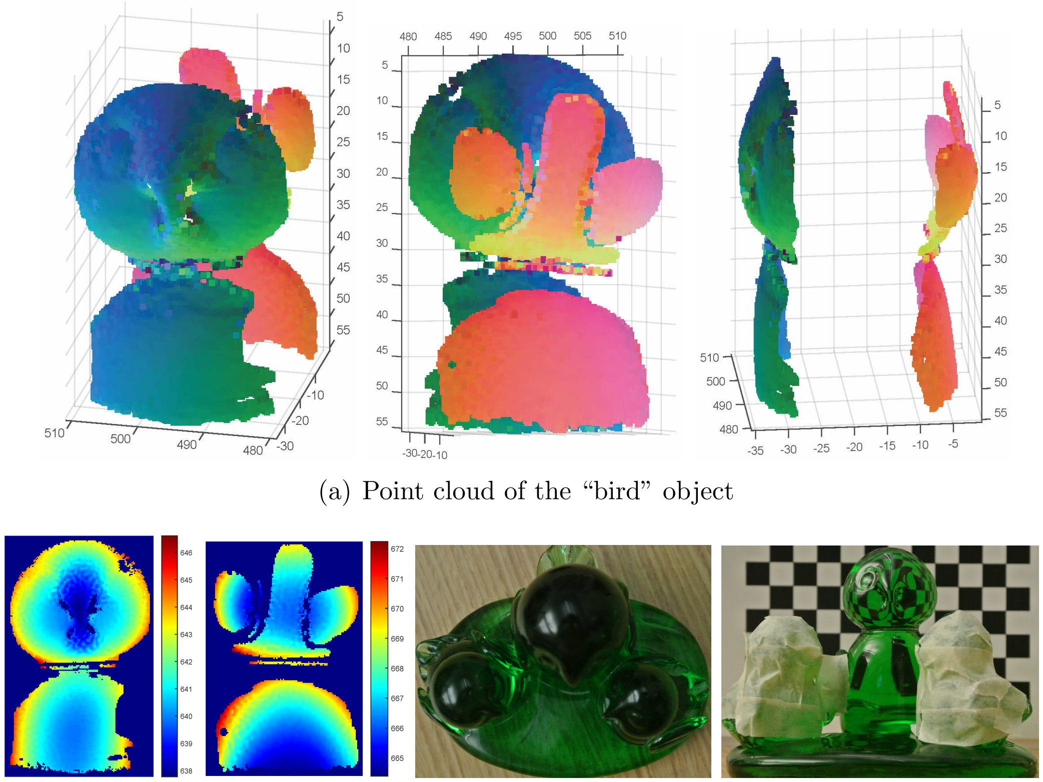
(b) Depth maps of the “ornament” object

(c) Point cloud of the “ball” object

(d) Depth maps of the “ball” object

Figure 4.6: Reconstruction results of the “ornament” (top) and “ball” (bottom) objects; please refer to Table 4.1 for the photos. (a) and (c) show the 3D point clouds colored with the PCA normals seen from three different viewpoints. Both the front and the back points are plotted in the same coordinate. (b) and (d) show the depth maps of the front and the back surfaces. Note that some holes exist on the surface because no ray-ray correspondences are obtained for those regions.

Table 4.2 shows the running time of the proposed approach on the three real objects.



(a) Point cloud of the "bird" object

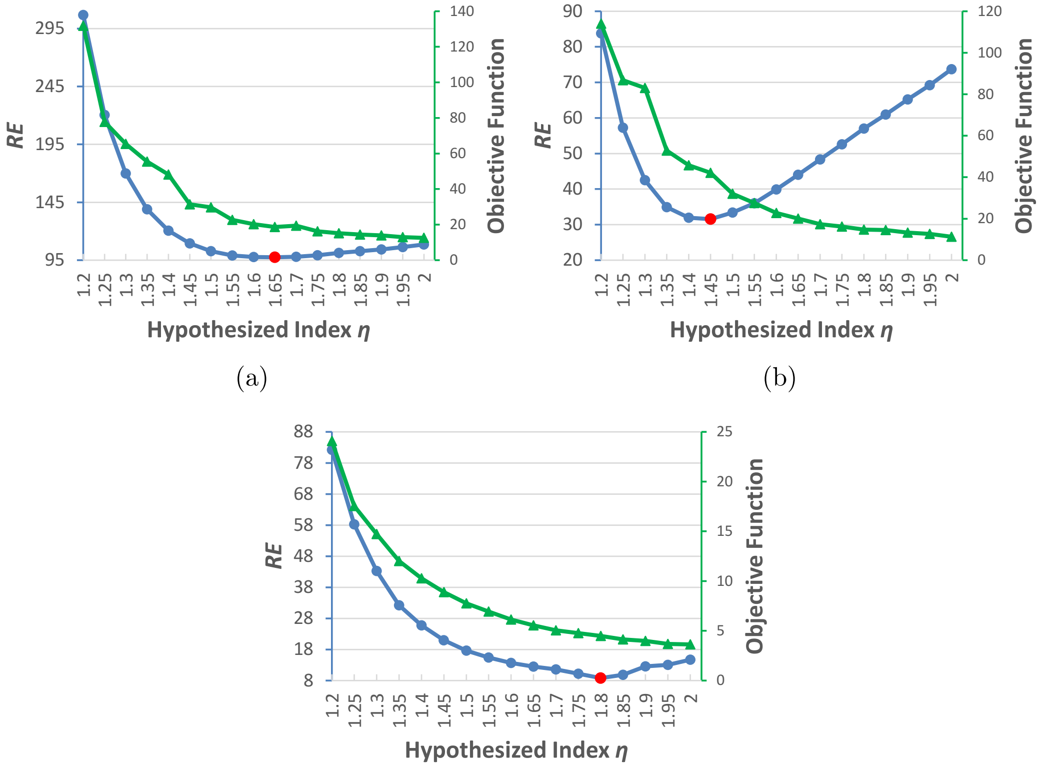
(b)

(c)

(d)

(e)

Figure 4.7: Reconstruction results of the "bird" object. (a) shows the point cloud colored with the PCA normals seen from three viewpoints. (b) and (c) show the depth map of the front and the back surfaces. (d) shows the top view of the object. Because the positions of the three birds overlap, multiple refractions may happen between the camera and the light source. To avoid that, I cover the two smaller birds with tapes as shown in (e) and only reconstruct the larger bird for illustration. Note that the obtained back surface of the head of the larger bird is incomplete because the correspondences are not available in those complex regions.



(a)

(b)

(c)

Figure 4.8: Refractive index estimation for the "ornament" (a), "ball" (b) and "bird" (c) object. Blue curves plot the proposed reconstruction error term as a function of hypothesized refractive index. Green curves plot the corresponding objective function. The minimum of the blue curve is marked in red.

Object	# of correspondences	Time (in minutes)		
		Refractive index estimation	Final reconstruction	Total time
Ornament	17268	55.8	13.1	68.9
Ball	64723	40.2	52.8	93.0
Bird	28340	54.6	25.8	80.4

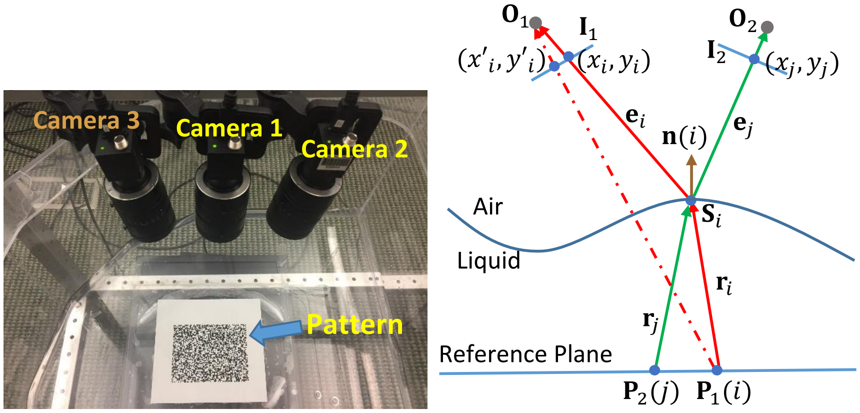
Table 4.2: Reconstruction time of the three real objects. As introduced in Section 4.3, a coarse-to-fine optimization procedure is used in my approach. At the coarse stage, I downsample the acquired correspondences to 1/4 of the original resolution to estimate the refractive index. At the fine stage, I reconstruct the final shape using the full resolution correspondences and the estimated refractive index.

Chapter 5

Reconstructing Dynamic Fluid Surfaces

Chapter 4 presents a refraction-based method for recovering the 3D shape of static transparent objects. In this chapter, I discuss the problem of reconstructing dynamic transparent surfaces (*i.e.* fluid surfaces), which is also an open and challenging problem in computer vision.

Unlike previous approaches that reconstruct each surface point independently and often return noisy depth maps, I propose a novel optimization-based approach that recovers both depths and normals of all 3D points simultaneously. Using the traditional refraction stereo setup, I capture the wavy appearance of a pre-generated random pattern, and then estimate the correspondences between the captured images and the known background by tracking the pattern. Assuming that the light is refracted only once through the fluid interface, I minimize an objective function that incorporates both the cross-view normal consistency constraint and the single-view normal consistency constraints. The key idea is that the normals required for light refraction based on Snell's law from one view should agree with not only the ones from the second view, but also the ones estimated from local 3D geometry. Moreover, an effective reconstruction error metric is designed for estimating the refractive index of the fluid. I report experimental results on both synthetic and real data demonstrating that the proposed approach is accurate and shows superiority over the conventional stereo-based method.



(a) (b)

Figure 5.1: Acquisition setup (a) and the corresponding refraction stereo geometry (b). Note that Camera 3 in the left figure is for accuracy assessment only and not used during 3D reconstruction.

5.1 Correspondence Acquisition and Matching

My approach computes the 3D shape of a transparent fluid surface based on how it refracts light. Specifically, for each pixel, the position of the corresponding background point is required, *i.e.* pixel-point correspondence. As shown in Fig. 5.1(a), I place a pre-generated pattern at the bottom of a tank, and capture the scene from two different viewpoints with Camera 1 and Camera 2, respectively. For each camera, I first capture the pattern without water as a reference image **B**. The cameras are synchronized for capturing dynamic surfaces after adding water. Note that an additional camera (Camera 3) is used for accuracy evaluation using an image-based rendering method in my real experiments, which is discussed in Sec. 5.4.2.

Fig. 5.1(b) illustrates the acquisition setup in 2D. Consider two perspective cameras centered at $\mathbf{O_1}$ and $\mathbf{O_2}$ observing a refractive surface against a flat background. Taking Camera 1 for example, for each pixel (x_i, y_i) in Camera 1, light originating from the corresponding point $\mathbf{P}_1(i)$ on the reference plane gets refracted at surface point $\mathbf{S}_i$. Let $\mathbf{I}_1(x_i, y_i, t)$ be the tth captured frame of the refraction distorted pattern. Here, the goal is to estimate the light source point $\mathbf{P}_1(i)$ for each pixel (x_i, y_i). I first apply the coarse-to-fine variational framework [7] to compute the optical flow (u_i, v_i) between $\mathbf{I}_1$ and $\mathbf{B_1}$. Then the forward projection (x_i', y_i') of point $\mathbf{P}_1(i)$ is easily computed as (x_i+u_i, y_i+v_i). Suppose the relative poses between the cameras and the reference plane are calibrated beforehand and fixed during acquisition, the 3D coordinates of $\mathbf{P}_1(i)$ is estimated by intersecting ray $\mathbf{O}_1\mathbf{P}_1(i)$ with the reference plane.

The choice of the displayed pattern is critical for accurate correspondence matching and subsequent 3D reconstruction. Traditional methods [17], [45] use a checkerboard pattern and track the feature corners. Correspondences of non-corner pixels are obtained by interpolation. However, these methods assume that the first frame of the liquid surface is nearly flat, which is usually impractical, so that a reliable initial correspondence field can be obtained. In contrast, inspired from the successful applications of random patterns in single-shot structured light [23], I choose a binary random pattern generated from Bernoulli distributions [68] as shown in Fig. 5.11. Different from a regular checkerboard, a Bernoulli random pattern contains fewer repetitive structures, which helps to reduce ambiguities while searching correspondences in a local window. Besides, the binary random pattern extends the advantage of a checkerboard in handling light absorption, dispersion, chromatic abberations, etc, compared to color-based ones [14], [70].

The correspondence matching for Camera 2 works analogously. The same procedure is applied to different frames. So far, I have obtained the pixel-point correspondences of a liquid motion sequence from two cameras. Next, I present a novel reconstruction framework that solves the following problem: *Given the pixel-point correspondence function* $\mathbf{P}_1()$ *and* $\mathbf{P}_2()$ *of each frame from two views, how to recover the depths and the normals of the dynamic surface, as well as the refractive index?*

5.2 Stereo-Based Reconstruction

My approach formulates an optimization framework which enforces two forms of normal consistency constraints. Specifically, for each 3D point, the normals estimated based on light refraction from two different viewpoints should be consistent. On the other hand, they are also required to agree with the normal estimated based on single-view local shape geometry.

5.2.1 Normal Definitions

Here I first explain the definitions of the different types of normals mentioned above. Similar to color-based stereo matching, I set Camera 1 as the primary camera and the fluid surface is represented by a depth[1] map $\mathbf{D}$ in the scope of Camera 1. As shown in Fig. 5.1(b), for the ith surface point $\mathbf{S}_i$ associated with pixel (x_i, y_i) of Camera 1, let d_i be its hypothesized depth. The 3D coordinates of $\mathbf{S}_i$ can then be computed by first assuming that the camera's parameters are known. Given the pixel-point correspondence $\mathbf{P}_1(i)$, I get the ray direction $\mathbf{r}_i$ by connecting $\mathbf{P}_1(i)$ and $\mathbf{S}_i$. Then, the normal of $\mathbf{S}_i$ can be computed based on Snell's law, given the incident and exiting rays $\mathbf{r}_i$ and $\mathbf{e}_i$, respectively. I refer to this normal as the *LeftSnell* normal, denoted by $\mathbf{n}_1(i)$. Snell's law states that the normal $\mathbf{n}_1(i)$, the incident ray $\mathbf{r}_i$ and the exiting ray $\mathbf{e}_i$ are co-planar, and thus $\mathbf{n}_1(i)$ can be represented as a linear combination of $\mathbf{r}_i$ and $\mathbf{e}_i$. That is, $\mathbf{n}_1(i) = (\eta_l \mathbf{r}_i - \eta_a \mathbf{e}_i)/\|\eta_l \mathbf{r}_i - \eta_a \mathbf{e}_i\|$, where η_l and η_a denote the refractive index of liquid and air, respectively. I set $\eta_a = 1$ in my experiments and here the medium's refractive index η_l is assumed to be known. How to deal with fluid surface with an unknown refractive index is discussed in Sec. 5.3.

On the other hand, by connecting $\mathbf{S}_i$ and $\mathbf{O}_2$, I get ray $\mathbf{e}_j$ and the forward projection (x_j, y_j). Similarly, since the correspondence source function $\mathbf{P}_2(j)$ is acquired beforehand, I can also compute another normal of $\mathbf{S}_i$ by Snell's law given light rays $\mathbf{r}_j$ and $\mathbf{e}_j$. I refer to this normal as the *RightSnell* normal, denoted by $\mathbf{n}_2(i)$. In a similar vein, $\mathbf{n}_2(i)$ is estimated by $\mathbf{n}_2(i) = (\eta_l \mathbf{r}_j - \eta_a \mathbf{e}_j)/\|\eta_l \mathbf{r}_j - \eta_a \mathbf{e}_j\|$.

In addition, the normal of a 3D point can be computed from its local shape geometry. That is, from the 3D locations of the neighboring points of $\mathbf{S}_i$, I can fit a tangent plane. Then the normal of $\mathbf{S}_i$ is approximated by the normal of the tangent plane. In particular, I estimate this normal by Principal Component

[1] In this thesis, depth is defined as the distance between a 3D point and the camera center along the z axis.

Analysis (PCA) [57], which is referred to as the *PCA* normal and denoted by $\mathbf{n}_p(i)$. The basic idea is to analyze the eigenvectors and eigenvalues of a covariance matrix constructed from nearby points of the query point. More specifically, the covariance matrix $\mathcal{M}$ at the point $\mathbf{S}_i$ is defined as:

$$\mathcal{M} = \frac{1}{|\mathcal{N}(i)|} \sum_{k \in \mathcal{N}(i)} (\mathbf{S}_k - \mathbf{S}_i)(\mathbf{S}_k - \mathbf{S}_i)^\top, \tag{5.1}$$

where $\mathcal{N}(i)$ denotes the local neighborhood of pixel i and $|\mathcal{N}(i)|$ the size of $\mathcal{N}(i)$. The *PCA* normal $\mathbf{n}_p(i)$ is thus the eigenvector of $\mathcal{M}$ with minimal eigenvalue.

5.2.2 Objective Function

To this end, I obtain three different normal estimations computed from different sources for each surface point $\mathbf{S}_i$. Ideally, the three estimates should be the same. Therefore, the difference between each pair of normals can be used to defined a normal consistency error. That is:

$$E_{12}(i) = 1 - \mathbf{n}_1(i) \cdot \mathbf{n}_2(i), \tag{5.2}$$

$$E_{1p}(i) = 1 - \mathbf{n}_1(i) \cdot \mathbf{n}_p(i), \tag{5.3}$$

$$E_{2p}(i) = 1 - \mathbf{n}_2(i) \cdot \mathbf{n}_p(i), \tag{5.4}$$

where E_{12} measures the cross-view normal consistency error, which is the one used in [45]. E_{1p} and E_{2p} are my new single-view normal consistency errors.

Furthermore, assuming that the fluid surface is piecewise smooth, I define the depth smoothness term at the ith point as:

$$E_{so}(i) = \sum_{k \in \mathcal{G}(i)} (d_i - d_k)^2, \tag{5.5}$$

where $\mathcal{G}(i)$ is the neighborhood pixel set containing the bottom and the right pixel of pixel i in my implementation.

Summing the above error terms and considering all the surface points, I obtain the following minimization problem:

$$\min_{d_i \in \mathbf{D}} \sum_{i \in \Omega_1} \left(\alpha E_{1p}(i) + \beta E_{2p}(i) + \gamma E_{12}(i) + \lambda E_{so}(i) \right), \tag{5.6}$$

where Ω_1 denotes the pixel set containing all the surface points in the region of interest. Hence, Eq.(5.6) couples both cross-view and single-view normal consistency constraints to optimize for the depths of all points simultaneously, whereas previous methods [17], [45] consider the cross-view error term E_{12} only and solve for each point independently. α, β, γ and λ are the parameters balancing different terms.

Note that Eq.(5.6) is defined w.r.t. a single frame. It is possible to solve the depth maps of all points from all frames by including them in Eq.(5.6) at the same time, which yield a large system that is computationally expensive. In contrast, I solve each frame independently and use the result of the last frame to initialize the current frame, which not only drastically reduces the running time and memory consumption but also maintains temporal coherence.

In addition, because of the complex operations involved computing the three normals, it is difficult to analytically derive the derivatives of Eq.(5.6). To tackle that, the previous method [45] employs the gold-section search [52] for pixelwise 1D optimization, which is computationally intensive when the number of unknowns is large and thus, the method is not applicable to my global objective function. Instead, in my implementation, I use the L-BFGS-B [80] method to optimize Eq.(5.6) using numerical differentiation.

5.3 Optimizing Depths and Refractive Index

As mentioned in Sec. 5.2.1, computing the *LeftSnell* and *RightSenll* normals both require the refractive index of the fluid. Given different refractive index hypotheses, solving Eq.(5.6) returns different depth maps. Hence, additional steps are required to get the desired 3D model when the index is unknown. Following previous methods [45], [62] and similar to the strategy used in Chapter 4, here I use a brute-force search approach. That is, I enumerate possible index hypotheses, evaluate the corresponding models based on a novel reconstruction error metric and pick the index with the minimal residual error.

The main idea of my proposed reconstruction error metric is based on the consistency of two optical flow fields estimated using different methods. On

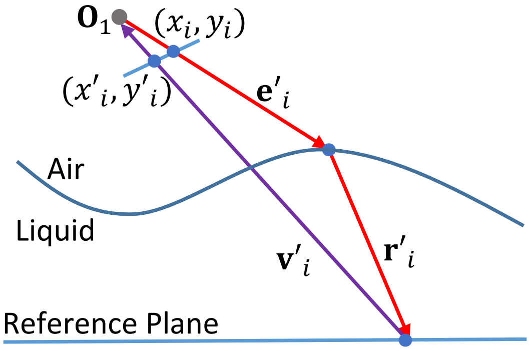


Figure 5.2: Ray tracing geometrically to estimate the shape-based optical flow field.

the one hand, as introduced in Sec. 5.1, for the ith pixel in Camera 1, I can compute the displacement vector (u_i, v_i) between the fluid image $\mathbf{I}_1$ and the reference image $\mathbf{B}_1$ using image-based cues [7]. On the other hand, since the 3D shape of the fluid surface is reconstructed, the flow field can also be obtained using shape-based cues. As shown in Fig. 5.2, for the ith pixel (x_i, y_i), I trace along each camera ray $\mathbf{e}'_i$ and locate its intersection with the fluid surface. The refracted ray $\mathbf{r}'_i$ is then obtained by Snell's law. Finally, the pixel coordinates (x'_i, y'_i) are obtained by projecting back to the camera center along the direction $\mathbf{v}'_i$, and the shape-based displacement vector is computed as $(u'_i, v'_i) = (x'_i - x_i, y'_i - y_i)$. Ideally, the image-based flow (IBF) vector (u_i, v_i) and the shape-based flow (SBF) vector (u'_i, v'_i) should be the same. A similar analysis can be applied to Camera 2. Hence, I design a novel error metric as follows:

$$EPE(k) = \sqrt{(u_k - u'_k)^2 + (v_k - v'_k)^2}, k \in \Omega_1 \cup \Omega_2, \tag{5.7}$$

which is based on the popular endpoint error (EPE) used in evaluating optical flow results [4]. Ω_c denotes the pixel set of the cth camera.

It is noteworthy that the proposed error metric Eq.(5.7) is different from the one used in [45]. Their error metric requires to compute the inverses of the correspondence functions $\mathbf{P}_1()$ and $\mathbf{P}_2()$, which unfortunately may not be generally invertible when multiple pixels receive contributions from the same point. In contrast, my metric does not have such a problem.

In practice, a coarse-to-fine optimization procedure is implemented to search for both the optimal depth map and the best refractive index. I first downsample the acquired correspondence functions $\mathbf{P}_1()$ and $\mathbf{P}_2()$ to 1/4 of the original resolution. Then, for each index hypothesis in a given range, I optimize Eq.(5.6) and evaluate the produced depths based on Eq.(5.7) under the coarse resolution. The index value that gives the smallest reconstruction error is selected. The final shape is reconstructed using the full correspondence functions and the optimal index.

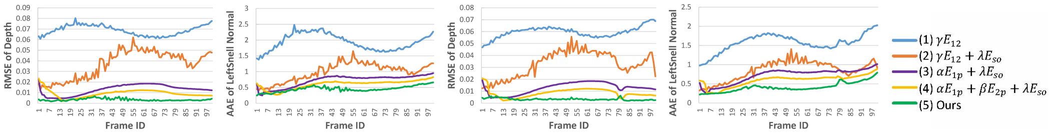
5.4 Experiments

The proposed approach is evaluated on both synthetic and captured data. The parameter settings $\alpha = \beta = 1, \gamma = 1000, \lambda = 100$ (*unit*) are used in synthetic data and $\alpha = \beta = 1, \gamma = 20, \lambda = 0.005$ (*mm*) are used in real experiments. During the coarse-to-fine minimization, the maximum iteration numbers of L-BFGS-B optimization are fixed to 200 and 20 for the downsampled and full resolutions, respectively, for the first frame. The iteration numbers are reduced by half for the remaining frames. I use the 5×5 and 3×3 local neighborhoods $\mathcal{N}()$ in Eq.(5.1) at the low and full resolution, respectively. Consider computing the normals $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_p$ for different points can be performed independently. I implement my algorithm employing parallelization in MATLAB R2016a on an 8-core PC with 3.2GHz Intel Core i7 CPU and 24GB RAM.

5.4.1 Synthetic Data

I first validate my approach on a synthetic sinusoidal wave: $z(x, y, t) = 2 + 0.1\cos(\pi(t + 50)\sqrt{(x-1)^2 + (y-0.5)^2}/80)$. In practice, the two cameras are placed at $(0, 0, 0)$ and $(0.05, 0, 0)$, respectively. The reference plane is at $z = 2.5$. By mapping a Bernoulli pattern on the reference plane, I start with rendering the reference image $\mathbf{B}$ without the fluid. Then the distorted image with the wavy surface is simulated using a ray-tracer as illustrated in Fig. 5.2. The correspondence functions are obtained by performing the correspondence matching algorithm in Sec. 5.1.

The proposed approach is evaluated using the following two measures: the



(a) Index $\eta_l = 1.33$ (b) Index $\eta_l = 1.33$ (c) Index $\eta_l = 1.55$ (d) Index $\eta_l = 1.55$ (e) Legend

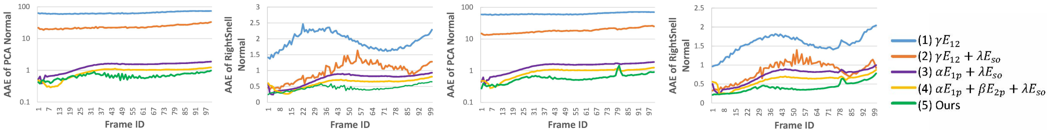
Figure 5.3: Different error measures as a function of frame id for synthetic wave. (a) and (b) shows the error plots when the refractive index $\eta_l = 1.33$ is used in wave simulation. (c) and (d) shows the error curves when $\eta_l = 1.55$ is used in wave simulation.

root mean square error (RMSE) between the ground-truth depths and the computed ones, and the average angular error (AAE) between the true normals and the recovered *LeftSnell* normals. Here the *LeftSnell* normals, which can be generated by both the existing method [45] and my approach, are used for fair comparisons.

To validate the effectiveness of the proposed constraints, I first evaluate the algorithm by removing different terms from Eq.(5.6). The objective function used in each case is listed in Fig. 5.3(e). Case 1 includes the cross-view term E_{12} only and corresponds to that used in the previous method [45]. Adding a spatial smoothness term (Case 2) can effectively reduces the errors and hence, the smoothness term is used for all other comparisons with [45]. Case 3 is equivalent to a single-view solution, where only the correspondence information from Camera 1 is used. Case 4 uses E_{1p} and E_{2p}, whereas my approach incorporates all three normal consistency constraints Eq.(5.2,5.3,5.4) in the objective function Eq.(5.6) and yields the smallest errors. Moreover, Fig. 5.3 also shows the robustness and temporal coherence of my approach over time.

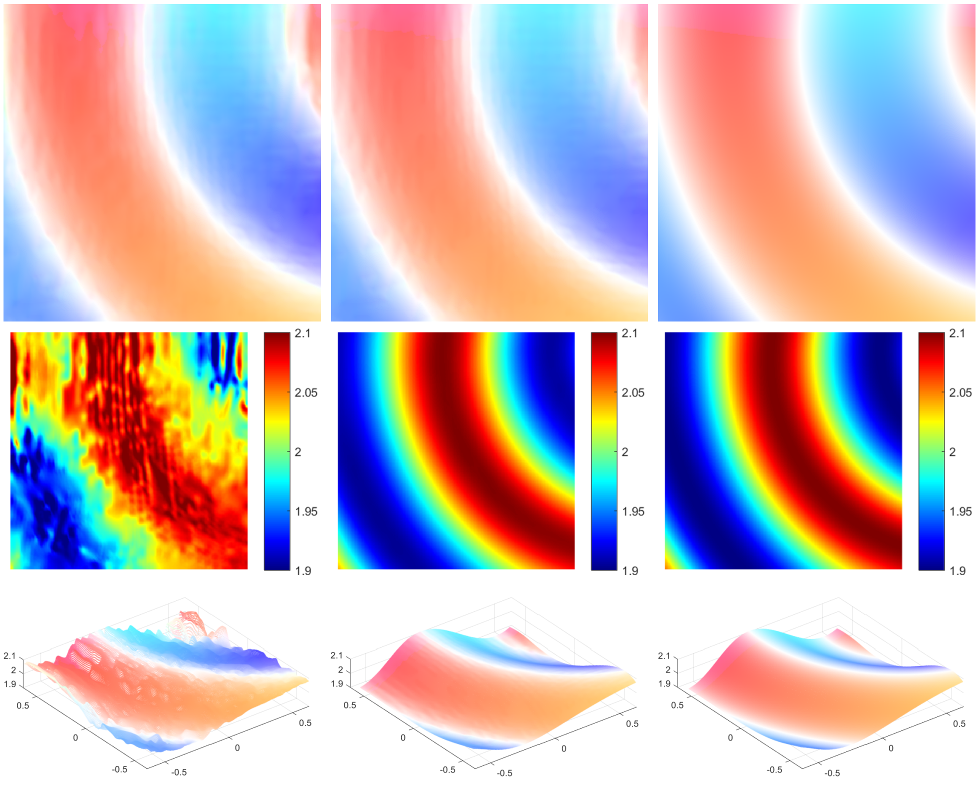
Besides, Fig. 5.4 shows the evaluation results based on the *PCA* and *RightSnell* normals, and the results are similar the ones of the *LeftSnell* normals shown in Fig. 5.3.

Fig. 5.5 compares the conventional stereo-based method [45] with ours. For fair comparisons, the pixel-point correspondences generated using my approach are used. The results show that, with added smoothness constraint,



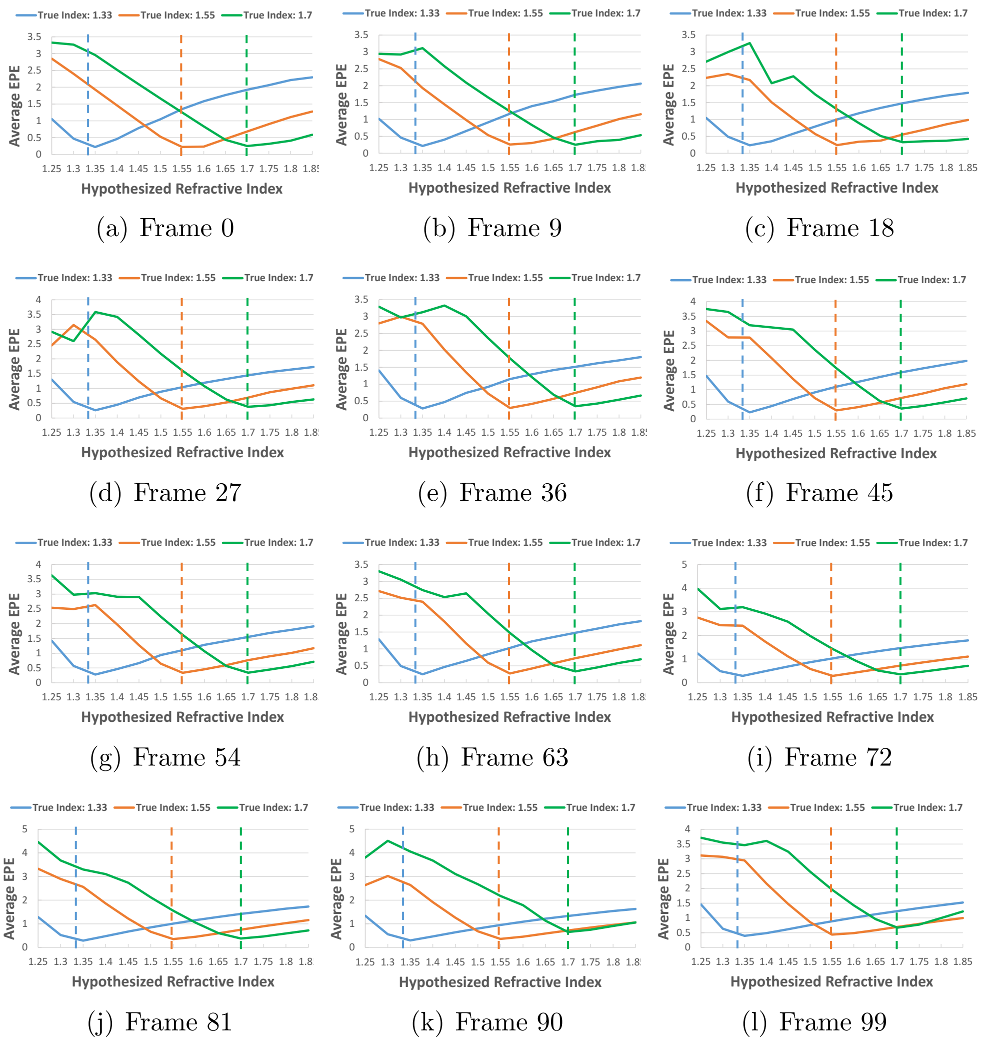
(a) Index $\eta_l = 1.33$ (b) Index $\eta_l = 1.33$ (c) Index $\eta_l = 1.55$ (d) Index $\eta_l = 1.55$ (e) Legend

Figure 5.4: Evaluations on the *PCA* and *RightSnell* normals for synthetic wave. (a) and (b) show the error plots when the refractive index $\eta_l = 1.33$ is used in wave simulation. (c) and (d) show the error curves when $\eta_l = 1.55$ is used in wave simulation. Note that, for better visualization, the 10-base log scale is used for the vertical axes in (a) and (c).



(a) [45]+Smoothness (b) Ours (c) Ground Truth

Figure 5.5: Visual comparisons between the method in [45] and ours for an example frame when $\eta_l = 1.33$ is used for simulation. From top to bottom, it shows the *LeftSnell* normal map, the depth map and the point cloud colored with *LeftSnell* normals. Please see the supplemental videos [63] for the full video sequence as well as the captured images.



(a) Frame 0 (b) Frame 9 (c) Frame 18

(d) Frame 27 (e) Frame 36 (f) Frame 45

(g) Frame 54 (h) Frame 63 (i) Frame 72

(j) Frame 81 (k) Frame 90 (l) Frame 99

Figure 5.6: Average EPE Eq.(5.7) as a function of refractive index hypotheses using several example frames of synthetic data. The vertical dashed lines indicate the true indices.

their estimated normal maps are similar to ours. However, their estimated depths are noisy whereas ours are smooth. More importantly, my approach simultaneously recovers the depths and the normals, which are both accurate and consistent with each other.

In addition to obtaining the 3D fluid surfaces, my approach can recover the refractive index of the fluid. Here I test the reliability of refractive index estimation. By setting different refractive indices in simulation, I render the distorted images with the fluid using my ray-tracer. As shown in Fig. 5.6, for each ground-truth index setting, I reconstruct the 3D shape and compute the

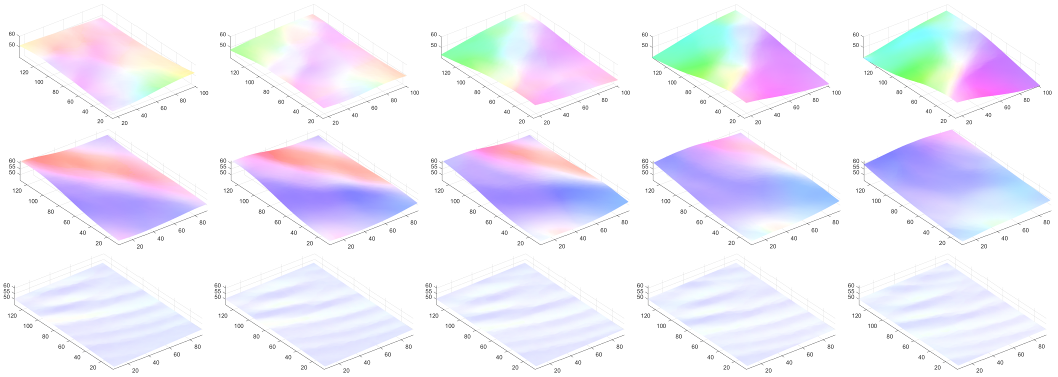


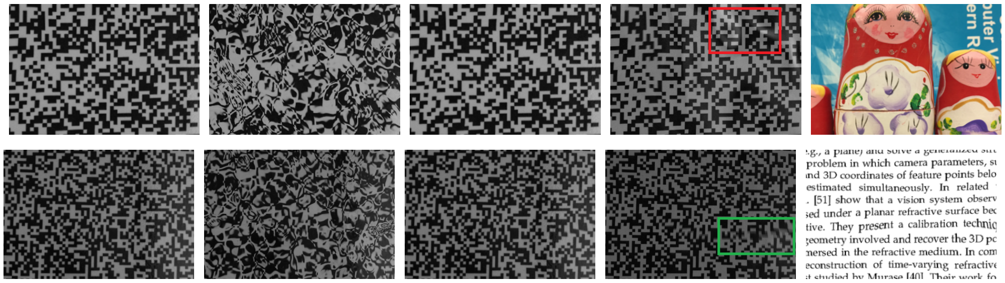
Figure 5.7: Point clouds of adjacent frames of Wave 1 (top), Wave 2 (middle), Wave 3 (bottom). It shows that my results are visually temporal coherent. In this chapter, a point cloud is plotted based on its corresponding depth map and colored with *LeftSnell* normals.

average EPE Eq.(5.7) under each index hypothesis in the range of $[1.25, 1.85]$ with increments of 0.05. The EPE curve exhibits a minimum that is close to the true refractive index, which demonstrates that the new error metric Eq.(5.7) can effectively estimate the refractive index.

5.4.2 Real Dynamic Water Surfaces

In order to capture real fluid surfaces, I set up a system as shown in Fig. 5.1(a). Three synchronized Point Grey Flea2 cameras are used for capturing video at 30fps at a resolution of 516×388. Cameras 1 and 2 are used for 3D reconstruction and refractive index estimation, whereas Camera 3 is used *for accuracy assessment only.* I print my binary random patterns on A4-sized papers using a commodity printer. The pattern is then laminated to be waterproof. The refraction effect caused by the thin laminated plastic layer is negligible. The pattern is attached to the bottom of the tank. Another feasible but more expensive solution is to use a waterproof tablet for displaying patterns. Before adding water, I calibrate the relative poses between the cameras and the pattern using a checkerboard [78].

In Fig. 5.7, three captured water waves are shown and the full sequences can be found in the supplemental videos [63]. Both Wave 1 and Wave 2 are generated by randomly perturbing the water surface at one end of the



(a) $IBF(\mathbf{B}_3)$ (b) $SBF(\mathbf{B}_3)$ [45] (c) My $SBF(\mathbf{B}_3)$ (d) Captured Image $\mathbf{I}_3$ (e) Composite

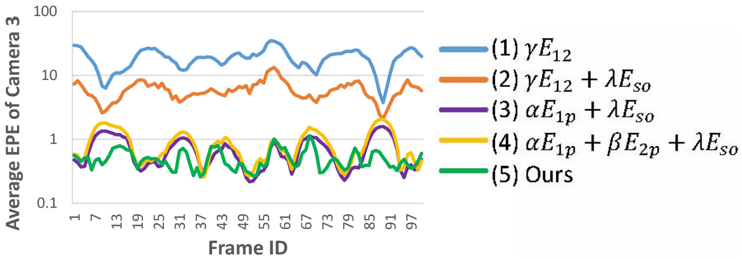
Figure 5.8: View synthesis using an example frame of Wave 1 (top) and Wave 2 (bottom). The shading effects caused by reflection/caustics (red box) and motion blur effects (green box) can be observed in captured images (d). In (e), I compose the reconstructed 3D surface onto new scenes using the ray-tracing method as depicted in Fig. 5.2.

tank and both exhibit large water fluctuations and fast evolutions. However, two different Bernoulli random patterns with different block sizes are used for evaluating the robustness of the proposed algorithm against pattern changes; see Fig. 5.8. Wave 3 is a small rippled case generated by dripping water drops near one side of the pattern. My approach can faithfully recover the propagating annular structures produced by the water drops.

Novel View Synthesis. To evaluate reconstruction quality, I first use the reconstructed surface shape to synthesize the view at Camera 3 and visually compare it with the image observed by the camera. In particular, I first compute the IBF field at Camera 3 using the observed image $\mathbf{I}_3$ and the reference image $\mathbf{B}_3$ as discussed in Sec. 5.1. I then compute the SBF field of Camera 3 from the reconstructed 3D surface using the ray-tracing method as discussed in Sec. 5.3 and shown in Fig. 5.2. I can now warp $\mathbf{B}_3$ using either the IBF or the SBF to obtain the synthesized image $IBF(\mathbf{B}_3)$ and $SBF(\mathbf{B}_3)$, respectively[2]. By comparing the captured image $\mathbf{I}_3$ with $IBF(\mathbf{B}_3)$ and $SBF(\mathbf{B}_3)$, I can qualitatively evaluate the accuracy of pixel-point correspondences and the quality of 3D reconstruction, respectively.

As shown in Fig. 5.8, my approach can faithfully synthesize the obser-

[2] Here I use the italic form $IBF()$ and $SBF()$ to denote the functions that compute the synthesized image using IBF and SBF, respectively.



Case	1	2	3	4	Ours
Mean	18.90	5.96	0.68	0.86	**0.50**
Stdev.	6.48	2.01	0.36	0.48	**0.17**

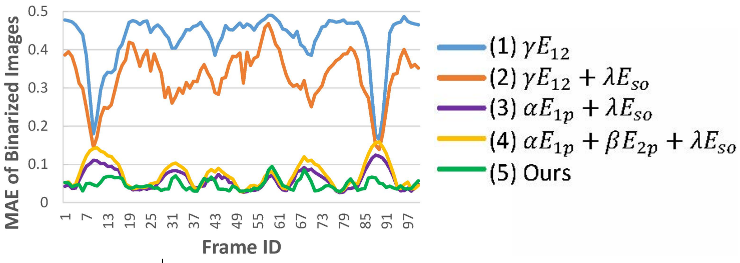
Figure 5.9: Quantitative evaluation on results generated under different constraints using Wave 1. The top figure plots average EPE as a function of frame ID. For better visualization, the 10-base log scale is used for the vertical axis. The bottom table shows the corresponding mean and standard deviation (stdev.) of EPE among all frames.

vations at Camera 3, whereas the results of [45] look quite different. The comparison also shows that: 1) the water surface may reflect environment light and may generate caustics, which cause intensity differences between the synthesize view and the captured image; and 2) the water surface moves very fast, which causes motion blur in captured images and is not generated in synthesized view.

Effectiveness of Constraints. My next experiment aims to quantitatively verify whether or not the novel single-view consistency constraints can help to improve reconstruction accuracy on real data. Since ground truth surfaces are difficult to obtain for real waves, I here use the EPE measure Eq.(5.7) between the IBF and SBF computed at Camera 3 as explained above. If the IBF is properly estimated and the surface shape is accurately reconstructed, the two flow fields should be consistent.

As shown in Fig. 5.9, the presented approach achieves the smallest average EPE, which suggests that the 3D shape reconstructed from two views (Camera 1 and 2) is the most consistent with the pixel-point correspondences acquired from the additional view (Camera 3).

In addition, directly comparing the intensities of the captured image $\mathbf{I}_3$ and the synthesized image $SBF(\mathbf{B}_3)$ is problematic, due to the caustic and motion blur effects in $\mathbf{I}_3$. To circumvent such an issue, I binarize both the



Case	1	2	3	4	Ours
Mean	0.428	0.329	0.061	0.074	**0.047**
Stdev.	0.070	0.067	0.026	0.035	**0.014**

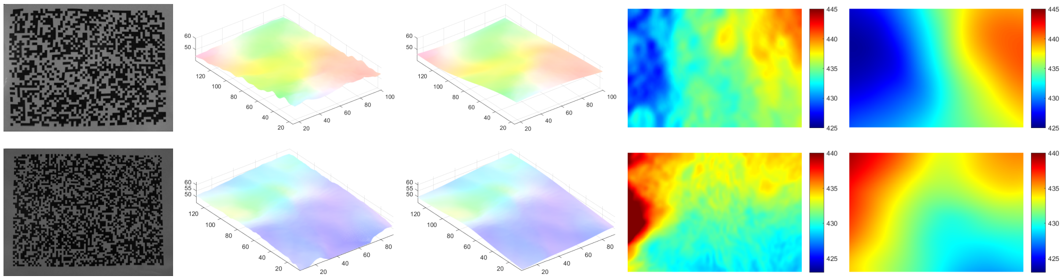
Figure 5.10: Quantitative evaluation on results generated under different constraints using Wave 1. The top figure plots MAE between binarized $\mathbf{I}_3$ and binarized $SBF(\mathbf{B}_3)$ as a function of frame ID. The bottom table shows the corresponding mean and standard deviation (stdev.) of MAE among all frames. Note that, here the evaluation results based on MAE are consistent with the ones based on average EPE as shown in Fig. 5.9.

images and evaluate the algorithm based on the mean absolute error (MAE) of the corresponding binarzied versions, as shown in Fig. 5.10.

Comparisons with [45]. Fig. 5.11 visually compares my approach and the traditional method [45] on my real waves. Because of the global formulation, my depths and normals are both consistent with the observed image distortions. My normals also reconcile with the obtained point clouds. In comparison, their normal map looks similar to ours but their depth map is noisy, which is consistent with the reported results in their paper.

Refractive Index Estimation. Following the previous work [45], I compute the average EPE Eq.(5.7) among 10 frames under different hypothesized refractive indices in the range of $[1.25, 1.41]$ with increments of 0.02. The minima of both curves in Fig. 5.12 are each close to the refractive index of water, *i.e.* 1.33.

Table 5.1 shows the processing time on the three captured waves.



(a) Captured Image $\mathbf{I}_1$ (b) Point Cloud [45] (c) My Point Cloud (d) Depth Map [45] (e) My Depth Map

Figure 5.11: Visual comparisons between the method of [45] and ours for an example frame of Wave 1 (top) and Wave 2 (bottom). Note that here I also impose a smoothness term in the algorithm of [45], *i.e.* Case 2, for fair comparisons.

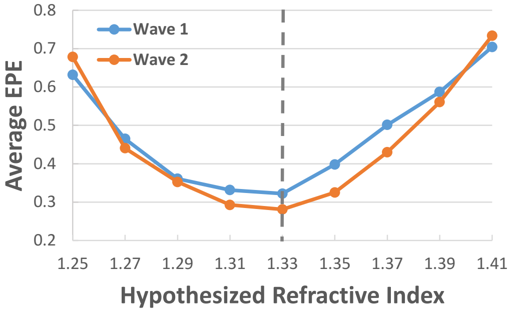


Figure 5.12: Average EPE Eq.(5.7) as a function of refractive index hypotheses for real data. The vertical dashed line indicate the refractive index of water, *i.e.* 1.33.

Water Wave		Wave 1	Wave 2	Wave 3
# of Frames		100	100	50
# of Correspondences Per Frame		101904	101904	101904
Refraction Index Estimation (in hours)		2.1	2.2	2.2
Reconstruction (in minutes per frame)	First Frame	11.3	11.6	11.0
	Other Frames	5.8	5.9	5.8

Table 5.1: Running time for the three real wave sequences.

Chapter 6

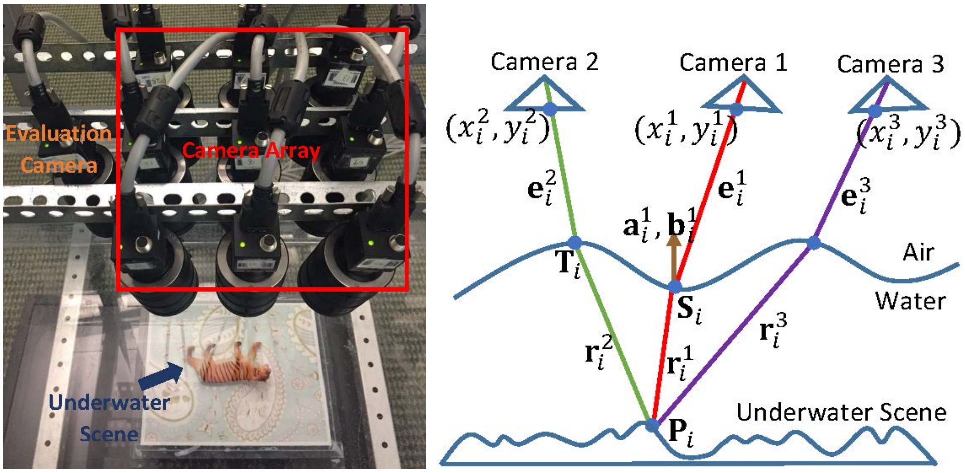
Jointly Reconstructing Water Surface and Underwater Scene

Chapter 5 presents a new method for reconstructing the 3D dynamic fluid surfaces under the assumption of the corresponding underwater scene is a known flat pattern. However, considering many underwater scenes are natural and thus non-flat in real life, this chapter aims to remove such an assumption by simultaneously recovering the 3D shape of both the wavy water surface and the moving underwater scene.

Specifically, a portable camera array system is constructed, which captures the scene from multiple viewpoints above the water. The correspondences across these cameras are estimated using an optical flow method and are used to infer the shape of the water surface and the underwater scene. I assume that there is only one refraction occurring at the water interface. Under this assumption, two estimates of the water surface normals should agree: one from Snell's law of light refraction and another from local surface structure. The experimental results using both synthetic and real data demonstrate the effectiveness of the presented approach.

6.1 Multi-View Acquisition Setup

As shown in Fig. 6.1(a), to capture the underwater scene, I build a small-scale, 3×3 camera array (highlighted in the red box) placed above the water surface. The cameras are synchronized and capture video sequences. For clarity, in the



(a) (b)

Figure 6.1: Acquisition setup using a camera array (a) and the corresponding imaging model illustrated in 2D (b). The evaluation camera in (a) is for accuracy evaluation only and is not used for 3D shape recovery.

following, I refer to the central camera in the array as the *reference* view, and the other cameras as the *side* views. Similar to the traditional multi-view triangulation-based framework for land-based 3D reconstruction, the 3D shapes of both the water surface and the underwater scene are represented in the reference view. Notice that an additional camera, referred to as the *evaluation* camera, is also used to capture the underwater scene at a novel view, which is for accuracy assessment in my real experiments and is presented in detail in Sec. 6.3.2.

Fig. 6.1(b) further illustrates the imaging model in 2D. I set Camera 1 as the reference camera and Camera $k \in \Pi$ as the side cameras, where Π is $\{2, 3, \cdots\}$. For each pixel (x_i^1, y_i^1) in Camera 1, the corresponding camera ray $\mathbf{e}_i^1$ gets refracted at the water surface point $\mathbf{S}_i$. Then the refracted ray $\mathbf{r}_i^1$ intersects with the underwater scene at point $\mathbf{P}_i$. The underwater scene point $\mathbf{P}_i$ is also observed by the side cameras through the same water surface but at different interface locations.

My approach builds upon the correspondences across multiple views. Specifically, I compute the optical flow field between the reference camera and each of the side cameras. Take side Camera 2 for example, for each pixel (x_i^1, y_i^1)

of Camera 1, I estimate the corresponding projection (x_i^2, y_i^2) of $\mathbf{P}_i$ in Camera 2, by applying the variational optical flow estimation method [7]. Suppose that the intrinsic and extrinsic parameters of the camera array are calibrated beforehand and fixed during capturing, I can easily compute the corresponding camera ray $\mathbf{e}_i^2$ of ray $\mathbf{e}_i^1$. The same procedure of finding correspondences applies to the other side views and each single frame is processed analogously.

After the above step, I obtain a sequence of the inter-view correspondences of the underwater scene. Below, I present a new reconstruction approach that solves the following problem: *Given the dense correspondences of camera rays* $\{\mathbf{e}^1 \Leftrightarrow \mathbf{e}^k, k \in \Pi\}$ *of each frame, how to recover the point set* $\mathbf{P}$ *of the underwater scene, as well as the depths and the normals of the dynamic water surface?*

6.2 Multi-View Reconstruction Approach

I tackle the problem using an optimization-based scheme that imposes a normal consistency constraint. Several prior works [45], [55], including the works of Chapter 4 and 5, have used such a constraint for water surface reconstruction. Here I show that, based on the similar form of normal consistency, I can simultaneously reconstruct dynamic water and underwater surfaces using multi-view data captured from a camera array. The key insight is that, at each water surface point, the normal estimated using its neighboring points should agree with the normal obtained based on the law of light refraction.

6.2.1 Normal Consistency at Reference View

As mentioned in Sec. 6.1, I represent the water surface by a depth map $\mathbf{D}$ and the underwater scene by a 3D point set $\mathbf{P}$, both in the reference view. In particular, as shown in Fig. 6.1(b), for each pixel in Camera 1, I have *four* unknowns: the depth $\mathbf{D}_i$ of point $\mathbf{S}_i$ and the 3D coordinates of point $\mathbf{P}_i$.

Given the camera ray $\mathbf{e}_i^1$, I can compute the 3D coordinates of $\mathbf{S}_i$ when a depth hypothesis $\mathbf{D}_i$ is assumed. At the same time, connecting the hypothesized point $\mathbf{P}_i$ and point $\mathbf{S}_i$ gives us the refracted ray direction $\mathbf{r}_i^1$. Then, the normal of $\mathbf{S}_i$ can be computed based on Snell's law, which is called the *Snell* normal in this chapter and denoted by $\mathbf{a}_i^1$. Here superscript 1 in $\mathbf{a}_i^1$ indicates that $\mathbf{a}_i^1$ is estimated using ray $\mathbf{e}_i^1$ of Camera 1. Consider the normal $\mathbf{a}_i^1$, the camera ray $\mathbf{e}_i^1$ and the refracted ray $\mathbf{r}_i^1$ are co-planar as stated in Snell's law. Hence, I can express $\mathbf{a}_i^1$ as a linear combination of $\mathbf{e}_i^1$ and $\mathbf{r}_i^1$, *i.e.* $\mathbf{a}_i^1 = \Psi(\eta_a \mathbf{e}_i^1 - \eta_f \mathbf{r}_i^1)$, where η_a and η_f are the refractive index of air and fluid, respectively. I fix $\eta_a = 1$ and $\eta_f = 1.33$ in my experiments. $\Psi()$ is a function defining the operation of vector normalization.

On the other hand, the normal of a 3D point can be obtained by analyzing the structure of its nearby points [57]. Specifically, suppose that the water surface is spatially smooth, at each point $\mathbf{S}_i$, I fit a local polynomial surface from its neighborhood and then estimate its normal based on the fitted surface. In practice, for a 3D point (x, y, z), I assume its z component can be represented by a quadratic function of the other two components:

$$z(x, y) = w_1 x^2 + w_2 y^2 + w_3 xy + w_4 x + w_5 y + w_6, \tag{6.1}$$

where $w_1, w_2 \ldots, w_6$ are unknown parameters. Stacking all quadratic equations of the set $\mathcal{N}_i$ of the neighboring points of $\mathbf{S}_i$ yields:

$$\mathbf{A}(\mathcal{N}_i)\mathbf{w}(\mathcal{N}_i) = \mathbf{z}(\mathcal{N}_i) \Leftrightarrow \begin{bmatrix} x_1^2 & y_1^2 & x_1 y_1 & x_1 & y_1 & 1 \\ & & \cdots & & & \\ x_m^2 & y_m^2 & x_m y_m & x_m & y_m & 1 \\ & & \cdots & & & \end{bmatrix} \times \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_6 \end{bmatrix} = \begin{bmatrix} z_1 \\ \vdots \\ z_m \\ \vdots \end{bmatrix}, \tag{6.2}$$

where $\mathbf{A}(\mathcal{N}_i)$ is a $|\mathcal{N}_i| \times 6$ matrix calculated from $\mathcal{N}_i$, and $|\mathcal{N}_i|$ the size of $\mathcal{N}_i$. $\mathbf{z}(\mathcal{N}_i)$ is a $|\mathcal{N}_i|$ dimensional vector. After getting the parameter vector $\mathbf{w}(\mathcal{N}_i)$, the normal of point (x, y, z) in this quadratic surface is estimated as the normalized cross product of two vectors: $[1, 0, \frac{\partial}{\partial x} z(x, y)]$ and $[0, 1, \frac{\partial}{\partial y} z(x, y)]$. Plugging in the 3D coordinates of $\mathbf{S}_i$, I obtain its normal $\mathbf{b}_i^1$, which is referred to as the *Quadratic* normal in this chapter.

So far, given the camera ray set $\mathbf{e}^1$ of Camera 1, I obtain two types of normals at each water surface point, which should be consistent if the hypothesized depth $\mathbf{D}$ and point set $\mathbf{P}$ are correct. I thus define the normal consistency error as:

$$E_i^1(\mathbf{D}, \mathbf{P}, \mathbf{e}_i^1) = \|\mathbf{a}_i^1 - \mathbf{b}_i^1\|_2^2 \tag{6.3}$$

at ray $\mathbf{e}_i^1$. Next, I show how to measure the normal consistency term at the side views using their camera ray sets $\{\mathbf{e}^k, k \in \Pi\}$, the point set $\mathbf{S}$ estimated from the depth hypothesis $\mathbf{D}$, and the hypothesized point set $\mathbf{P}$.

6.2.2 Normal Consistency at Side Views

I take side Camera 2 for illustration and the other side views are analyzed in a similar fashion. As shown in Fig. 6.1(b), point $\mathbf{P}_i$ is observed by Camera 2 through the water surface point $\mathbf{T}_i$. Similarly, I have the *Snell* normal $\mathbf{a}_i^2$ and the *Quadratic* normal $\mathbf{b}_i^2$ at $\mathbf{T}_i$.

To compute the *Snell* normal $\mathbf{a}_i^2$ via Snell's law, the camera ray $\mathbf{e}_i^2$ and the refracted ray $\mathbf{r}_i^2$ are required. $\mathbf{e}_i^2$ is acquired beforehand in Sec. 6.1. Considering the point hypothesis $\mathbf{P}_i$ is given, $\mathbf{r}_i^2$ can be obtained if the location of $\mathbf{T}_i$ is known. Hence, the problem of estimating normal $\mathbf{a}_i^2$ is reduced to the problem of locating the first-order intersection between ray $\mathbf{e}_i^2$ and the water surface point set $\mathbf{S}$. A similar problem has been studied in ray tracing [1]. In practice, I first generate a triangular mesh for $\mathbf{S}$ by creating a Delaunay triangulation of 2D pixels of Camera 1. I then apply the Bounding Volume Hierarchy-based ray tracing algorithm [36] to locate the triangle that $\mathbf{e}_i^2$ intersects. Using the neighboring points of that intersecting triangle, I fit a local quadratic surface as described in Sec. 6.2.1, and the final 3D coordinates of $\mathbf{T}_i$ is obtained by the standard ray-polynomial intersection procedure. Meanwhile, the fitted quadratic surface gives us the *Quadratic* normal $\mathbf{b}_i^2$ of point $\mathbf{T}_i$.

In summary, given each ray $\mathbf{e}_i^k$ of each side Camera k, I obtain two normals $\mathbf{a}_i^k$ and $\mathbf{b}_i^k$. The congruity between them results in the normal consistency error:

$$E_i^k(\mathbf{D}, \mathbf{P}, \mathbf{e}_i^k) = \|\mathbf{a}_i^k - \mathbf{b}_i^k\|_2^2, \ k \in \Pi. \tag{6.4}$$

6.2.3 Solution Method

Here I first discuss the feasibility of recovering both the water surface and the underwater scene using normal consistency at multiple views. Combining the error terms Eq.(6.3) at the reference view and Eq.(6.4) at the side views, I have:

$$E_i^k(\mathbf{D}, \mathbf{P}, \mathbf{e}_i^k) = 0, \text{ for each } i \in \Omega \text{ and } k \in \Phi, \tag{6.5}$$

where Ω is the set of all pixels of Camera 1, and $\Phi = \{1\} \cup \Pi$ the set of camera indices. Let $\bar{\mathtt{i}} = |\Omega|$ and $\bar{\mathtt{k}} = |\Phi|$ be the size of Ω and Φ, respectively. Assume that each camera ray $\mathbf{e}_i^1$ can find a valid correspondence in all side views, I get a total of $\bar{\mathtt{i}} \times \bar{\mathtt{k}}$ equations. Additionally, recall that I have 4 unknowns at each pixel of Camera 1, so I have $\bar{\mathtt{i}} \times 4$ unknowns. Hence, to make the problem solvable, I should have $\bar{\mathtt{i}} \times \bar{\mathtt{k}} \geq \bar{\mathtt{i}} \times 4$, which means that at least 4 cameras are required. In reality, some camera rays (*e.g.* those at corner pixels) of the reference view cannot locate a reliable correspondence in all side views because of occlusion or of the field of view. I essentially need more than four cameras.

Directly solving Eq.(6.5) is impractical due to the complex operations involved in computing the *Snell* and *Quadratic* normals. Therefore, I cast the reconstruction problem as minimizing the following objective function:

$$\min_{\mathbf{D},\mathbf{P}} \sum_{i \in \Omega} \sum_{k \in \Phi} E_i^k(\mathbf{D}, \mathbf{P}, \mathbf{e}_i^k) + \lambda \sum_{i \in \Omega} F_i(\mathbf{D}, \mathbf{e}_i^1), \tag{6.6}$$

where the first term enforces the proposed normal consistency constraint. The second term ensures the spatial smoothness of the water surface. In particular, I set

$$F_i(\mathbf{D}, \mathbf{e}_i^1) = \|\mathbf{A}(\mathcal{N}_i)\mathbf{w}(\mathcal{N}_i) - \mathbf{z}(\mathcal{N}_i)\|_2^2, \tag{6.7}$$

which measures the local quadratic surface fitting error using the neighborhood $\mathcal{N}_i$ of the water surface point $\mathbf{S}_i$. Adding such a polynomial regularization term helps to increase the robustness of my multi-view formulation, as demonstrated in my experiments in Sec. 6.3.1. Please also note that this smoothness term is only defined w.r.t Camera 1 since I represent my 3D shape in that view. λ is a parameter balancing the two terms.

While it may be tempting to enforce the spatial smoothness of underwater surface points $\mathbf{P}$ computed for different pixels as well, it is not imposed in my approach for the following reason. As shown in Fig. 6.2, when the light paths are refracted at the water surface, the neighborhood relationship among underwater scene points can be different from the neighborhood relationship

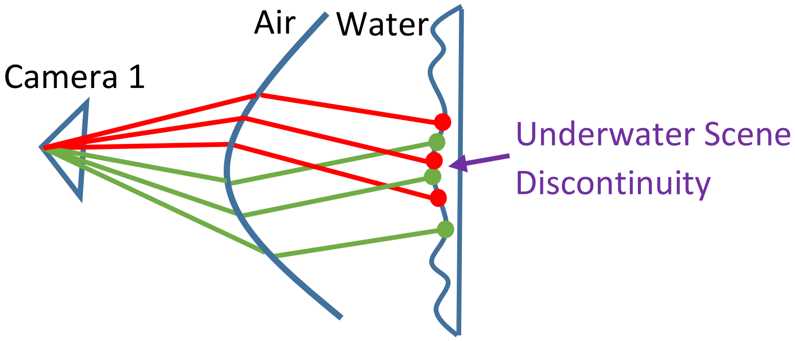


Figure 6.2: Discontinuity of underwater scene points. As indicated by the purple arrow, the red points are interlaced with the green points, although the red and green rays are each emitted from contiguous pixels.

among observed pixels in Camera 1. Hence, I cannot simply enforce that the 3D underwater surface points computed for adjacent camera rays are also adjacent.

Optimization

Computing the normal consistency errors in Eq.(6.6) involves some non-invertible operations such as vector normalization, making the analytic derivatives difficult to derive. To handle such a problem, I use the L-BFGS method [80] with numerical differentiation for optimization. However, calculating numerical derivatives is computationally expensive especially for a large-scale problem. I elaborately optimize my implementation by sharing common intermediate variables in derivative computation at different pixels. In addition, solving Eq.(6.6) is unfortunately a non-convex problem; hence, there is a chance of getting trapped by local minima. Here I adopt a coarse-to-fine optimization procedure commonly used in refractive surface reconstruction [53], [55], [62]. Specifically, I first downsample the correspondences acquired in Sec. 6.1 to 1/8 of the original resolution. I then use the results under the coarse resolution to initialize the optimization at the final scale.

Notice that the input of Eq.(6.6) is the multi-view data of a single time instance. Although it is possible to process all frames in a sequence simultaneously by concatenating them into Eq.(6.6), a large system with high computational complexity will be produced accordingly. In contrast, I process each frame independently and initialize the current frame using the results of the last one. Such a single-shot method effectively reduces the computational cost in terms of running time and memory consumption and, more importantly, can handle moving underwater scenes.

It is also noteworthy that, even when the underwater scene is strictly static, my recovered point set $\mathbf{P}$ could be different for different frames. This is because each point $\mathbf{P}_i$ can be interpreted as the intersection between the refracted ray $\mathbf{r}_i^1$ and the underwater scene, as shown in Fig. 6.1(b). When the water surface is flowing, because $\mathbf{S}_i$ relocates, the refracted ray direction is altered, and thus the intersection $\mathbf{P}_i$ is changed. My frame-by-frame formulation naturally handles such a varying representation of point set $\mathbf{P}$.

6.3 Experiments

The proposed approach is tested on both synthetic and real-captured data. Here I provide some implementation details. While computing the *Quadratic* normals at both the reference and side views, I set the neighborhood size to 5×5. The parameter λ is fixed at 2 units in the synthetic data and 0.1 mm in the real experiments. During the coarse-to-fine optimization of Eq.(6.6), the maximum number of L-BFGS iterations at the coarse scale is fixed to 2000 and 200 for synthetic data and real scenes, respectively, and is set to 20 at the full resolution in both cases. The linear least squares system Eq.(6.2) is solved via normal equations using Eigen [26]. As the *Snell* and *Quadratic* normal computations at different pixels are independent, I implement my algorithm in C++, with parallelizable steps optimized using OpenMP [16], on an 8-core PC with 3.2GHz Intel Core i7 CPU and 32GB RAM.

6.3.1 Synthetic Data

I use the ray tracing method [36] to generate synthetic data for evaluation. In particular, two scenes are simulated: a static Stanford Bunny observed through a sinusoidal wave: $z(x, y, t) = 2 + 0.1\cos(\pi(t+50)\sqrt{(x-1)^2 + (y-0.5)^2}/80)$, and a moving Stanford Dragon seen through a different water surface: $z(x, y, t) =$

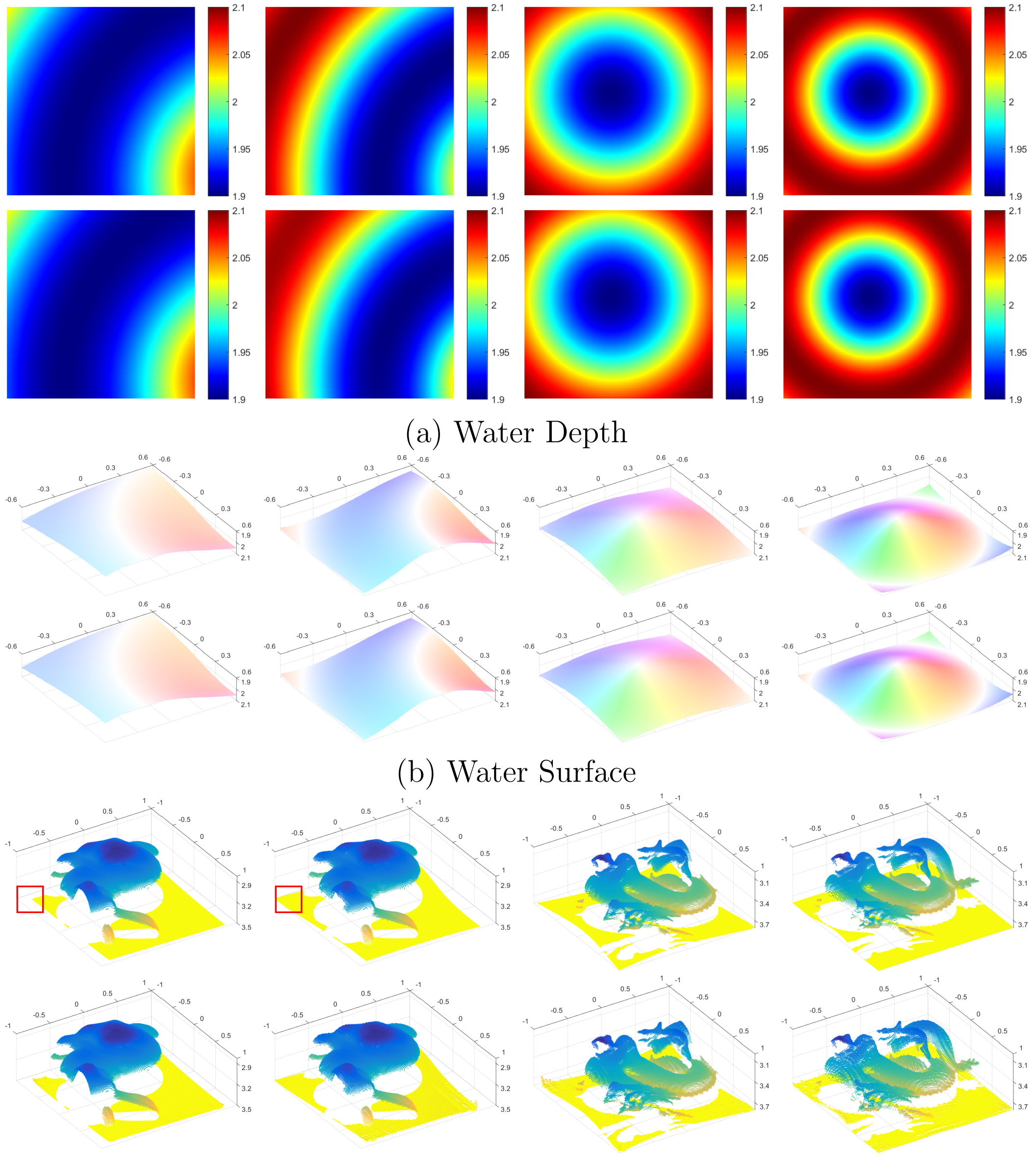
Table 6.1: Reconstruction errors of the synthetic Bunny scene and the Dragon scene. Here, for each scene, I list the average errors by considering all frames.

Scene	RMSE of $\mathbf{D}$ (*units*)	MAD of $\mathbf{a}^1$ (°)	MAD of $\mathbf{b}^1$ (°)	MED of $\mathbf{P}$ (*units*)
Bunny	0.006	0.76	0.77	0.01
Dragon	0.002	0.36	0.37	0.01

$2 - 0.1\cos(\pi(t+60)\sqrt{(x+0.05)^2 + (y+0.05)^2}/75)$. The Dragon object moves along a line with a uniform speed of 0.01 units per frame. Because of the different sizes of the two objects, I place the Bunny and Dragon objects on top of a flat backdrop positioned at $z = 3.5$ and $z = 3.8$, respectively. The synthetic scenes are captured using a 3×3 camera array. The reference camera is placed at the origin and the baseline between adjacent cameras in the array system is set to 0.3 and 0.2 for the Bunny and Dragon scene, respectively.

I start with quantitatively evaluating the proposed approach. Since my approach can return the depths and the normals of the water surface, and the 3D point set of the underwater scene, I employ the following measures for accuracy assessment: the root mean square error (RMSE) between the ground truth (GT) depths and the estimated depths $\mathbf{D}$, the mean angular difference (MAD) between the GT normals and the recovered *Snell* normals $\mathbf{a}^1$, the MAD between the true normals and the computed *Quadratic* normals $\mathbf{b}^1$, and the mean Euclidean distance (MED) between the reconstructed point set $\mathbf{P}$ of the underwater scene and the GT one. Table 6.1 shows my reconstruction accuracy by averaging over all frames. It is noteworthy that the average MAD of the *Snell* normals and that of the *Quadratic* normals are quite similar for both scenes, which coincides with my normal consistency constraint.

Fig. 6.3 visually shows the reconstruction results of several example frames. The complete sequences can be found in the supplementary materials [63]. Compared to the GT, my approach accurately recovers both the dynamic water surfaces and the underwater scenes. I can also observe that, while the underwater scene in the Bunny case is statically positioned in the simulation, different point clouds are obtained at different frames (see the red boxes in Fig. 6.3(c)), echoing my varying representation $\mathbf{P}$ of underwater points. Besides, with the frame-by-frame reconstruction scheme, my approach suc-



(a) Water Depth

(b) Water Surface

(c) Underwater Point Set

Figure 6.3: Visual comparisons with GT on two example frames of the Bunny scene (left two columns) and the Dragon scene (right two columns). In each subfigure, I show the GT and my result in the top and bottom row, respectively. (a) shows the GT water surface depth and the estimated one. (b) shows the GT water surface colored with the GT normal map, and the computed one colored with the *Quadratic* normals. The *Snell* normals are not shown here because they are similar to the *Quadratic* normals. (c) shows the GT point set of the underwater scene and the recovered one, where each point is colored with its z-axis coordinate. The red boxes highlight an obvious different region of the underwater point clouds of two different frames; see text for details.

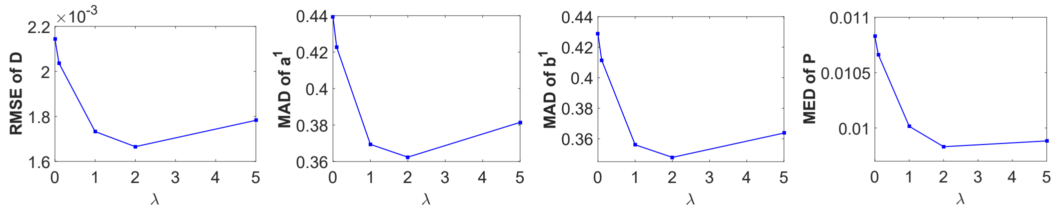


Figure 6.4: Different error measures as a function of the balancing parameter λ.

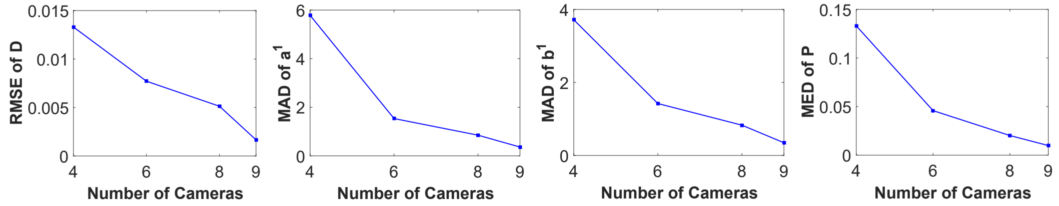


Figure 6.5: Different error measures as a function of the number of cameras used.

cessfully captures the movement of the underwater Dragon object. In short, accurate results are obtained for the two scenes generated using different water fluctuations, different underwater objects (static or moving), and data acquisition settings, which demonstrate the robustness of my approach.

I then adjust the weight λ in Eq.(6.6) to validate the effect of the polynomial smoothness term Eq.(6.7). Here I use the Dragon scene for illustration. As shown in Fig. 6.4, when $\lambda = 0$, the method depends on the normal consistency prior only. Explicitly applying a smoothness term with a proper setting $\lambda = 2$ performs favorably against other choices w.r.t. all error metrics. Fig. 6.5 further shows my reconstruction accuracy under different number of cameras used. Using a larger number of cameras gives a higher accuracy.

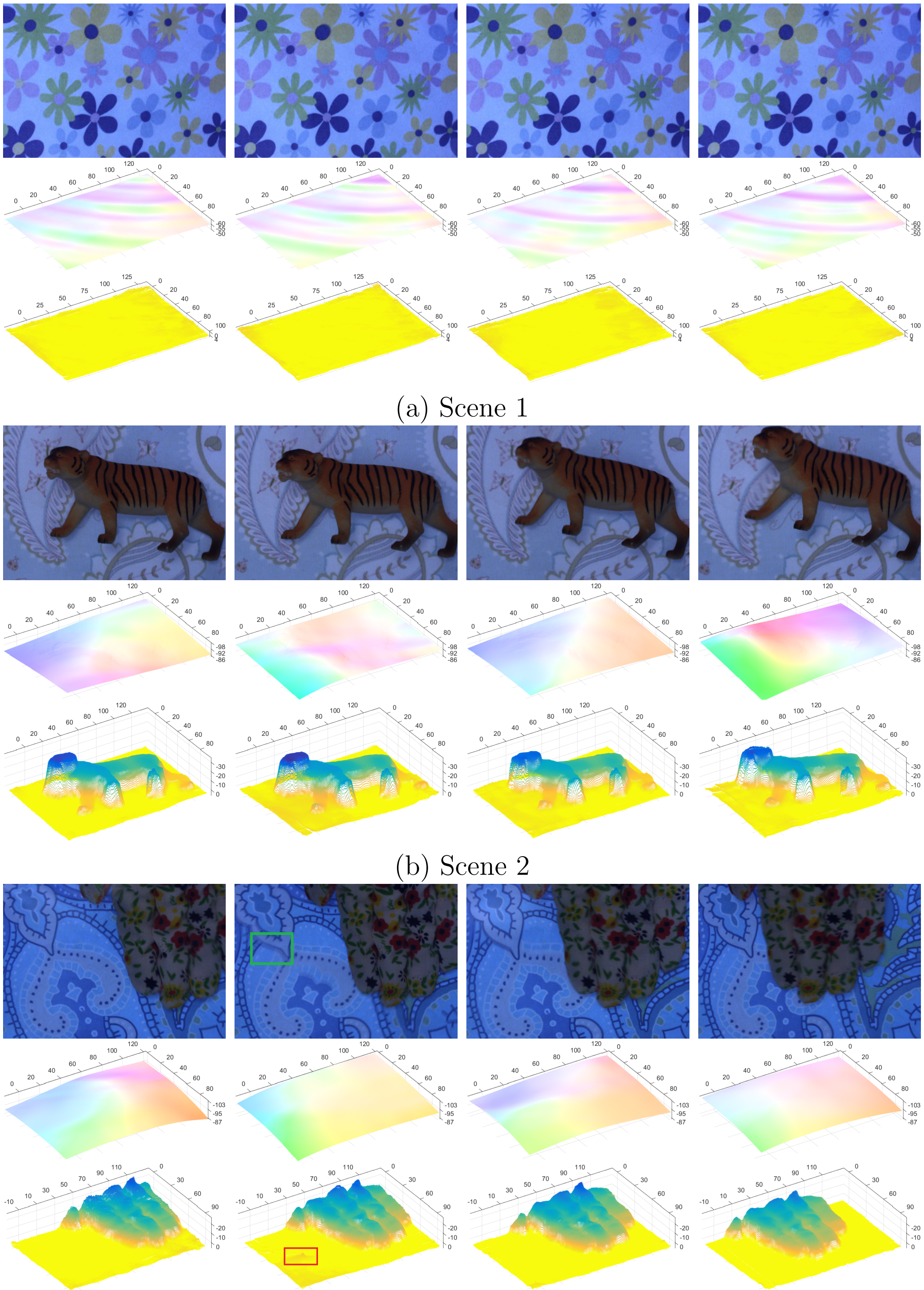
Here I also compare my approach with a baseline method that assumes a flat water surface, in which the underwater scene points are obtained by intersecting the refracted rays. Take the synthetic Dragon for example, by fixing the depth of the flat water surface to $1.9, 2.0, 2.1$, the MED of the recovered underwater points are, respectively, $0.61, 0.61, 0.62$, which are more than an order of magnitude higher than my MED 0.01 shown in Table 6.1.

6.3.2 Real Data

To capture real scenes from multiple viewpoints, I build a camera array system as shown in Fig. 6.1(a). Ten PointGrey Flea2 cameras are mounted on three metal frames to observe the bottom of a glass tank containing water. The cameras are connected to a PC via two PCI-E Firewire adapters, which enables us to use the software provided by PointGrey for synchronization. I use 9 cameras highlighted by the red box in Fig. 6.1(a) for multi-view 3D reconstruction, whereas the 10th camera, *i.e.* the evaluation camera, is used for *accuracy evaluation only.* I calibrate the intrinsic and extrinsic parameters of the cameras using a checkerboard [78]. The baseline between adjacent cameras is about 75mm and the distance between the camera array and the bottom of the tank is about 55cm. All the cameras capture video at 30fps with a resolution of 516 × 388. Flat textured backdrops are glued to the bottom of the tank, which is for facilitating optical flow estimation.

In order to verify my approach on real data, I first capture a simple scene: a flat textured plane placed at the bottom of the tank, which is referred to as Scene 1. The water surface is perturbed by continuously dripping water drops near one corner of the pattern. As shown in Fig. 6.6(a), my approach not only faithfully recovers the quarter-annular ripples propagated from the corner with the dripping water, but also accurately returns the 3D underwater plane without any prior knowledge of the flat structure. For accuracy assessment, I also fit a plane for the reconstructed underwater point set of each frame using RANSAC [22]. The MED between the reconstructed points and the fitted plane is 0.44mm by averaging over all frames. It is noteworthy that no post-processing steps like smoothing are performed here.

Two non-flat underwater scenes are then used to test my approach: (i) a toy tiger that is moved by strong water turbulence, and (ii) a moving hand in a textured glove. I refer to the two scenes as Scene 2 and Scene 3, respectively. In both cases, to generate water waves, I randomly disturb the water surface at one end of the tank. Fig. 6.6(b,c) shows several example results on Scene 2 and Scene 3, and the full videos can be found in the supplemental materials.



(a) Scene 1

(b) Scene 2

(c) Scene 3

Figure 6.6: Reconstruction results of four example frames of my captured scenes. In each subfigure, I show the captured image of the reference camera (top), the point cloud of the water surface colored with the *Quadratic* normals (middle), the point cloud of the underwater scene colored with the z-axis coordinates (bottom). Note that the motion blur (green box) in the captured image may affect the reconstruction result (red box).

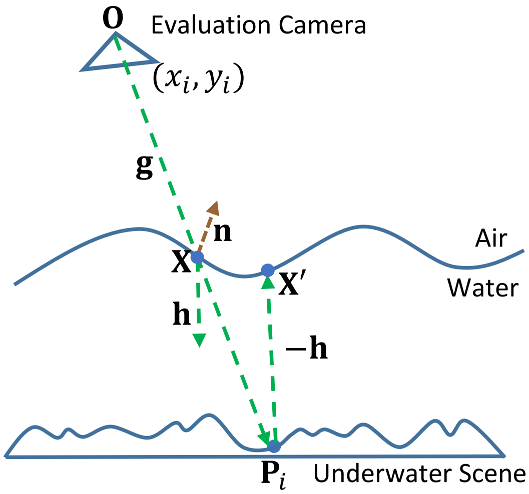


Figure 6.7: 2D illustration of forward projection through the water surface.

My approach successfully recovers the 3D shapes of the tiger object and the moving hand, as well as the fast evolving water surfaces.

Novel View Synthesis. Since obtaining GT shapes in my problem is difficult, I leverage the application of novel view synthesis to examine reconstruction quality. In particular, as shown in Fig. 6.1(a), I observe the scene at an additional calibrated view, *i.e.* the evaluation camera. At each frame, given the 3D point set of the underwater scene, I project each scene point to the image plane of the evaluation camera through the recovered water surface. Here such a forward projection is non-linear because of the light bending at the water surface. As shown in Fig. 6.7, for each underwater scene point $\mathbf{P}_i$, given the recovered point set $\mathbf{S}$ of the water surface and the camera center $\mathbf{O}$ of the evaluation camera, I aim to estimate the projection (x_i, y_i). Such a forward projection is non-linear because of light refraction at the water surface. Nevertheless, the projection (x_i, y_i) can be easily obtained if the corresponding interface point $\mathbf{X}$ is known.

Previous works [5], [37], [46] in underwater camera calibration propose an iterative procedure to locate $\mathbf{X}$ when the interface can be parametrized (*e.g.* it is flat or cylindrical). Here I present a modification of the method in [37] to locate $\mathbf{X}$ for each underwater point $\mathbf{P}_i$. Algorithm 1 shows my modified algorithm for this task, which differs from the previous method in [37] in two

Algorithm 1 Iterative Forward Projection for Point $\mathbf{P}_i$

```
Input: point set $\mathbf{S}$ of the water surface, underwater scene point $\mathbf{P}_i$, the intrinsic and extrinsic parameters of the evaluation camera, threshold $\epsilon = 10^{-6}$ and $T = 200$
Output: projection $(x_i, y_i)$
1:  initialize $\mathbf{X}$ as the intersection between the water point set $\mathbf{S}$ and ray $\overrightarrow{\mathbf{OP}_i}$

2:  repeat
3:    compute ray direction $\mathbf{g} := \overrightarrow{\mathbf{OX}}$
4:    estimate the normal $\mathbf{n}$ of $\mathbf{X}$ by fitting a local quadratic surface
5:    compute ray direction $\mathbf{h}$ using Snell's law, given $\mathbf{g}$ and $\mathbf{n}$
6:    shoot a ray from $\mathbf{P}_i$ along direction $-\mathbf{h}$, and this ray intersects with the water surface at $\mathbf{X}'$
7:    if the distance $\|\mathbf{X} - \mathbf{X}'\|_2^2 < \epsilon$ then
8:      go to Step 13
9:    else
10:     $\mathbf{X} := (\mathbf{X} + \mathbf{X}')/2$
11:   end if
12: until the number of iterations is greater than $T$
13: project $\mathbf{X}$ to the image plane of the evaluation camera
```

aspects. Firstly, since the interface is assumed to be cylindrical in [37], they use the standard ray-cylinder intersection procedure to intersect a ray with the interface. In comparison, my reconstructed water surface cannot be simply parametrized using a cylinder. I instead apply the ray tracing-based method presented in Section 6.2.2. Secondly, to estimate the normal of an interface point, their method again utilizes the cylindrical parametrization, whereas my modification is based on the local quadratic surface fitting as discussed in Section 6.2.1.

Specifically, I start with connecting point $\mathbf{P}_i$ with the evaluation camera center $\mathbf{O}$ and initializing $\mathbf{X}$ as the intersection between the water surface $\mathbf{S}$ and ray $\overrightarrow{\mathbf{OP}_i}$. I then compute ray $\mathbf{g}$ by connecting $\mathbf{X}$ and $\mathbf{O}$, and estimate the normal $\mathbf{n}$ of $\mathbf{X}$. By Snell's law, the refracted ray $\mathbf{h}$ is computed as:

$$\mathbf{h} = \eta \mathbf{g} + \left(-\eta \mathbf{n} \cdot \mathbf{g} - \sqrt{1 - \eta^2 \left(1 - (\mathbf{n} \cdot \mathbf{g})^2\right)} \right) \mathbf{n}, \tag{6.8}$$

where η is the ratio of the refractive indices of air and water. I set $\eta = \frac{1}{1.33}$ in my implementation. I then shoot a ray from the underwater point $\mathbf{P}_i$ along the negative direction of $\mathbf{h}$ and estimate the intersection $\mathbf{X}'$ between the shot

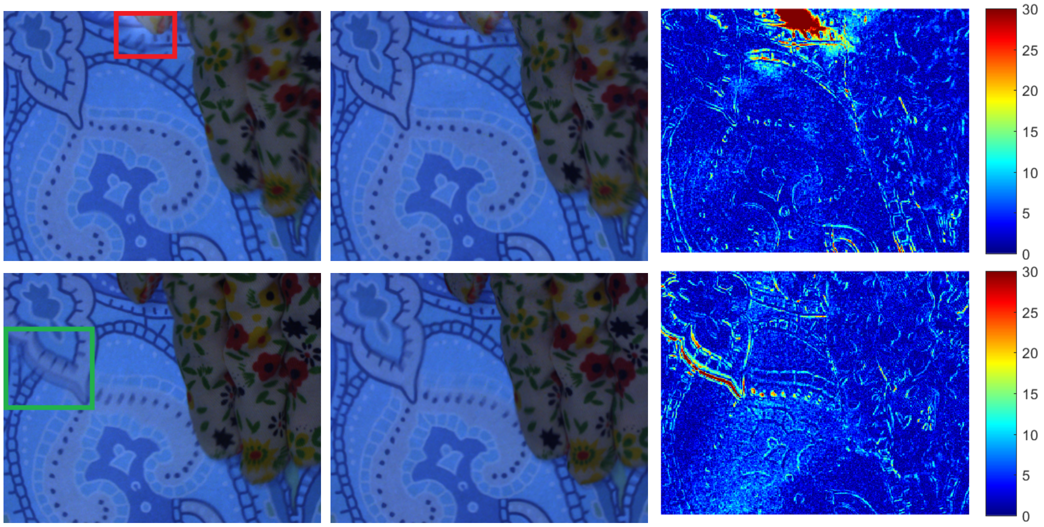


Figure 6.8: View synthesis on two example frames (top and bottom) of Scene 3. From left to right, it shows the images captured using the evaluation camera, the synthesized images and the absolute difference maps between them. The effects of specular reflection (red box) and motion blur (green box) can be observed in the captured images. These effects cannot be synthesized, leading to higher differences in the corresponding areas.

ray and the water surface. If the distance between $\mathbf{X}$ and $\mathbf{X}'$ is larger than a threshold ϵ, I compute the average point $(\mathbf{X}+\mathbf{X}')/2$ as the new value of $\mathbf{X}$ and iterate until their distance is small enough. Finally, I project $\mathbf{X}$ to the image plane of the evaluation camera using the conventional linear projection model [28]. Then, the final synthesized image at the evaluation camera is obtained using bilinear interpolation.

Fig. 6.8 shows that the synthesized images and the captured ones look quite similar, which validates the accuracy of my approach. Take Scene 2 and Scene 3 for example, the average peak signal-to-noise ratio by comparing the synthesized images to the captured images is 30dB and 31dB, respectively.

Running Time. For my real-captured data, each scene contains 100 frames and each frame has 119,808 water surface points and 119,808 underwater scene points. It takes about 5.5 hours to process each whole sequence, as shown in Table 6.2.

Table 6.2: Average running time of the three real scenes.

Scene	Scene 1	Scene 2	Scene 3
Optical Flow Estimation (minutes per frame)	0.74	0.74	0.77
3D Reconstruction (minutes per frame)	2.55	2.50	2.52

Chapter 7

Conclusion and Future Work

This thesis presents several new methods for 3D reconstruction in the presence of light refraction. I start by studying the problem of environment matting for reflective and refractive objects. Then, I present a class of 3D reconstruction methods for static transparent objects, dynamic fluid surfaces, and natural underwater scenes.

7.1 Summary

In Chapter 3, I present a novel frequency-based environment matting approach, which mainly addresses two major limitations of existing approaches. First, by leveraging CS, I simplify the data acquisition process of the conventional frequency-based environment matting [81]. Second, by augmenting with phase information, I further reduce the measurement cost and accelerate the expensive signal reconstruction process in CS, while accurately locating the contributing sources at the pixel level.

In Chapter 4, I present a refraction-based approach for reconstructing transparent objects. I first develop a simple acquisition setup that uses a pair of cameras and one monitor. Compared to existing methods, my system is non-intrusive, and requires no special devices or precise light source movement. By introducing a novel position-normal consistency constraint, I propose an optimization framework which can simultaneously reconstruct the 3D positions and normals of both the front and back surfaces. Note that many existing methods can only reconstruct either the depth or the normal of a single surface. In addition, I show that it is possible to estimate the refractive index of transparent objects using only two views.

In Chapter 5, I revisit the problem of dynamic refraction stereo [45] by presenting a new global optimization-based framework. I first formulate an objective function which couples both the conventional cross-view normal consistency constraint and the new single-view normal consistency priors that take local surface geometry into consideration. By solving all surface points at the same time, I obtain accurate and consistent depths and normals. Most importantly, my approach successfully avoids the fundamental limitation of previous methods that require using surface integration without accurate boundary conditions. Furthermore, I develop a novel error metric which can reliably estimate the refractive index of liquid in a computer vision fashion. It is also noteworthy that my reconstructed fluid surfaces are highly accurate for the application of novel view synthesis, which cannot be achieved in existing methods.

In Chapter 6, I present a novel approach for a 3D reconstruction problem: recovering underwater scenes through dynamic water surfaces. My approach exploits multiple viewpoints by constructing a portable camera array. After acquiring the correspondences across different views, the unknown water surface and underwater scene can be estimated through minimizing an objective function under a normal consistency constraint. My approach is validated using both synthetic and real data. To my best knowledge, this is the first approach that can handle both dynamic water surfaces and dynamic underwater scenes, whereas the previous work [76] uses a single view and cannot handle moving underwater scenes.

7.2 Limitations and Future Work

7.2.1 Environment Matting

The proposed approach in Chapter 3 addresses some limitations of the previous environment matting method [81]. Nevertheless, one limitation remains unsolved in my approach. That is, for acceleration purpose, I assume that the unknown light transportation matrix $\mathbf{W}$ can be decomposed into the element-wise product of a row vector and a column vector, *i.e.* Eq.(3.4). While this assumption has limited impacts on the algorithm's capability in handling contributions from broad areas of the background (*e.g.* "Pie Pan") or contributions from a large number of scattered sources, it may introduce visual artifacts when a foreground pixel has two non-adjacent dominating contributing regions. For example, the foreground highlighted by the red box in Figure 3.10 mainly receives lights from two regions in the background, one coming from refraction and the other from reflection. Under the above assumption, my approach may either locate additional but incorrect contributing sources or lose the weaker one. Thus the composited result may appear different from the photograph. In the future, I plan to address this ambiguity problem using additional diagonal patterns [13].

The environment matting problem is addressed in Chapter 3 using the proposed CS-based framework in the frequency domain. I argue that the proposed framework is also applicable to other many-to-one decomposition problems, *e.g.* dual photograph [60], where many projector pixels are merged into one camera pixel. CS has been utilized to reduce the complexity of data acquisition in dual photography [61], whereas the process of reconstructing the light reflection functions is very slow (almost 3 hours on a 24-node cluster for rendering a 256×256 image). Hence, I plan to apply my new frequency-based framework to tackle the dual photography problem in the near future.

7.2.2 Static Transparent Object Reconstruction

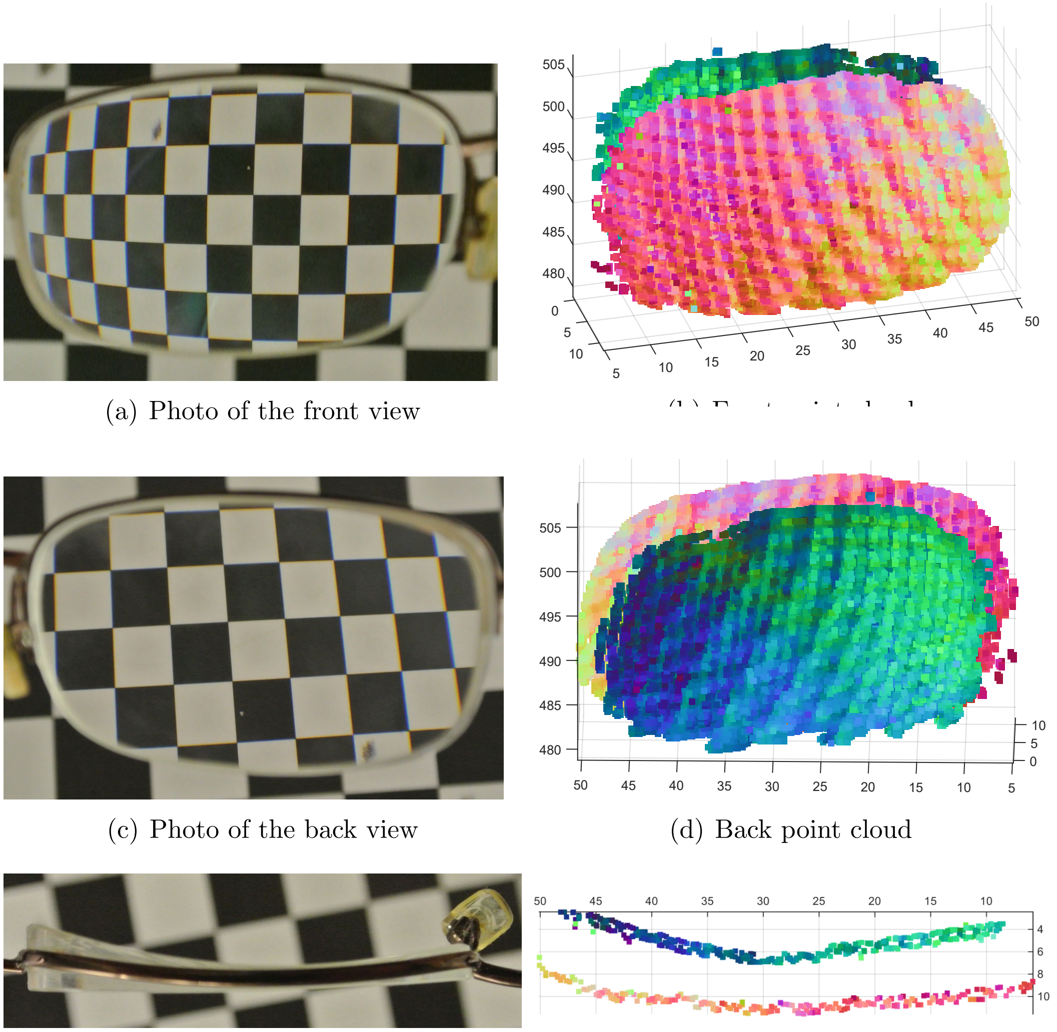
For static transparent object reconstruction, the approach presented in Chapter 4 works under the following assumptions: i) the object is solid and homogeneous, ii) the light path between the source and the camera goes through two refractions, and iii) the object surface is smooth enough so that surface normals can be reliably estimated using available sample points. Note that these assumptions are commonly used by refraction-based methods [38], [67]. Moreover, my acquisition is simple and inexpensive, but at the cost of capturing thousands of images since the ray-light source correspondences are required at each of the four monitor positions. As discussed above, replacing the monitor with light field probes [69] helps to reduce the number of images needed, but at the expense of loosing sampling density and not being able to handle colored objects. It is also noteworthy that the radiometric cues proposed in [12] may be incorporated to eliminate the monitor movements.

Essentially, my approach searches for a smooth surface that can best explain the observed ray-ray correspondences in terms of position-normal consistency. It implicitly assumes that there is only one feasible explanation for the observed correspondences. This assumption may not hold when the object is thin, in which case the refraction effects are mostly affected by the object thickness, rather than its shape. Hence, even though the reconstructed shape satisfies the position-normal consistency, it may not depict the real object shape. Fig. 7.1 shows such a failure case.

Although only two cameras are used in my experiments, the proposed optimization procedure Eq.(4.3) can be extended to more than two views so that different cameras can fully cover the transparent objects. How to compute the PCA and Snell normals under such settings certainly deserves further investigation.

7.2.3 Dynamic Fluid Surface Reconstruction

The approaches presented in Chapter 5 and Chapter 6 work under several assumptions that are also commonly used in state-of-the-art works in shape from refraction. Firstly, I assume that the medium (*i.e.* water in my case) is transparent and homogeneous, and thus light is refracted exactly once from water to air. Secondly, the water surface is assumed to be locally smooth, so that the normal of each surface point can be reliably estimated based on the local neighborhood. Thirdly, the optical flow method is applied to estimate the pixel-point correspondences when the underwater scene is flat, and to estimate the multi-view correspondences when the underwater scene is non-flat. To faithfully acquire the correspondence data, the underwater scene (either flat or non-flat) is assumed to be textured. The above assumptions may be violated in real-world scenarios. For example, water phenomena like bubbles, breaking



(a) Photo of the front view

(b) Front point cloud

(c) Photo of the back view

(d) Back point cloud

(e) Photo of the top view

(f) A slice of the model seen from the top view

Figure 7.1: Results of a failure case for a piece of myopia glass. Both front and back points are plotted in the same coordinate system. Since small normal perturbations on thin objects do not result in large ray correspondence changes, the estimated shape is quite noisy. Nevertheless, a slice through the center of the point cloud seen from the top (f) suggests that my approach properly models the thickness variation of the glasses.

waves, light scattering, may lead to multiple light bending events along a given light path. In the future, I plan to tackle the fluid surface reconstruction problem in these challenging scenarios.

Although promising reconstruction performance is demonstrated in Chapter 6, my approach is just a preliminary attempt to solving such a challenging problem. The obtained results are not perfect, especially at the boundary regions of the surfaces, as shown in Fig. 6.6. That is because those regions are covered by fewer views compared to other regions. To cope with this issue, I plan to build a larger camera array or use a light-field camera for video capture. In addition, occlusion is a known limitation in a multi-view setup because correspondence matching in occluded areas is not reliable. I plan to accommodate occlusion in my model in the near future.

References

[1] A. Adamson and M. Alexa, "Ray tracing point set surfaces," in *Shape Modeling International, 2003*, IEEE, 2003, pp. 272–279.

[2] M. Alterman, Y. Y. Schechner, and Y. Swirski, "Triangulation in random refractive distortions," in *Computational Photography (ICCP), 2013 IEEE International Conference on*, IEEE, 2013, pp. 1–10.

[3] Y. Asano, Y. Zheng, K. Nishino, and I. Sato, "Shape from water: Bispectral light absorption for depth recovery," in *European Conference on Computer Vision*, Springer, 2016, pp. 635–649.

[4] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[5] J. Belden, "Calibration of multi-camera systems with refractive interfaces," *Experiments in fluids*, vol. 54, no. 2, p. 1463, 2013.

[6] M. Ben-Ezra and S. K. Nayar, "What does motion reveal about transparency?" In *ICCV*, IEEE, 2003.

[7] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European Conference on Computer Vision*, Springer, 2004, pp. 25–36.

[8] E. J. Candes, "Compressive sampling," in *Proceedings oh the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, 2006, pp. 1433–1452.

[9] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[10] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *ECCV*, Springer, 2008, pp. 155–168.

[11] Y.-J. Chang and T. Chen, "Multi-view 3d reconstruction for scenes under the refractive plane with known vertical direction," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 2011, pp. 351–358.

[12] V. Chari and P. Sturm, "A theory of refractive photo-light-path triangulation," in *CVPR*, 2013, pp. 1438–1445.

[13] Y.-Y. Chuang, D. E. Zongker, J. Hindorff, B. Curless, D. H. Salesin, and R. Szeliski, "Environment matting extensions: Towards higher accuracy and real-time capture," in *Proceedings of ACM SIGGRAPH 00*, ACM Press/Addison-Wesley Publishing Co., 2000, pp. 121–130.

[14] Y.-Y. Chuang, D. E. Zongker, J. Hindorff, B. Curless, D. H. Salesin, and R. Szeliski, "Environment matting extensions: Towards higher accuracy and real-time capture," in *SIGGRAPH*, 2000, pp. 121–130.

[15] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt, "3d shape scanning with a time-of-flight camera," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, IEEE, 2010, pp. 1173–1180.

[16] L. Dagum and R. Menon, "Openmp: An industry standard api for shared-memory programming," *IEEE computational science and engineering*, vol. 5, no. 1, pp. 46–55, 1998.

[17] Y. Ding, F. Li, Y. Ji, and J. Yu, "Dynamic fluid surface acquisition using a camera array," in *ICCV*, IEEE, 2011.

[18] D. L. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[19] Q. Duan, J. Cai, and J. Zheng, "Compressive environment matting," *The Visual Computer*, pp. 1–14, 2014.

[20] Q. Duan, J. Cai, J. Zheng, and W. Lin, "Fast environment matting extraction using compressive sensing," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, IEEE, 2011, pp. 1–6.

[21] R. Ferreira, J. P. Costeira, and J. A. Santos, "Stereo reconstruction of a submerged scene," in *Iberian Conference on Pattern Recognition and Image Analysis*, Springer, 2005, pp. 102–109.

[22] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[23] J. Geng, "Structured-light 3d surface imaging: A tutorial," *Advances in Optics and Photonics*, vol. 3, no. 2, pp. 128–160, 2011.

[24] M. Goesele, H. Lensch, J. Lang, C. Fuchs, and H.-P. Seidel, "Disco: Acquisition of translucent objects," in *TOG*, ACM, 2004, pp. 835–844.

[25] M. Gong, J. M. Selzer, C. Lei, and Y.-H. Yang, "Real-time backward disparity-based rendering for dynamic scenes using programmable graphics hardware," in *Proceedings of Graphics Interface*, ACM, 2007, pp. 241–248.

[26] G. Guennebaud, B. Jacob, *et al.*, *Eigen v3*, http://eigen.tuxfamily.org, 2010.

[27] K. Han, K.-Y. K. Wong, and M. Liu, "A fixed viewpoint approach for dense reconstruction of transparent objects," in *CVPR*, 2015, pp. 4001–4008.

[28] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[29] J. Huang, X. Huang, and D. Metaxas, "Learning with dynamic group sparsity," in *Computer Vision, 2009 IEEE 12th International Conference on*, IEEE, 2009, pp. 64–71.

[30] M. B. Hullin, M. Fuchs, I. Ihrke, H.-P. Seidel, and H. P. Lensch, "Fluorescent immersion range scanning," *TOG*, pp. 87–87, 2008.

[31] N. Hurley and S. Rickard, "Comparing measures of sparsity," *Information Theory, IEEE Transactions on*, vol. 55, no. 10, pp. 4723–4741, 2009.

[32] I. Ihrke, K. N. Kutulakos, H. Lensch, M. Magnor, and W. Heidrich, "Transparent and specular object reconstruction," in *Computer Graphics Forum*, 2010, pp. 2400–2426.

[33] I. Ihrke, K. N. Kutulakos, H. P. Lensch, M. Magnor, and W. Heidrich, "State of the art in transparent and specular object reconstruction," in *EUROGRAPHICS*, 2008.

[34] B. Jähne, J. Klinke, and S. Waas, "Imaging of short ocean wind waves: A critical theoretical review," *JOSA A*, vol. 11, no. 8, pp. 2197–2209, 1994.

[35] Y. Ji, J. Ye, and J. Yu, "Reconstructing gas flows using light-path approximation," in *CVPR*, IEEE, 2013.

[36] T. L. Kay and J. T. Kajiya, "Ray tracing complex scenes," in *ACM SIGGRAPH computer graphics*, ACM, vol. 20, 1986, pp. 269–278.

[37] L. Kudela, F. Frischmann, O. Yossef, A. Uzan, S. Kollmannsberger, Z. Yosibash, and E. Rank, "Image-based mesh generation of tubular geometries under circular motion in refractive environments," *Machine Vision and Applications*, 2017.

[38] K. N. Kutulakos and E. Steger, "A theory of refractive and specular 3d shape by light-path triangulation," *International Journal of Computer Vision*, vol. 76, no. 1, pp. 13–29, 2008.

[39] C. L. Lawson and R. J. Hanson, *Solving least squares problems.* SIAM, 1974, vol. 161.

[40] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, *et al.*, "The digital michelangelo project: 3d scanning of large statues," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 2000, pp. 131–144.

[41] D. Liu, X. Chen, and Y.-H. Yang, "Frequency-based 3d reconstruction of transparent and specular objects," in *CVPR*, IEEE, 2014, pp. 660–667.

[42] M. Liu, R Hartley, and M Salzmann, "Mirror surface reconstruction from a single image.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 4, pp. 760–773, 2015.

[43] D. Miyazaki and K. Ikeuchi, "Shape estimation of transparent objects by using inverse polarization ray tracing," *PAMI*, pp. 2018–2030, 2007.

[44] N. J. Morris and K. N. Kutulakos, "Reconstructing the surface of inhomogeneous transparent scenes by scatter-trace photography," in *ICCV*, IEEE, 2007, pp. 1–8.

[45] ——, "Dynamic refraction stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1518–1531, 2011.

[46] C Mulsow, "A flexible multi-media bundle approach," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, vol. 38, pp. 472–477, 2010.

[47] H. Murase, "Surface shape reconstruction of a nonrigid transparent object using refraction and motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 1045–1052, 1992.

[48] P. Peers and P. Dutré, "Wavelet environment matting," in *Proceedings of the 14th Eurographics workshop on Rendering*, Eurographics Association, 2003, pp. 157–166.

[49] P. Peers, D. K. Mahajan, B. Lamond, A. Ghosh, W. Matusik, R. Ramamoorthi, and P. Debevec, "Compressive light transport sensing," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 1, p. 3, 2009.

[50] *Persistence of vision (tm) raytracer*, `http://www.povray.org/`.

[51] E. Petriu, Z Sakr, H. Spoelder, and A Moica, "Object recognition using pseudo-random color encoded structured light," in *Instrumentation and Measurement Technology Conference, 2000. IMTC 2000. Proceedings of the 17th IEEE*, IEEE, vol. 3, 2000, pp. 1237–1241.

[52] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C*. Cambridge university press Cambridge, 1996, vol. 2.

[53] Y. Qian, M. Gong, and Y. Hong Yang, "3d reconstruction of transparent objects with position-normal consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4369–4377.

[54] Y. Qian, M. Gong, and Y. H. Yang, "Frequency-based environment matting by compressive sensing," in *ICCV*, IEEE, 2015.

[55] ——, "Stereo-based 3d reconstruction of dynamic fluid surfaces by global optimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1269–1278.

[56] Y. Qian, Y. Zheng, M. Gong, and Y.-H. Yang, "Simultaneous 3d reconstruction for water surface and underwater scene," in *Proceedings of the European Conference on Computer Vision*, 2018.

[57] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, 2009.

[58] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[59] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*, IEEE, vol. 1, 2006, pp. 519–528.

[60] P. Sen, B. Chen, G. Garg, S. R. Marschner, M. Horowitz, M. Levoy, and H. Lensch, "Dual photography," in *ACM Transactions on Graphics (TOG)*, ACM, vol. 24, 2005, pp. 745–755.

[61] P. Sen and S. Darabi, "Compressive dual photography," in *Computer Graphics Forum*, Wiley Online Library, vol. 28, 2009, pp. 609–618.

[62] Q. Shan, S. Agarwal, and B. Curless, "Refractive height fields from single and multiple images," in *CVPR*, IEEE, 2012, pp. 286–293.

[63] *Supplemental videos*, `https://www.youtube.com/channel/UCFaEqO89z-AMuzeZqdi33Zg`.

[64] *Swarovski report*, `http://www.crystalandglassbeads.com/blog/2012/diamonds-cubic-zirconia-swarovski-whats-the-difference.html`.

[65] M. Tarini, H. P. Lensch, M. Goesele, and H.-P. Seidel, *3D acquisition of mirroring objects.* Max-Planck-Institut für Informatik, 2003.

[66] B. Trifonov, D. Bradley, and W. Heidrich, "Tomographic reconstruction of transparent objects," in *Proc. Eurographics Symposium on Rendering*, 2006, pp. 51–60.

[67] C.-Y. Tsai, A. Veeraraghavan, and A. C. Sankaranarayanan, "What does a light ray reveal about a transparent object?" In *ICIP*, IEEE, 2015.

[68] J. V. Uspensky, "Introduction to mathematical probability," Tech. Rep., 1937.

[69] G. Wetzstein, R. Raskar, and W. Heidrich, "Hand-held schlieren photography with light field probes," in *ICCP*, IEEE, 2011, pp. 1–8.

[70] G. Wetzstein, D. Roodnick, W. Heidrich, and R. Raskar, "Refractive shape from light field distortion," in *ICCV*, IEEE, 2011, pp. 1180–1186.

[71] Y. Wexler, A. W. Fitzgibbon, and A. Zisserman, "Image-based environment matting.," in *Rendering Techniques*, 2002, pp. 279–290.

[72] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 210–227, 2009.

[73] J. Xiao, *3d reconstruction is not just a low-level task: Retrospect and survey.*

[74] J. Ye, Y. Ji, F. Li, and J. Yu, "Angular domain reconstruction of dynamic 3d fluid surfaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 310–317.

[75] S.-K. Yeung, T.-P. Wu, C.-K. Tang, T. F. Chan, and S. Osher, "Adequate reconstruction of transparent objects on a shoestring budget," in *CVPR*, IEEE, 2011.

[76] M. Zhang, X. Lin, M. Gupta, J. Suo, and Q. Dai, "Recovering scene geometry under wavy fluid via distortion and defocus analysis," in *European Conference on Computer Vision*, Springer, 2014, pp. 234–250.

[77] X. Zhang and C. S. Cox, "Measuring the two-dimensional structure of a wavy water surface optically: A surface gradient detector," *Experiments in Fluids*, vol. 17, no. 4, pp. 225–237, 1994.

[78] Z. Zhang, "A flexible new technique for camera calibration," *PAMI*, pp. 1330–1334, 2000.

[79] L. Zheng, G. Li, and J. Sha, "The survey of medical image 3d reconstruction," in *Fifth International Conference on Photonics and Imaging in Biology and Medicine*, International Society for Optics and Photonics, 2007, 65342K–65342K.

[80] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software*, vol. 23, no. 4, pp. 550–560, 1997.

[81] J. Zhu and Y.-H. Yang, "Frequency-based environment matting," in *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, IEEE, 2004, pp. 402–410.

[82] D. E. Zongker, D. M. Werner, B. Curless, and D. H. Salesin, "Environment matting and compositing," in *Proceedings of ACM SIGGRAPH 99*, ACM Press / ACM SIGGRAPH / Addison Wesley Logman, 1999, pp. 205–214.