

# Single View Reconstruction of Transparent, Mirror and Diffuse Surfaces



**Kai Han**  
韓鎔

Department of Computer Science  
The University of Hong Kong

This dissertation is submitted for  
*Doctor of Philosophy*

April, 2018



To my parents.



## Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text.

---

Kai Han

April, 2018



## Acknowledgements

This thesis summarizes the results of my research conducted in the Computer Vision Group of the Department of Computer Science at The University of Hong Kong. It would not be possible without the generous help from people around. I would like to express my greatest gratitude to my supervisor, Dr. Kwan-Yee Kenneth Wong, for his constant support and thoughtful guidance at each stage of my study. It was under Dr. Wong's guidance that my research began to take root. I appreciate all his efforts to pave my way towards the scientific truth in computer vision.

I was lucky to work with many talented minds and would like to thank for their invaluable advice and enthusiastic help: Dr. Miaomiao Liu and Dr. Dirk Schnieders for my first two projects (Chapter 2 and Chapter 3); Dr. Xiao Tan for my third project (Chapter 4); Prof. Jean Ponce, Prof. Cordelia Schmid, Dr. Minsu Cho, Dr. Bumsub Ham, and Mr. Rafael S. Rezende for my fourth project (Chapter 5), which is partially done during my internship at WILLOW group (INRIA / École Normale Supérieure), Paris, France. In particular, the theory of semantic correspondence estimation in Chapter 5 is the result of a close collaboration with them, and Rafael produced the dense correspondences based on the results of semantic region matching.

I would like to show my sincere thanks to my colleagues and friends for their irreplaceable friendship. Thanks go to Dr. Xiaolong Zhu, Dr. Xiao Tan, Mr. Wei Liu, Mr. Chaofeng Chen, Mr. Ernest Cheung, Mr. Kaicheng Yu, Mr. Guanying Chen, Mr. Zhenfang Chen, Ms. Jingjing Zhang, Ms. Bingbin Liu, and Mr. Huiquan Zhou for their infectious enthusiasm in helping me solving problems and creating such a pleasant working atmosphere. I feel grateful for my friends Dr. Xuhui Jia, Mr.

Ruoxin Sang, Mr. Angran Lin, Mr. Haoyuan Zhang, Mr. Kan Wu, Mr. Yixiang Fang, Mr. Yudian Zhen, Mr. Guanbing Li, Mr. Xiaoguang Han, Mr. Yuxiang Yang, Mr. Changjian Li, Ms. Lingjie Liu, and Ms. Menglu Li for the enjoyable time of gathering.

Last, but not least, I wish to thank my parents and beloved one Hongjie Wu for their unconditional and everlasting love, which is the source of my motivation for research.



Abstract of thesis entitled

**“Single View Reconstruction of Transparent, Mirror and  
Diffuse Surfaces”**

Submitted by

**Kai Han**

for the degree of Doctor of Philosophy

at The University of Hong Kong

in April, 2018

3D reconstruction has been a fundamental problem in computer vision and has many applications. However, existing methods are mostly designed for diffuse surfaces under multiple viewpoints. This thesis tackles three reconstruction problems under a single view, namely, transparent object reconstruction, mirror surface reconstruction, and diffuse surface reconstruction. Besides, semantic correspondence, which is essential for not only 3D reconstruction but also image understanding, is also investigated in this thesis.

In the first part of this thesis, a novel and practical approach is presented for transparent object reconstruction under a fixed viewpoint. A simple and handy setup is introduced to alter the incident light paths before light rays enter the object, followed by a surface recovery method based on reconstructing and triangulating such incident light paths. Our approach does not need to explicitly model the complex interactions of light as it travels through the object, assuming neither any parametric form for shape of the object nor exact number of refractions and reflections occur when light travels through the object. It can handle a transparent object with a complex structure, with an unknown and even inhomogeneous refractive index.

This thesis then considers the problem of mirror surface reconstruction under a fixed viewpoint. We first derive an analytical solution to recover the camera projection matrix, and then optimize the camera projection matrix by minimizing reprojection

errors with a cross-ratio formulation. The mirror surface is finally reconstructed based on the optimized cross-ratio constraint. The proposed method only needs reflection correspondences as input and removes the restrictive assumptions of known motions,  $C^n$  continuity of the surface, and calibrated camera(s) that are being used by other existing methods. This greatly simplifies the challenging problem of mirror surface recovery.

In the third part of this thesis, a novel self-calibration method is introduced for single view diffuse surface reconstruction using an unknown mirror sphere. We first derive an analytical solution to recover the focal length of the camera given its principal point, and then introduce a robust algorithm to estimate accurate principal point and the focal length of the camera. Besides, we also present a novel approach for estimating both principal point and focal length of the camera when only a single image of the sphere is available. With the estimated camera intrinsics, the sphere position and a scaled 3D scene object can be obtained.

This thesis finally considers the problem of semantic correspondence estimation, which is crucial for 3D reconstruction as well as scene understanding. Most previous approaches to semantic correspondence focus on combining an effective spatial regularizer with hand-crafted features, or learning a correspondence model for appearance only. We proposed a convolutional neural network architecture, called SCNet, for learning a geometrically plausible model for semantic correspondence. SCNet uses region proposals as matching primitives, and explicitly incorporates geometric consistency. (461 words)

# Contents

<b>Contents</b>	<b>xi</b>
<b>List of Figures</b>	<b>xv</b>
<b>List of Tables</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Contributions . . . . .	4
1.3 Thesis Outline . . . . .	5
<b>2 Single View Transparent Object Reconstruction</b>	<b>9</b>
2.1 Introduction . . . . .	9
2.2 Related Work . . . . .	11
2.3 Shape Recovery of Transparent Objects . . . . .	15
2.3.1 Notations and Problem Formulation . . . . .	15
2.3.2 Setup and Assumptions . . . . .	16
2.3.3 Dense Refraction Correspondences . . . . .	16
2.3.4 Light Path Triangulation . . . . .	17
2.3.5 Surface Normal Reconstruction . . . . .	19
2.4 Recovery of Thin Transparent Objects . . . . .	20
2.4.1 Setup and Assumptions . . . . .	20
2.4.2 Surface Reconstruction . . . . .	21

---

2.5	Discussions . . . . .	23
2.5.1	Total Internal Reflection . . . . .	23
2.5.2	Object Analysis . . . . .	23
2.5.3	Single Refraction Approximation . . . . .	25
2.5.4	Limitations . . . . .	27
2.6	Experimental Evaluation . . . . .	28
2.6.1	Synthetic Data . . . . .	28
2.6.2	Real Data . . . . .	34
2.7	Conclusions . . . . .	41
<b>3</b>	<b>Single View Mirror Surface Reconstruction</b>	<b>43</b>
3.1	Introduction . . . . .	43
3.2	Related Work . . . . .	45
3.3	Acquisition Setup . . . . .	48
3.4	A Closed-form Solution . . . . .	48
3.4.1	Plücker Coordinates . . . . .	49
3.4.2	Line Projection Matrix . . . . .	50
3.4.3	Estimating the Line Projection Matrix . . . . .	51
3.4.4	Enforcing Constraints . . . . .	53
3.5	Cross-ratio Based Formulation . . . . .	54
3.6	Evaluation . . . . .	56
3.6.1	Synthetic Data . . . . .	57
3.6.2	Real Data . . . . .	60
3.7	Discussions and Conclusions . . . . .	63
<b>4</b>	<b>Single View Diffuse Surface Reconstruction</b>	<b>65</b>
4.1	Introduction . . . . .	65
4.2	Related Work . . . . .	67
4.3	Theoretical Background . . . . .	69
4.4	Estimating Camera Intrinsic Parameters . . . . .	70

---

4.4.1	Focal Length . . . . .	71
4.4.2	Principal Point . . . . .	73
4.5	Shape Recovery . . . . .	76
4.6	Experimental Results . . . . .	77
4.6.1	Synthetic Data . . . . .	78
4.6.2	Real Data . . . . .	80
4.7	Discussions and Conclusions . . . . .	81
<b>5</b>	<b>Learning Semantic Correspondence</b>	<b>85</b>
5.1	Introduction . . . . .	85
5.2	Related Work . . . . .	87
5.3	Our Approach . . . . .	88
5.3.1	Model . . . . .	89
5.3.2	Similarity Function and Geometry Kernel . . . . .	90
5.3.3	Gradient-based Learning . . . . .	92
5.4	SCNet Architecture . . . . .	93
5.5	Experimental Evaluation . . . . .	95
5.5.1	Experimental Details . . . . .	95
5.5.2	Proposal Flow Components . . . . .	97
5.5.3	Flow Field . . . . .	100
5.6	Conclusions . . . . .	105
<b>6</b>	<b>Conclusions</b>	<b>107</b>
6.1	Summary . . . . .	107
6.2	Future Work . . . . .	109
	<b>Appendix A Pose Estimation with Reflection Correspondences</b>	<b>111</b>
	<b>Appendix B Line Projection Matrix and Camera Projection Matrix</b>	<b>115</b>
	<b>Appendix C Back-propagation for Hough Voting</b>	<b>117</b>

References

121

# List of Figures

1.1	Examples of diffuse, mirror and transparent surfaces . . . . .	2
1.2	A mirror reflecting the surrounding scene . . . . .	3
2.1	Real reconstruction setup and examples of transparent objects . . . . .	10
2.2	Partition of a light path . . . . .	14
2.3	Sample images of a transparent hemisphere . . . . .	18
2.4	PBC reconstruction and FEP estimation . . . . .	18
2.5	PBC and visual ray construction, and surface estimation for thin objects	22
2.6	A total internal reflection example . . . . .	24
2.7	Refractions at the surfaces of a parallel planar plate . . . . .	25
2.8	Error induced by the single refraction approximation . . . . .	26
2.9	Reconstruction of a synthetic semi-ellipsoid using the first method . . .	29
2.10	RMS errors for the positions and normals of the FEPs reconstructed using the first method . . . . .	30
2.11	Reconstruction of a concave transparent object using the first method .	32
2.12	Reconstruction of two synthetic thin objects using the second method .	32
2.13	Error of surface normals recovered using the second method for a syn- thetic thin convex cone . . . . .	34
2.14	Error of surface normals recovered using the second method for a syn- thetic thin spherical shell . . . . .	35
2.15	Reconstruction results of the first method on real data . . . . .	36
2.16	Two views of the reconstructed surface using the first method . . . . .	37

---

2.17	Reconstruction result for <i>bottle</i> using the first method . . . . .	39
2.18	Two views of the <i>bottle</i> reconstructed using the first method . . . . .	39
2.19	Real reconstruction setup for thin transparent objects . . . . .	40
2.20	Reconstruction results of the second method on real data . . . . .	41
2.21	Two views of the surface reconstructed using the second method . . . . .	42
3.1	A stationary uncalibrated camera observing the reflections of a reference plane undergoing an unknown motion . . . . .	44
3.2	Setup used for mirror surface reconstruction . . . . .	48
3.3	Visualization of line projection matrix . . . . .	51
3.4	Minimizing point-to-line distance does not guarantee minimizing point- to-point distance . . . . .	55
3.5	Camera projection matrix and mirror surface points are recovered by minimizing reprojection errors . . . . .	55
3.6	RMS errors for the mirror <i>Stanford bunny</i> . . . . .	57
3.7	Reconstructed surface of the mirror <i>Stanford bunny</i> . . . . .	58
3.8	Comparison with a calibrated method for mirror surface recovery . . . . .	59
3.9	<i>sauce boat</i> and <i>two spheres</i> in real experiments . . . . .	60
3.10	Reconstructions of <i>sauce boat</i> and <i>two spheres</i> . . . . .	61
4.1	A perspective camera located at $\mathbf{O}$ observes a sphere $S$ of radius $r_s$ . . . . .	69
4.2	Removing the effect of $\mathbf{T}$ . . . . .	72
4.3	An example of camera intrinsics estimation from the conic of a single sphere . . . . .	76
4.4	A perspective camera observes the reflections of a scene point . . . . .	77
4.5	Synthetic experiment results under different noise levels . . . . .	79
4.6	Reconstruction of scene points under noise . . . . .	79
4.7	Real experiment setup and reflections on a sphere at four distinct positions	80
4.8	Reconstructed corners of a box . . . . .	82
4.9	Recovered diffuse surfaces . . . . .	82



---

5.1	Learning semantic correspondence . . . . .	86
5.2	The SCNet architectures . . . . .	92
5.3	Performance of SCNet on PF-PASCAL . . . . .	98
5.4	Region matching examples . . . . .	100
5.5	Quantitative comparison of dense correspondence . . . . .	101
A.1	Plane pose estimation with reflection correspondences . . . . .	111



# List of Tables

2.1	Statistics for our real experiment with the first method . . . . .	36
2.2	Reconstruction errors of <i>hemisphere</i> using the first method . . . . .	38
2.3	Reconstruction error measurements of <i>ornament</i> . . . . .	38
2.4	Reconstruction errors of <i>bottle</i> using the first method . . . . .	40
3.1	Estimation error of <i>Stanford bunny</i> under noise . . . . .	58
3.2	Camera intrinsic and extrinsic estimation error under different noise level	59
3.3	Real experiments evaluation on mirror surface recovery . . . . .	62
4.1	Estimation of camera intrinsic parameters . . . . .	81
4.2	RMS angle error and length ratio error of the recovered box . . . . .	81
5.1	Runtime comparison. The time is the mean time cost (second) for each image pair in testing set of PF-PASCAL-RP-1000. . . . .	99
5.2	Per-class PCK on PF-PASCAL . . . . .	102
5.3	Fixed-threshold PCK on PF-WILLOW . . . . .	103
5.4	Results on Caltech-101 . . . . .	104
5.5	Results on PASCAL Parts . . . . .	105



# Chapter 1

## Introduction

### 1.1 Motivation

3D reconstruction has always been a hot topic in the field of computer vision, and has many important applications in robotics, augmented reality, video games, movie production, reverse engineering, etc.

Many sophisticated methods have been developed over the past few decades, and tremendous success has been achieved to reconstruct opaque objects with a diffuse surface. For examples, structure-from-motion methods [1–3], shape-from-silhouette methods [4, 5], shape-from-shading methods [6, 7], shape-from-shadows methods [8, 9], photometric stereo methods [10, 11], etc.

Most of these methods, however, cannot handle mirror surfaces and transparent objects due to the fact that the information they exploited is derived under the assumption of a diffuse surface whose appearance is *viewpoint independent* (see Fig. 1.1(a)). A mirror surface (or a transparent object) does not have a unique appearance of its own. Its appearance depends on light reflected (refracted) from its surrounding environment and is therefore *viewpoint dependent* (see Fig. 1.1(b,c)). In order to reconstruct mirror surfaces and transparent objects, special algorithms have to be designed based on the physical laws of reflection and refraction.

The difficulties in the study of mirror surfaces and transparent objects originate



Fig. 1.1 Examples of diffuse, mirror and transparent surfaces. (a) A teapot with a diffuse surface. (b) A teapot with a mirror surface. (c) A teapot with a transparent surface.

from the complex interactions of light. A mirror surface alters an incident light path by reflection at its surface. A transparent object, on the other hand, may alter an incident light path by reflection, refraction, absorption and scattering at both its exterior surface as well as its interior structure. Even with restrictive assumptions and special hardware setups, state-of-the-art methods can only handle mirror surfaces or transparent objects with a simple shape [12–15].

Meanwhile, it is not difficult for one to realize that there are as many (if not more) mirror surfaces and transparent objects as purely diffuse objects in our world (e.g., polished metallic parts, chromed surfaces, mirrors, liquids, glass, plastics, crystals and diamonds). Transparent objects reconstruction can also be used in detecting defects in windshields in automobile industry and defects in glass-wall-panels in building industry, and in shape analysis of crystals and diamonds, etc. Mirror surface reconstruction can be useful in mirror surface inspection and calibration of non-central camera systems, etc. Hence, the study of 3D model reconstruction cannot be considered completed without taking mirror surfaces and transparent objects into accounts.

Mirror and transparent surfaces can also be used to facilitate the reconstruction of diffuse surface. For example, mirrors can provide a wider field of view (see Fig. 1.2), which benefits 3D reconstruction by bringing a larger range of the scene around into images. By introducing one or more mirrors into the scene and observing the reflec-



Fig. 1.2 A mirror reflecting the surrounding scene.

tion(s) on the mirror(s), multiple observations of the same scene point can be obtained in a single viewpoint (e.g., [16]), which allows us to identify multiple light paths for each scene point. Thus, we can capture multi-view information from a single view, and this makes single view 3D reconstruction possible. Similarly, transparent objects can also be used to construct multiple light paths for each scene point under a fixed viewpoint (e.g., [17]).

The workhorse for 3D reconstruction is visual correspondence. High quality visual correspondences are also critical for image retrieval, image registration, and object recognition. In general, visual correspondence estimation spans the range from low-level feature matching (stereo) to high-level object or scene understanding (semantic). Stereo correspondences are formed from pixels of the same object or scene point in different images, while semantic correspondences are formed from pixels of the same/different object(s) or scene point(s) in different images having the same semantic meaning. Great successes have been achieved for stereo correspondence estimation based on hand-crafted features such as SIFT [18], HOG [19], Gray Code [20], Phase Shift [21], etc. Semantic correspondence estimation, however, still remains an open and challenging problem, due to the complex appearance variation and shape deforma-

tion of objects within the same category. It has a huge potential to benefit single view reconstruction of transparent, mirror and diffuse surfaces, since the correspondence for single view reconstruction can be considered as a special case of semantic correspondence. The reflection and refraction correspondences are distorted, but they depict exactly the same scene points, thus containing exactly the same semantic meaning.

## 1.2 Contributions

The main contributions of this thesis are:

- a fixed viewpoint approach to **transparent object reconstruction** based on refraction of light. Our approach does not need to explicitly model the complex interactions of light as it travels through the object, assuming neither any parametric form for shape of the object nor exact number of refractions and reflections occur when light travels through the object. It can handle a transparent object with a complex structure, with an unknown and even inhomogeneous refractive index. Preliminary results of this research have been published in [22, 23].
- a fixed viewpoint **mirror surface reconstruction** solution under an unknown motion of a reference plane and an uncalibrated camera. We propose (i) a closed-form (linear) solution for estimating the camera projection matrix from reflection correspondences; (ii) a cross-ratio based nonlinear formulation that allows a robust estimation of the camera projection matrix together with the mirror surface. Preliminary results of this research have been published in [24].
- a single view **diffuse surface reconstruction** method using an uncalibrated camera and an unknown mirror sphere. We propose (i) an analytical solution for recovering the focal length of a camera from an image of an unknown sphere given the principal point of the camera; (ii) a robust method for estimating both the principal point and focal length of a camera from multiple images of an unknown sphere placed at different positions; (iii) a novel method for estimating



both the principal point and focal length of a camera from just one single image of an unknown sphere. Preliminary results of this research have been published in [25].

- a convolutional neural network (CNN) for **establishing semantic correspondence**. We introduce a simple and efficient approach for learning to match regions using both appearance similarity and geometry consistency constraints. Our model achieves state-of-the-art results on several benchmarks, which demonstrate the advantage of taking geometry constraint into consideration. Preliminary results of this research have been published in [26].

## 1.3 Thesis Outline

The remainder of this thesis is organized as follows.

**Chapter 2** This chapter addresses the problem of reconstructing the surface shape of transparent objects. The difficulty of this problem originates from the viewpoint dependent appearance of a transparent object, which quickly makes reconstruction methods tailored for diffuse surfaces fail disgracefully. In this chapter, we develop a fixed viewpoint approach to dense surface reconstruction of transparent objects based on refraction of light. We introduce a simple setup that allows us to alter the incident light paths before light rays enter the object, and develop a method for recovering the object surface based on reconstructing and triangulating such incident light paths. Our proposed approach does not need to model the complex interactions of light as it travels through the object, neither does it assume any parametric form for the object shape nor the exact number of refractions and reflections taken place along the light paths. It can therefore handle transparent objects with a relatively complex shape and structure, with unknown and even inhomogeneous refractive index. We also show that for thin transparent objects, our proposed acquisition setup can be further simplified by adopting a single refraction approximation. Experimental results on both synthetic

and real data demonstrate the feasibility and accuracy of our proposed approach.

**Chapter 3** This chapter addresses the problem of mirror surface reconstruction, and a solution based on observing the reflections of a moving reference plane on the mirror surface is proposed. Unlike previous approaches which require tedious work to calibrate the camera, our method can recover both the camera intrinsics and extrinsics together with the mirror surface from reflections of the reference plane under at least three unknown distinct poses. Existing work has demonstrated that 3D poses of the reference plane can be registered in a common coordinate system using reflection correspondences established across images. This leads to a bunch of registered 3D lines formed from the reflection correspondences. Given these lines, we first derive an analytical solution to recover the camera projection matrix through estimating the *line projection matrix*. We then optimize the camera projection matrix by minimizing reprojection errors computed based on a cross-ratio formulation. The mirror surface is finally reconstructed based on the optimized cross-ratio constraint. Experimental results on both synthetic and real data are presented, which demonstrate the feasibility and accuracy of our method.

**Chapter 4** In this chapter, we develop a novel self-calibration method for single view 3D reconstruction using a mirror sphere. Unlike other mirror sphere based reconstruction methods, our method needs neither the intrinsic parameters of the camera, nor the position and radius of the sphere be known. Based on eigen decomposition of the matrix representing the conic image of the sphere and enforcing a repeated eigenvalue constraint, we derive an analytical solution for recovering the focal length of the camera given its principal point. We then introduce a robust algorithm for estimating both the principal point and the focal length of the camera by minimizing the differences between focal lengths estimated from multiple images of the sphere. We also present a novel approach for estimating both the principal point and focal length of the camera in the case of just one image of the sphere. With the estimated camera intrinsic parameters, the position(s) of the sphere can be readily retrieved from the

eigen decomposition(s) and a scaled 3D reconstruction follows. Experimental results on both synthetic and real data are presented, which demonstrate the feasibility and accuracy of our approach.

**Chapter 5** This chapter addresses the problem of establishing *semantic correspondences* between images depicting different instances of the same object or scene category. Previous approaches focus on either combining a spatial regularizer with hand-crafted features, or learning a correspondence model for appearance only. We propose instead a convolutional neural network architecture, called *SCNet*, for learning a *geometrically* plausible model for semantic correspondence. SCNet uses region proposals as matching primitives, and explicitly incorporates geometric consistency in its loss function. A comparative evaluation on several standard benchmarks demonstrates that the proposed approach substantially outperforms both recent deep learning architectures and previous methods based on hand-crafted features.

**Chapter 6** This chapter summarizes the theories and algorithms developed in this dissertation, followed by a brief discussion of potential future work.



# Chapter 2

## Single View Transparent Object Reconstruction

### 2.1 Introduction

Reconstructing a 3D model of an object from its 2D images has always been a hot topic in the field of computer vision. It has many important applications in robotics, augmented reality, video games, movie production, reverse engineering, etc. Despite the problem of 3D model reconstruction has virtually been solved for opaque objects with a diffuse surface, the literature is relatively sparse when it comes to shape recovery of transparent objects. It is still very challenging and remains an open problem. The viewpoint dependent appearance of a transparent object quickly renders reconstruction methods tailored for diffuse surfaces useless, and most of the existing methods for transparent object reconstruction are still highly theoretical. In fact, even with restrictive assumptions and special hardware setups, state-of-the-art methods can only handle transparent objects with a very simple shape. Meanwhile, it is not difficult to see that there exist many transparent objects in our world (e.g., glasses, plastics, crystals and diamonds). Hence, the study of 3D model reconstruction cannot be considered completed without taking transparent objects into account.

As mentioned previously, the difficulty of reconstructing a transparent object origi-

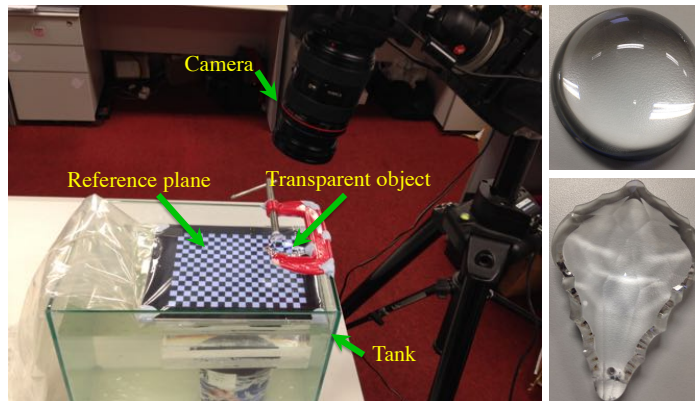


Fig. 2.1 Real reconstruction setup and examples of transparent objects.

nates from its viewpoint dependent appearance. A transparent object may alter a light path by reflection, refraction, absorption and scattering at both its exterior surface as well as its interior structure. A number of existing work attempted to reconstruct a transparent object by exploiting specular highlights produced on the object surface [13, 14]. This approach considers only reflection of light taken place at the object surface, and greatly simplifies the problem by making it not necessary to consider the complex interactions of light as it travels through the object. However, refraction of light is indeed an important and unique characteristic of transparent objects. It provides information on surface shape and should not be ignored. On the other hand, methods based on reflection of light often work only under very restrictive assumptions and precisely controlled environments, making them not very practical.

In this chapter, we focus our study in dense surface reconstruction of transparent objects. We introduce a fixed viewpoint approach to recovering the surface of a transparent object based on refraction of light. Like those methods that are based on specular highlights, our fixed viewpoint approach does not need to explicitly model the complex interactions of light as it travels through the object. We present a simple setup (see Fig. 2.1) that allows us to alter the incident light paths before light rays enter the object by immersing the object partially in a liquid, and develop a method for recovering the surface of a transparent object through reconstructing and trian-

gulating these incident light paths. We also show that for thin transparent objects, the acquisition setup can be further simplified by adopting a single refraction approximation. Compared with existing methods, our proposed method has the following benefits:

- It does not assume any parametric form for the shape of a transparent object.
- It can handle a transparent object with a complex structure, with an unknown and inhomogeneous refractive index.
- It considers only the incident light paths before light rays enter a transparent object, and makes no assumption on the exact number of refractions and reflections taken place as light travels through the object.
- The proposed setup is simple and inexpensive.

The rest of the chapter is organized as follows. Section 2.2 briefly reviews existing techniques in the literature for shape recovery of transparent objects. Section 2.3 describes our proposed approach to dense surface reconstruction of transparent objects in detail. Section 2.4 introduces our simplified approach to thin transparent object reconstruction. Section 2.5 discusses the problem of total internal reflection and objects that are suitable for our approach. Experimental results on both synthetic and real data are presented in Section 2.6, followed by conclusions in Section 2.7.

## 2.2 Related Work

Great efforts have been devoted to the problem of transparent object reconstruction in the past two decades. To formulate this problem, existing methods often make assumptions such as orthographic projection [27–29],  $C^n$  continuity of the surface [29], known exact number of refractions along each light path [30–32], etc. In [27, 28], Murase reconstructed a rippling water surface from the average observed coordinates of an underwater pattern under orthographic projection. Morris and Kutulakos [33]

solved a similar problem with an unknown refractive index of the liquid using two calibrated cameras and a known reference pattern. In [29], Shan *et al.* introduced a framework for optimizing a refractive height field from a single image under the assumptions of an orthographic camera, known background, single refractive material and differentiable height field. In [34], Hata *et al.* used structured light and genetic algorithm to estimate the shape of a transparent paste drop on a board. Ben-Ezra and Nayar [35] assumed a parametric form for the shape of a transparent object and estimated the shape parameters under the assumptions of a known camera motion and a distant background. In [30, 31], Kutulakos and Steger categorized reconstructible specular scenes, and developed algorithms for depth map computation in the cases where refraction/reflection of light occurs exactly once and twice respectively. Following the same fashion, Tsai *et al.* [32] demonstrated two depth-normal ambiguities for transparent object recovery assuming the light path refracts exactly twice. In [36], Zuo *et al.* developed an interactive specular and transparent object reconstruction system based on visual hull refinement given the silhouettes under multiple views and labeled contours of the object in sparse key frames. In [37], Qian *et al.* introduced a method to recover transparent objects by solving an optimization function with a position-normal consistency constraint, under the assumption of two refractions along each light path. Their system consists of two cameras and one display serving as a light source for correspondence estimation.

Many hardware setups have also been designed to recover the surfaces of transparent objects. In [38], Wetzstein *et al.* proposed a single image approach to reconstructing thin refractive surfaces using light field probes. In [39], Ding *et al.* introduced a  $3 \times 3$  camera array to acquire correspondences for fluid surface recovery. In [40], Eren *et al.* determined the surface shape of a glass object using laser surface heating and thermal imaging. In [41], Ihrke *et al.* dyed water with a fluorescent chemical and presented a level set method for reconstructing a free flowing water surface from multi-video input data by minimizing a photo-consistency error computed using ray-tracing. Miyazaki and Ikeuchi [42] proposed an iterative method to estimate the front



surface shape of a transparent object by minimizing the difference between observed polarization data and polarization raytracing result under the assumptions of a known refractive index, a known illumination distribution and a known back surface shape. In [43], Trifonov *et al.* introduced a visible light tomographic reconstruction method by immersing a transparent object into a fluid with a similar refractive index. The 3D shape was recovered by building the light paths within the fluid and the object. In [44], Hullin *et al.* embedded a transparent object into fluorescence and reconstructed the object surface by detecting the intersections of the visible laser sheets with the visual rays. A similar light sheet range scanning approach was introduced by Narasimhan *et al.* in [45] for acquiring object geometry in the presence of a scattering medium. In [46], O’Toole *et al.* developed the structured light transport (SLT) technique. Based on SLT, they implemented an imaging device that allows one-shot indirect-invariant imaging for reconstructing transparent and mirror surfaces using structured light. In [47], Ma *et al.* reformulated the intensity transport equation in terms of light fields, and presented a technique for refractive index field reconstruction using coded illumination. In [48], Ji *et al.* estimated the refractive index field of a gas volume by establishing ray-to-ray correspondences using a light field probe, and reconstructed the light paths through the refractive index field using a variational method based on Fermat’s Principle.

Like specular surfaces, transparent objects also exhibit reflection properties. Hence, reflection correspondences designed for specular surface reconstruction (e.g., [15, 49]) can also be adopted for the reconstruction of transparent objects. In [50], Morris and Kutulakos introduced *scatter-trace* of a pixel and recovered the exterior surface of a transparent object using the non-negligible specular reflection component. Similarly, Yeung *et al.* [51] exploited specular highlights and proposed a dual-layered graph-cut method to reconstruct the surface of a solid transparent object. In [52], Chari and Sturm introduced a method that integrates radiometric information into light path triangulation for reconstruction of transparent objects from a single image. In [53], Liu *et al.* proposed a frequency based method for establishing correspondences

on transparent and mirror surfaces, and reconstruction can then be done using any stereo methods.

Note that existing solutions for surface reconstruction of transparent objects often work only under restrictive assumptions (e.g., known refractive index, single refractive material, known exact number of refractions, non-negligible reflection of light, orthographic projection), using special hardware setups (e.g., light field probes, laser surface heating with thermal imaging, dying liquids with fluorescent chemical, immersing objects into liquids with similar refractive indexes), or for a particular class of objects (e.g., with known parametric model/average shape). There exists no general solution to this challenging and open problem.

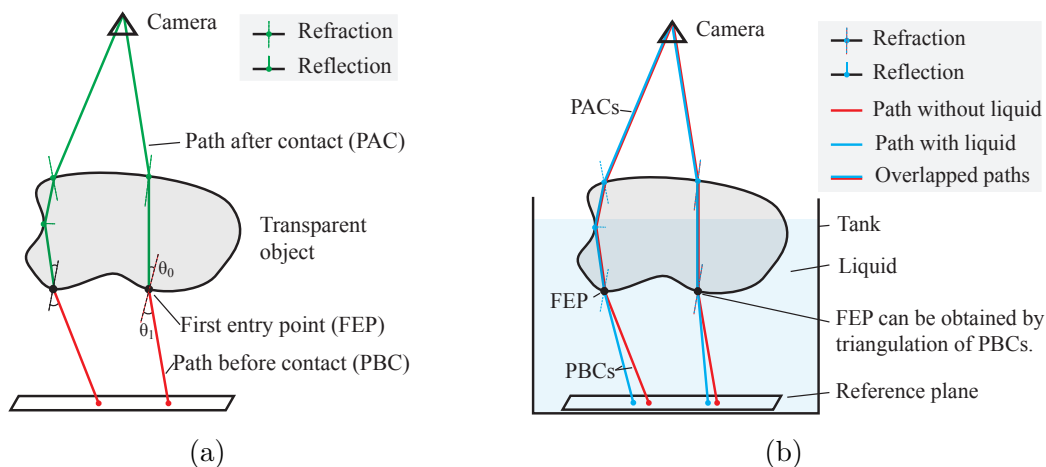


Fig. 2.2 (a) A light path through an object is partitioned into two parts, namely i) the *path before contact* (PBC) which originates from the reference pattern to the *first entry point* (FEP) on the object surface (i.e., the red paths) and ii) the *path after contact* (PAC) that originates from FEP, passes through the interior of the object and terminates at the optical center of the camera (i.e., the green paths). (b) The PBC can be altered by filling the tank with a liquid, and the FEP can be recovered by triangulating two PBCs.

In this chapter, we develop a fixed viewpoint approach to dense surface reconstruction of transparent objects based on altering and triangulating the incident light paths before light rays enter the object. We present a simple setup that allows us to alter the incident light paths by means of refraction of light. Under this proposed setup,

the segment of a light path between the first entry point on the object surface and the optical center of the camera remains fixed. This allows us to ignore the details of the complex interactions of light inside the object. Compared with existing methods, our proposed approach (1) assumes neither a known nor homogeneous refractive index of the object; (2) places no restriction on the exact number of refractions and reflections taken place along a light path; and (3) assumes no parametric form for the object shape. This allows our approach to handle transparent objects with a relatively complex structure.

For thin transparent objects, we show that our acquisition setup can be further simplified by adopting a single refraction approximation. Such an approximation has been used by existing methods for recovering liquid surfaces (e.g., [33, 39]), one-side flatten (e.g., [29]) or thin transparent surfaces (e.g., [38]). The altering of the incident light paths can be achieved by the object itself without using any extra medium, and the surface can be recovered using the same formulation as the general approach.

## 2.3 Shape Recovery of Transparent Objects

### 2.3.1 Notations and Problem Formulation

To solve the surface reconstruction problem, we consider a set of light paths originating from a reference pattern placed behind a transparent object, passing through the object and eventually reaching the image plane. We partition every such light path into two parts, namely (i) the *path before contact* (PBC) which originates from the reference pattern and ends at the *first entry point* (FEP) on the object surface (see the red paths in Fig. 2.2(a)) and (ii) the *path after contact* (PAC) which originates from the FEP, passes through the interior of the object and finally terminates at the optical center of the camera (see the green paths in Fig. 2.2(a)). We can now reformulate the surface reconstruction problem into estimating the FEP. The approach we take to tackle this problem is by altering the PBC while fixing the PAC for each light path. This enables us to ignore the details of the complex interactions of light inside the object, and

recover the FEP by triangulating the PBCs. In the next section, we present a simple setup that allows us to alter the PBCs by means of refraction of light.

### 2.3.2 Setup and Assumptions

In our proposed setup, a camera is used to capture images of a transparent object in front of a reference pattern. The camera and the object are kept fixed with respect to each other to ensure the PACs remain unchanged for all the image points. The reference pattern is placed at two distinct positions and is used for reconstructing the PBCs. As mentioned before, our approach is based on altering and triangulating the PBCs. To achieve this, we employ a water tank and immerse the object partially into a liquid so as to alter the PBCs by means of refraction of light (see Fig. 2.2(b)). Two images of the transparent object are acquired for each position of the reference pattern, one without liquid in the tank and one with liquid in the tank. By calibrating the positions of the reference pattern and establishing correspondences between image points and points on the reference pattern, we can reconstruct two PBCs for each image point, one in air and one in the liquid, respectively. The FEP can then be recovered by triangulating these two PBCs.

Note that our proposed approach does not require the prior knowledge of the refractive index of the object or that of the liquid. If, however, the refractive index of the liquid is known a priori, it is possible to also recover the surface normal at each FEP. The only assumption made in our approach is that the PACs remain unchanged when the object is immersed partially into the liquid.

### 2.3.3 Dense Refraction Correspondences

Before we can triangulate PBCs to recover the FEP, we first need to reconstruct the PBCs from the images. To achieve this, we first calibrate the two distinct positions of the reference pattern using [54]. It is then straightforward to reconstruct the PBC for an image point by locating a correspondence point on the reference pattern under

each of the two distinct positions in the same medium (i.e., with/without liquid in the tank). It is obvious that the quality of the correspondences will have a direct effect on the quality of the reconstruction. There exist many methods for establishing correspondences [55], such as Gray Code [20], Phase Shift [21], etc. However, these methods often can only provide sparse correspondences with limited precisions (e.g., a small patch of pixels is mapped to a small region on a reference plane due to finite discretization). In this work, we would like to establish quasi-point-to-point correspondences between the image and the reference pattern. We employ a portable display screen (e.g., an iPad) to serve as the reference pattern, and show a sequence of a thin stripe sweeping across the screen in vertical direction and then in horizontal direction [30, 31]. We capture an image for each of the positions of the sweeping stripe (see Fig. 2.3). For each image point, its correspondence on the reference pattern can then be solved by examining the sequence of intensity values of the image point for each sweeping direction and locating the peak intensity value. The position of the stripe that produces the peak intensity value in each sweeping direction then gives us the position of the correspondence on the reference pattern. In order to improve the accuracy of the peak localization, we fit a quadratic curve to the intensity profile in the neighborhood of the sampled peak value, and solve for the exact peak analytically.

### 2.3.4 Light Path Triangulation

Suppose high quality correspondences have been established between the images and the reference pattern under each of the two distinct positions and in each of the two media (i.e., with and without liquid in the tank). We can reconstruct two PBCs for each image point using the calibrated positions of the reference pattern. The FEP can then be recovered as the point of intersection between the two PBCs. Below we derive a simple solution for the FEP based on the established correspondences of an image point.

Consider an image point  $q$  (see Fig. 2.4). Suppose  $\mathbf{M}_0$  and  $\mathbf{M}_1$  denote, respectively, its correspondences on the reference pattern under position 0 and position 1 with liquid

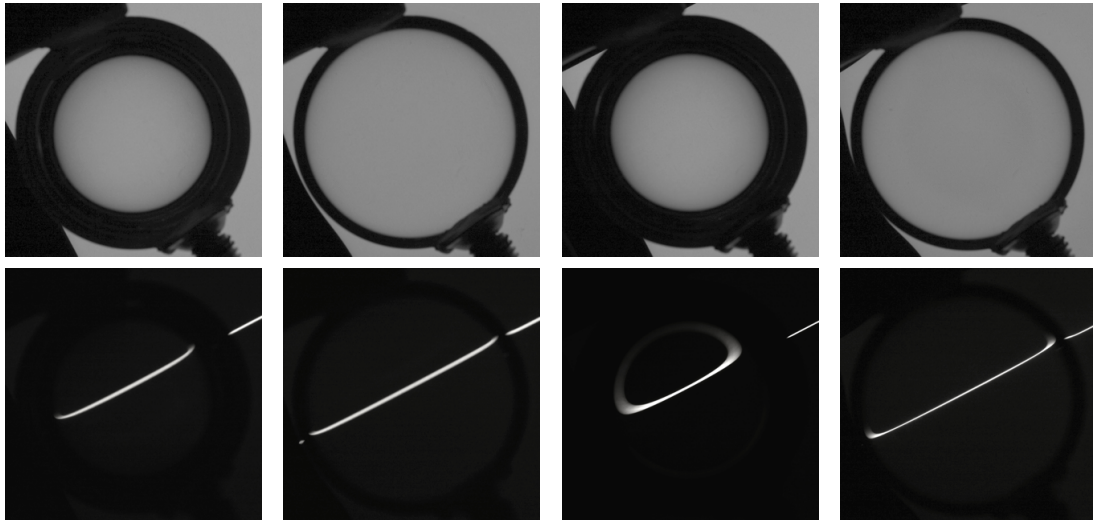


Fig. 2.3 The upper row shows images of a transparent hemisphere captured in front of a gray background (from left to right: reference pattern at a high position and without water, reference pattern at a high position and with water, reference pattern at a low position and without water, and reference pattern at a low position and with water). The lower row shows images of the hemisphere captured in front of a sweeping stripe (in the same order).

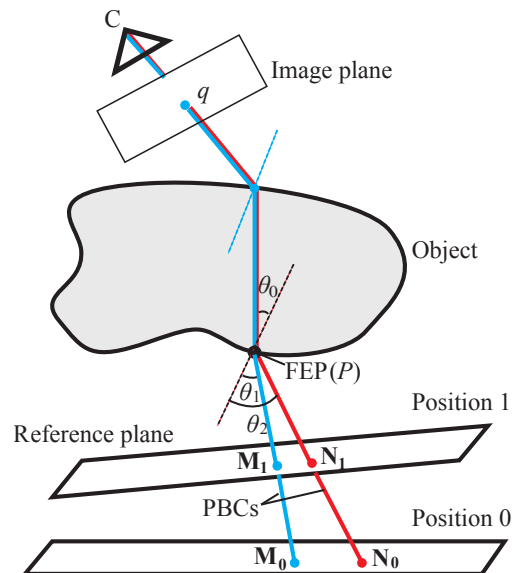


Fig. 2.4 PBC reconstruction and FEP estimation. The correspondences of an image point  $q$  on the reference pattern under position 0 and position 1 define a PBC. Given two PBCs in two different media, the FEP for  $q$  can be obtained by triangulating the PBCs.

in the tank. Similarly, let  $\mathbf{N}_0$  and  $\mathbf{N}_1$  denote, respectively, its correspondences on the reference pattern under position 0 and position 1 without liquid in the tank.  $\mathbf{M}_0, \mathbf{M}_1, \mathbf{N}_0$  and  $\mathbf{N}_1$  are in  $\mathbb{R}^3$ . The PBCs for  $q$  in liquid and in air can be expressed as

$$L_M : \mathbf{M}(s) = \mathbf{M}_0 + s\mathbf{U}, \quad (2.1)$$

$$L_N : \mathbf{N}(t) = \mathbf{N}_0 + t\mathbf{V}, \quad (2.2)$$

where  $\mathbf{U} = \frac{\mathbf{M}_1 - \mathbf{M}_0}{\|\mathbf{M}_1 - \mathbf{M}_0\|}$  and  $\mathbf{V} = \frac{\mathbf{N}_1 - \mathbf{N}_0}{\|\mathbf{N}_1 - \mathbf{N}_0\|}$ . Under a perfect situation, the FEP for  $q$  is given by the point of intersection between  $L_M$  and  $L_N$ .

Due to noise, however,  $L_M$  and  $L_N$  often may not intersect with each other exactly at a point. In this situation, we seek the point  $\mathbf{M}_c = \mathbf{M}(s_c)$  on  $L_M$  and the point  $\mathbf{N}_c = \mathbf{N}(t_c)$  on  $L_N$  such that the Euclidean distance between  $\mathbf{M}_c$  and  $\mathbf{N}_c$  is a minimum. The distance between  $\mathbf{M}_c$  and  $\mathbf{N}_c$  can be taken as a quality measure of the reconstruction. If the distance is below a specified threshold, the mid-point between  $\mathbf{M}_c$  and  $\mathbf{N}_c$  can be taken as the FEP for  $q$ . Note that if  $\mathbf{U}$  and  $\mathbf{V}$  are parallel, there will not be a unique solution. This corresponds to the case where the two PBCs overlap with each other. This is a degenerate case which happens only when the incident ray is parallel to the surface normal.

### 2.3.5 Surface Normal Reconstruction

Recall that for the purpose of surface reconstruction, neither the refractive index of the object nor that of the liquid is needed. If, however, the refractive index of the liquid is known a priori, it is possible to recover the surface normal at each FEP. Let  $\theta_1$  and  $\theta_2$  denote the incident angles of the PBCs in the liquid and air, respectively, at the surface point  $P$ , and  $\theta_0$  denote the refracted angle (see Fig. 2.4). Suppose the refractive index of the object, liquid and air are given by  $\lambda_0$ ,  $\lambda_1$  and  $\lambda_2$ , respectively. By Snell's Law, we have

$$\lambda_0 \sin \theta_0 = \lambda_1 \sin \theta_1 = \lambda_2 \sin \theta_2. \quad (2.3)$$

Let  $\Delta\theta = \cos^{-1}(\mathbf{U} \cdot \mathbf{V})$  denote the angle between the two PBCs. Substituting this into (2.3) gives

$$\lambda_1 \sin \theta_1 = \lambda_2 \sin(\theta_1 + \Delta\theta). \quad (2.4)$$

With known refractive indices  $\lambda_1$  and  $\lambda_2$  for the liquid and air, respectively, the incident angle  $\theta_1$  can be recovered by

$$\theta_1 = \tan^{-1} \left( \frac{\lambda_2 \sin \Delta\theta}{\lambda_1 - \lambda_2 \cos \Delta\theta} \right). \quad (2.5)$$

The surface normal  $\mathbf{n}_p$  at  $P$  is then given by

$$\mathbf{n}_p = \mathbf{R}(\theta_1, \mathbf{V} \times \mathbf{U})\mathbf{U}, \quad (2.6)$$

where  $\mathbf{R}(\theta, \mathbf{a})$  denotes a Rodrigues rotation matrix for rotating about the axis  $\mathbf{a}$  by the angle  $\theta$ .

## 2.4 Recovery of Thin Transparent Objects

The method proposed in Section 2.3 requires immersing an object partially into a liquid. However, this is not an easy task for flat thin transparent objects (e.g., glass plates, thin lens). In this section, we show that for thin transparent objects, the requirement of immersing the object partially into a liquid can be removed by a single refraction approximation, resulting in a simplified setup.

### 2.4.1 Setup and Assumptions

We follow the same notations used in Section 2.3. As discussed in [38], it is generally true for thin transparent objects to assume only one refraction occurs along each light path passing through the object. In general, a light path originates from the reference plane will be refracted (at least) twice at the surface of the object before it reaches the camera. However, if the object is very thin, the light path segment inside the object



becomes negligible. In this case, we can assume only one single refraction along each light path. To reconstruct a PBC and a visual ray for each image point, the reference plane is placed at two distinct positions. For each position, two images of the reference pattern are captured, one with the object between the camera and the reference plane and the other without the object. After calibrating the positions of the reference plane and establishing correspondences between the image and the reference plane, we can reconstruct a PBC and a visual ray for each surface point (see Fig. 2.5). The visual ray can be reconstructed from the direct view of the pattern (red path in Fig. 2.5)<sup>1</sup>, and the PBC can be reconstructed from the refraction of the pattern caused by the thin surface (blue path in Fig. 2.5).

### 2.4.2 Surface Reconstruction

Assuming a single refraction occurs along each light path passing through a thin transparent object, the FEP can be recovered by triangulating the visual ray and the PBC of each image point. Compared with the general setup discussed in Section 2.3, the requirement of immersing the object partially into the liquid to alter the incident rays is removed along with the need for a water tank. However, the baseline between these two rays is quite narrow for a thin surface. It leads to noisy FEP cloud estimation. With a known refractive index of the object, the surface normal can be recovered using the method introduced in Section 2.3. In particular, we only need to replace  $\lambda_1$  (refractive index of the liquid) by  $\lambda_0$  (refractive index of the object) in (2.5). The surface normal can then be obtained by (2.6). We therefore reconstruct the surface by integrating surface normals estimated from these rays, which proves to be more robust to noise.

---

<sup>1</sup>If the camera is calibrated w.r.t the reference plane, it is straightforward to recover the visual ray of an image point, and two images are sufficient to construct the blue PBC. By using four images as described in the main text, the PBC and visual ray can be constructed even without calibrating the camera. We only need to calibrate the pattern poses, which is also required by the two-image method.

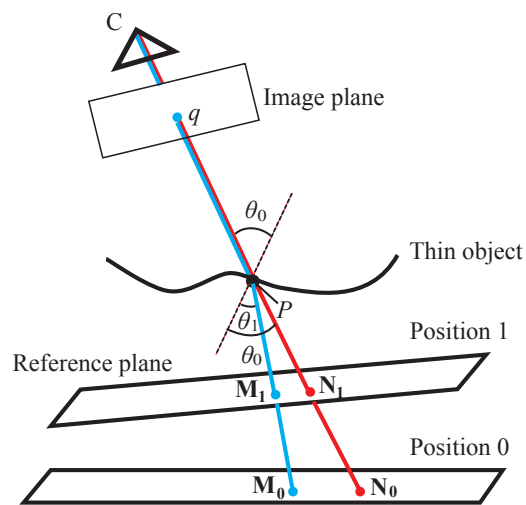


Fig. 2.5 PBC and visual ray construction and surface estimation for thin objects. The correspondences of an image point  $q$  on the reference pattern under position 0 and position 1 define a visual ray (red) and a PBC (blue). The visual ray and PBC are constructed from the direct view of the pattern and the refraction of the pattern caused by the object, respectively. Given these two rays, the surface point  $P$  for  $q$  can be obtained by ray triangulation, meanwhile the normal for  $P$  can also be recovered using method described in Section 2.3.5.

## 2.5 Discussions

### 2.5.1 Total Internal Reflection

It is well known that total internal reflection will occur if a light ray propagates from one medium with a larger refractive index to another medium with a smaller refractive index (e.g., from glass to air), but not for the opposite propagation direction (e.g., from air to glass). In the scenario of transparent surface reconstruction, as the refractive index of a solid object is generally larger than that of its surrounding environment (either air or liquid), total internal reflection will inevitably happen. Here, we discuss the potential total internal reflection situation when adopting our approach.

Consider a light path traveling from one medium with a refractive index  $\lambda_1$  to another medium with a refractive index  $\lambda_2$ , where  $\lambda_1 > \lambda_2$ . Total internal reflection only happens when the incident angle is greater than the critical angle  $\theta_c = \sin^{-1}(\frac{\lambda_2}{\lambda_1})$ . Fig. 2.6 depicts an example of total internal reflection, where  $\theta_v < \theta_c$  and  $\theta_w > \theta_c$ . When total internal reflection happens, the light path may not reach the pattern and the refraction correspondences cannot be established. On the other hand, if after total internal reflection at **W**, the light ray continues to propagate to another surface point **X**, and refracts at **X**, and eventually reaches the pattern, our approach can still handle this case.

In practice, total internal reflection does not frequently happen, as the critical angle is normally very large (e.g.,  $\theta_c = 41.8^\circ$  for glass to air). Only specially designed objects, like diamonds, will purposely make total internal reflection happen.

### 2.5.2 Object Analysis

Our general method only has the assumption that light paths (propagating from the pattern to the camera center) will not re-enter the liquid used for immersing the object once they enter the object. This assumption holds true for transparent objects with a convex shape, and for objects with holes completely enclosed inside the object. It allows us to handle object with an inhomogeneous refractive index. In practice,

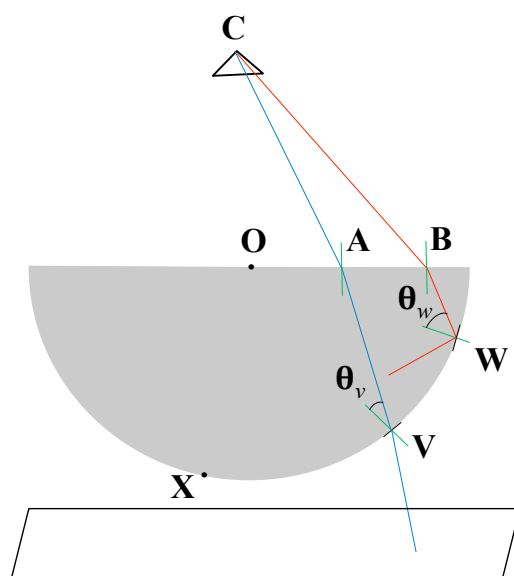


Fig. 2.6 A total internal reflection example. A camera centered at  $C$  observes a transparent hemisphere centered at  $O$  and a reference plane is placed below the object. The blue light path traveling along  $C$ ,  $A$  and  $V$  has two refractions at  $A$  and  $V$  respectively. While the red light path traveling along  $C$ ,  $B$  and  $W$  first refracts at  $B$  and has a total internal reflection at  $W$ .

our method can also handle objects with shallow concavities as long as the previous assumption is satisfied.

Our thin object reconstruction method has the assumption of a single refraction. This is generally true for most thin transparent surfaces. The exception happens when the back and front sides of the surface are planar and parallel. For such surfaces (see Fig. 2.7), the normals at the first and second refraction points are parallel and in opposite direction. In this case, the visual ray and PBC are parallel. Our thin transparent object reconstruction method cannot handle this case.

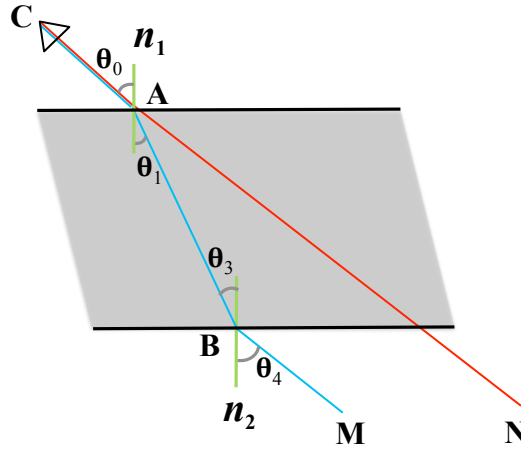


Fig. 2.7 Refractions at the surfaces of a parallel planar plate. A camera centered at **C** observes a parallel planar surface. **A** is an upper surface point and **B** is a lower surface point along a light path. As their normals  $\mathbf{n}_1$  and  $\mathbf{n}_2$  are parallel, the PBC (**BM**) is parallel to the visual ray (**CA**). The angle between these two rays is  $\Delta\theta = 0$ .

### 2.5.3 Single Refraction Approximation

Here we analyze the error induced by the single refraction approximation used in our second method, and demonstrate that such an approximation is appropriate for thin transparent objects.

Referring to Fig. 2.8, we have a camera centered at **C** observing a thin transparent object in front of a reference pattern. Let us consider the light path through an arbitrary image point  $q$ , and traverse this light path in reverse direction (i.e., beginning

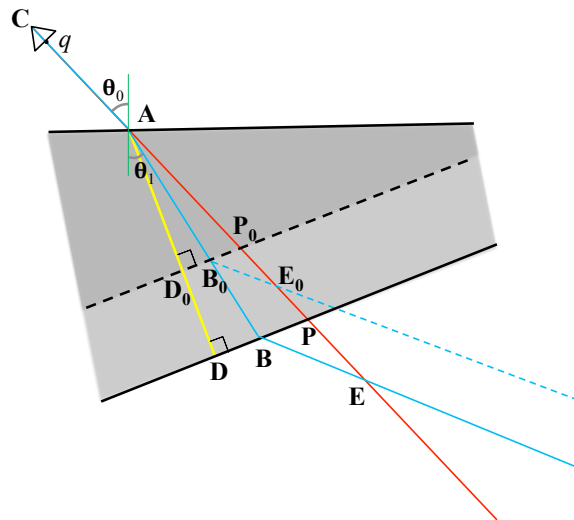


Fig. 2.8 Error induced by the single refraction approximation. It can be shown that the approximation error (i.e., distance between **B** and **E**) is linearly proportional to the thickness of the transparent object (i.e., distance between **A** and **D**). Please refer to the main text for details.

from the optical center of the camera, travelling through the thin transparent object, and eventually terminating at the reference pattern). After leaving the camera, this light path first refracts at point **A** on the upper surface of the transparent object. It continues to travel through the interior of the object, and refracts at point **B** on the lower surface of the object. After leaving the object, it continues to travel through the air and eventually terminates at a point on the reference pattern. If we apply the single refraction approximation, we will obtain point **E** from the intersection of the PBC and the visual ray of  $q$ . Note that the FEP for  $q$  should be point **B**, and therefore the distance between **B** and **E** is the approximation error.

Suppose we make the thin transparent object even thinner by moving its lower surface along its surface normal towards its upper surface, resulting in the new lower surface represented by the dotted line in Fig. 2.8. After leaving the camera, the light path for  $q$  again first refracts at point **A** on the upper surface of the transparent object. It continues to travel through the interior of the object, but this time refracts at point **B<sub>0</sub>** on the lower surface of the thinner object. After leaving the object, it continues to

travel through the air and eventually terminates at a point on the reference pattern. If we apply the single refraction approximation again, we will obtain point  $\mathbf{E}_0$  from the intersection of the PBC and the visual ray of  $q$ . Similarly, the distance between  $\mathbf{B}_0$  and  $\mathbf{E}_0$  is the approximation error for this thinner object.

Consider the similar triangles  $\triangle \mathbf{AB}_0\mathbf{P}_0$  and  $\triangle \mathbf{ABP}$ . It is easy to see from Fig. 2.8 that  $\|\mathbf{AB}_0\| : \|\mathbf{AB}\| = \|\mathbf{AD}_0\| : \|\mathbf{AD}\|$ . Hence, we have  $\|\mathbf{B}_0\mathbf{P}_0\| : \|\mathbf{BP}\| = \|\mathbf{AB}_0\| : \|\mathbf{AB}\| = \|\mathbf{AD}_0\| : \|\mathbf{AD}\|$ . Consider now the similar triangles  $\triangle \mathbf{B}_0\mathbf{P}_0\mathbf{E}_0$  and  $\triangle \mathbf{BPE}$ . We have  $\|\mathbf{B}_0\mathbf{E}_0\| : \|\mathbf{BE}\| = \|\mathbf{B}_0\mathbf{P}_0\| : \|\mathbf{BP}\| = \|\mathbf{AD}_0\| : \|\mathbf{AD}\|$ . Hence, we can conclude that the approximation error is linearly proportional to the thickness of the transparent object, and therefore the single refraction approximation is appropriate for thin transparent objects.

It is also worth to note that the difference between the surface normals at the upper and lower surface points along the same light path will also affect the reconstruction accuracy under the single refraction approximation. If we increase the difference between the surface normals at  $\mathbf{A}$  and  $\mathbf{B}$  by rotating the lower surface around  $\mathbf{B}$ ,  $\mathbf{E}$  will get closer to  $\mathbf{B}$ , which indicates a smaller reconstruction error.

### 2.5.4 Limitations

Our first method assumes that light paths will not re-enter the object once they exit the object. However, there do exist some complex transparent objects that break this assumption, such as objects with multiple holes which are not fully enclosed inside the objects. Our first method cannot handle such objects. Besides, if the incident light ray is parallel to the surface normal, our method cannot reconstruct the FEP. Our first method can only reconstruct one side of the object at a time. When a complete 3D model of the object is required, we need to rotate the object to reconstruct it part by part. Our second method adopts a single refraction approximation for thin transparent objects. This approximation can simplify our setup, but it will also introduce some errors as discussed above. Therefore, it cannot be applied to highly accurate surface reconstruction. When the back and front sides of the surface are planar and parallel,

the single refraction approximation cannot be applied.

## 2.6 Experimental Evaluation

We now demonstrate the effectiveness of our approach on synthetic and real objects. In the remainder of this section, we present both quantitative and qualitative reconstruction results. In the following, for the sake of clarity, we denote our general method that uses liquid to alter the incident light path as the first method, and the method that uses the object itself to alter the incident light path, which is tailored for thin transparent objects, as the second method.

### 2.6.1 Synthetic Data

**First method on a convex object.** For our synthetic experiments, we used *Pov-Ray* to simulate the entire experimental setup. First, we modeled a convex transparent object as a semi-ellipsoid with the following parametric equation

$$\begin{cases} \left(\frac{x}{12.5}\right)^2 + \left(\frac{y}{12.5}\right)^2 + \left(\frac{z}{5}\right)^2 = 1, \\ z > 0. \end{cases} \quad (2.7)$$

We further assumed the transparent ellipsoid has a refractive index  $\lambda = 1.5$ .<sup>2</sup> A reference plane displaying a set of thin stripe sweeping patterns was placed at two different positions. The size of the reference plane was  $32 \times 32$  units in *Pov-Ray* environment and the thickness of the stripe was  $\frac{1}{32}$  unit. A synthetic perspective camera with a resolution of  $1024 \times 1024$  was used to capture the refraction of the reference pattern through the transparent object immersed in air ( $\lambda = 1.0$ ) and liquid ( $\lambda = 1.3$ ) respectively. We adopted the strategies described in Section 2.3.3 to obtain dense refraction correspondences. More than 700K refraction correspondences were used in our synthetic experiment.

---

<sup>2</sup>The transparent object can be inhomogeneous, namely the refractive index varies across the interior of the object.



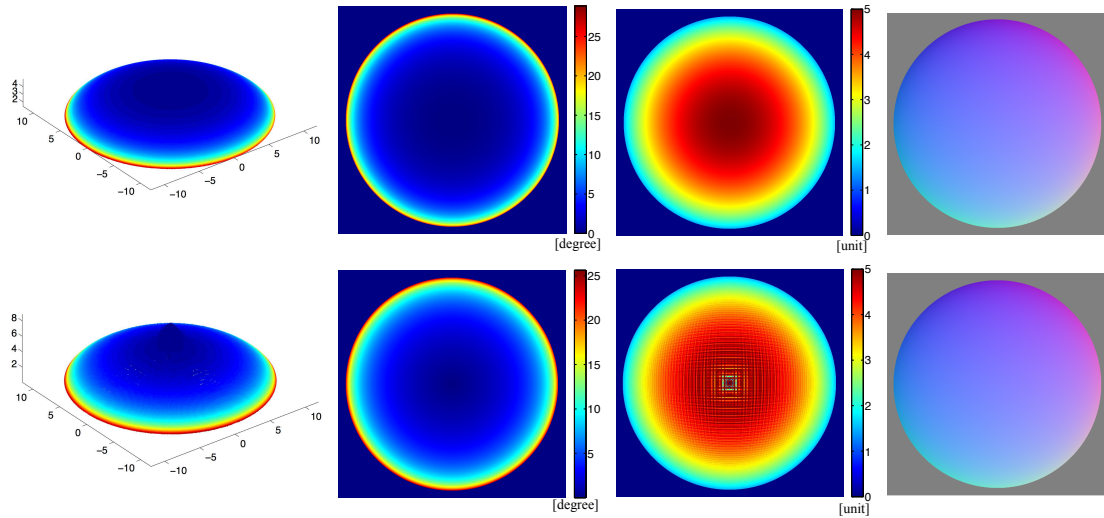


Fig. 2.9 Reconstruction of a synthetic semi-ellipsoid using the first method. First row: Ground truth. Second row: Our reconstructed results. First column: FEP cloud. Second column: angle between the PBCs in a pair. Third column: depth map. Fourth column: normal map.

We reconstructed a pair of PBCs for each FEP based on the retrieved refraction correspondences. The transparent surface was then recovered from the ray triangulation of PBC pairs. We also computed the surface normals from the PBC pairs and the refractive indices of the media. Fig. 2.9 depicts the reconstructed FEP cloud as well as surface normals. It also shows the depth map of the reconstructed object for accuracy evaluation<sup>3</sup>.

In practice, reconstruction errors originate from the inaccuracy in finding the refraction correspondences on the reference patterns. Errors may increase as the relative distance between the two positions of the reference pattern decreases. We therefore carried out a joint analysis by adding 2D zero-mean Gaussian noise to the extracted dense correspondences on the reference pattern together with varying the relative distance between the two positions of the reference pattern. The noise level ranged from 0.1 to 1.0 unit. The relative distance between the positions of the reference pattern varied from 5 to 20 units. We fixed the first position of the reference pattern at  $z = 10$  in our experiment, and varied the second position of the pattern by placing it

<sup>3</sup>The depth map is defined as the  $z$  component for each 3D point.

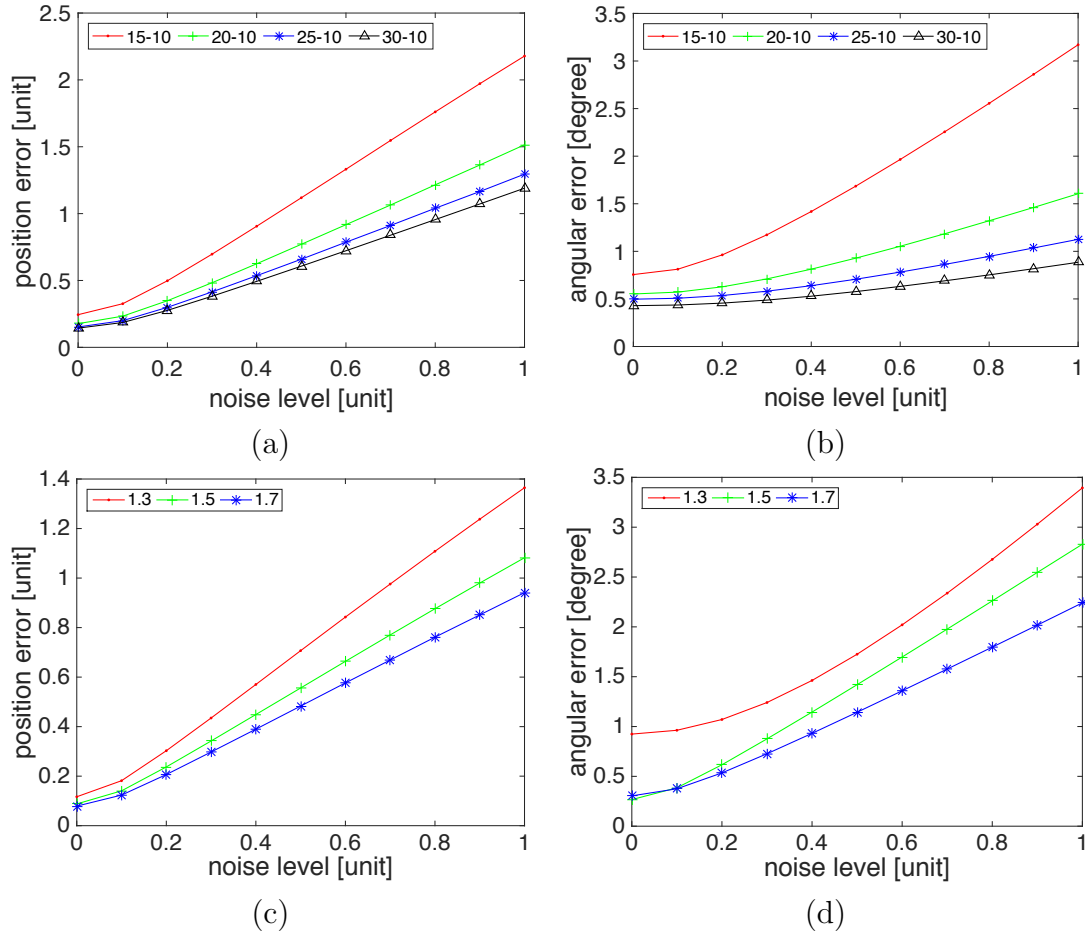


Fig. 2.10 RMS errors for the positions and normals of the FEPs reconstructed using the first method. (a)-(b) show the RMS errors for the positions and normals, respectively, over 500 rounds with different random noise and relative distances between the two positions of the reference pattern. In particular, we fixed the first position of the reference pattern at  $z = 10$  in our experiment, and varied the second position of the pattern by placing it at  $z = 15, 20, 25, 30$ , respectively. (c)-(d) show the RMS errors for the positions and normals, respectively, over 500 rounds with different random noise and refractive indices of the media. In particular, we tested three refractive indices ( $\lambda = 1.3, 1.5, 1.7$ ).

at  $z = 15, 20, 25, 30$ , respectively. The reconstruction accuracy was evaluated based on the root mean square (RMS) error between the ground truth surface and the reconstruction. We further computed the angular distances between our reconstructed normals and the ground truth normals computed from the analytical equation of the semi-ellipsoid. Fig. 2.10(a-b) show the RMS errors for the positions and normals of the reconstructed FEPs under different noise level and relative distances between the two positions of the reference pattern. It shows that the reconstruction errors decrease as the distance between the two positions of the reference pattern increases.

We further conducted an analysis on the reconstruction error with respect to the refractive index of the liquid medium. Two other media with different refractive indices were tested in the experiment, namely  $\lambda = 1.5$  and  $\lambda = 1.7$ . The reference pattern was placed at  $z = 6$  and  $z = 10$ , respectively. Fig. 2.10(c-d) show that the reconstruction errors decrease as the refractive index of the medium increases.

**First method on a concave object.** We also evaluated our method on a concave surface with  $\lambda = 1.7$ , which was defined by the difference between a cylinder and a right circular cone. The radius and height of the cylinder were 5 and 10 respectively, while the height and radius of the cone were 4 and 10 respectively. The resulting shape was an object with a cylinder outer shape and a right circular cone inner shape. We reconstructed the inner shape of the object in this experiment by immersing the concave side of the object into water. Fig. 2.11 summarizes the results. The RMS error for FEP was 0.141 unit, and the RMS error for normal was  $1.58^\circ$ . It shows that our method can successfully reconstruct object with concavity as long as the light rays do not re-enter the object once they exit the object.

**Second method on a thin convex cone.** Under the same synthetic environment as described above, we rendered a thin object to evaluate our approach in Section 2.4. We evaluated our thin transparent object reconstruction method on a right circular cone with  $\lambda = 1.7$ . Its height and radius were 1 and 4 respectively.

The reference plane was placed at  $z = 20$  and  $z = 30$  respectively. A synthetic perspective camera with a resolution of  $1024 \times 1024$  was used to capture the image of

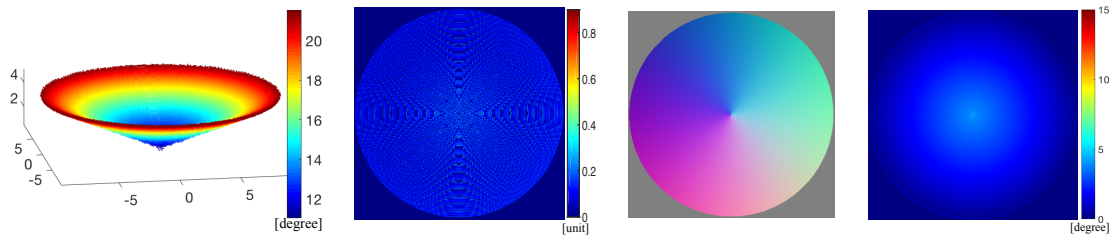


Fig. 2.11 Reconstruction of a concave transparent object using the first method. From left to right: the reconstructed FEP cloud (color coded by angle between the PBCs in a pair); error map between ground truth FEPs and reconstructed FEPs; reconstructed normal map; error map between ground truth normals and reconstructed normals.

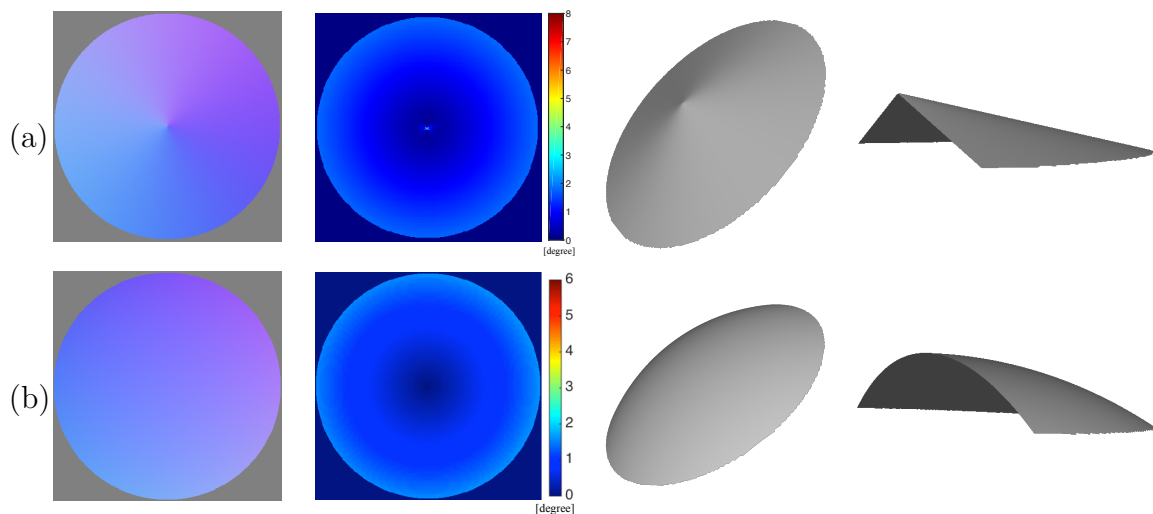


Fig. 2.12 Reconstruction of two synthetic thin objects using the second method: (a) thin cone; (b) spherical shell. First column: reconstructed normal map. Second column: error map between ground truth normals and reconstructed normals. Last two columns: two views of the reconstructed surface (the second view is a cross-section view).

the pattern directly and through the thin object, respectively. After we established four correspondences for each pixel, a PBC and a visual ray were reconstructed accordingly. The surface normals were then recovered using our method described in Section 2.4. Other than the correspondence quality, the distance of the two positions of the pattern and the refractive index as discussed above, the accuracy of our thin transparent object reconstruction method will also be affected by the thickness of the objects. Hence we carried out a joint analysis by adding  $2D$  zero-mean Gaussian noise to the extracted dense correspondences on the reference pattern together with varying thickness of the object. The noise level ranged from 0.1 to 1.0 units. The varying of the thickness was achieved by padding the cone with a cylinder of the same radius and setting the height of the cylinder to  $h = 0.0, 0.5, 1.0, 2.0$ , respectively. Fig. 2.12(a) shows the reconstruction result for  $h = 0$ , i.e., the cone without padding the cylinder. The errors induced by the single refraction approximation were small ( $< 2^\circ$ ). The joint analysis results are presented in Fig. 2.13. For a fixed thickness, with an increase of noise level, the RMS error of the estimated surface normals does not change a lot. This demonstrates the robustness of our approach. It can also be seen that the errors decrease with thickness of the object.

**Second method on a spherical shell.** We constructed a thin transparent spherical shell with  $\lambda = 1.7$  by subtracting a solid transparent sphere defined by (2.8) from another solid transparent sphere defined by (2.9).

$$\left(\frac{x}{10}\right)^2 + \left(\frac{y}{10}\right)^2 + \left(\frac{z-s}{10}\right)^2 = 1, s \in \{1, 2, 3, 4, 5\} \quad (2.8)$$

$$\left(\frac{x}{10}\right)^2 + \left(\frac{y}{10}\right)^2 + \left(\frac{z}{10}\right)^2 = 1 \quad (2.9)$$

Similar joint analysis as before was also conducted for this object. The varying of the thickness was achieved by setting different  $s$  value in (2.8), which specifies the distance between the two sphere centers. Fig. 2.12(b) shows the reconstruction result for  $s = 3$ . Fig. 2.14 depicts the results of the joint analysis. For this object, the error

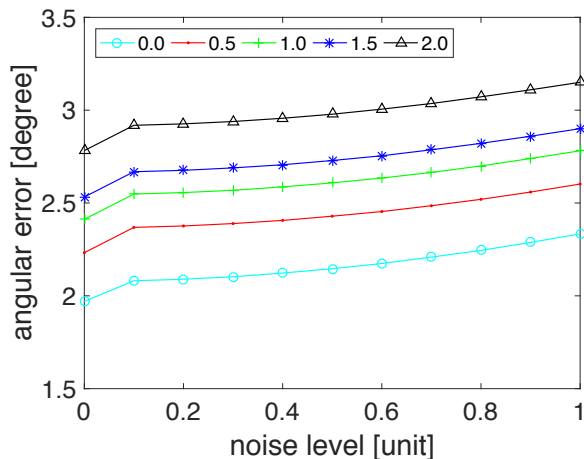


Fig. 2.13 Error of surface normals recovered using the second method for a synthetic thin convex cone under different thicknesses. A cylinder was padded to the cone to change its thickness. The thickness values of the padded cylinder are shown in the legend.

does not keep decreasing with its thickness. The error decreases with the thickness at first, but then it starts to increase after some particular thickness (e.g., 3 units in our experiment). For the spherical shell, with the decrease of its thickness, the difference between the normals at the upper and lower surface points will also decrease. When the object gets too thin, the normals at the upper and lower surface points along each light path tend to become parallel. In this case, the single refraction assumption is no longer applicable. In contrast, for the thin convex cone, the normals at the upper and lower surface points along each light path keep the same with the change of the thickness of the object, and the errors decrease with thickness of the object (see Fig. 2.13).

## 2.6.2 Real Data

To evaluate the accuracy of our **first method on real data**, we performed experiments on a smooth glass *hemisphere*, a diamond-shape *ornament* with piecewise planar surfaces (see Fig. 2.1), and a small *bottle* (see Fig. 2.17). We acquired images with a *Canon EOS 40D* camera equipped with a 24 *mm* lens and used a 9.7-*inch* iPad

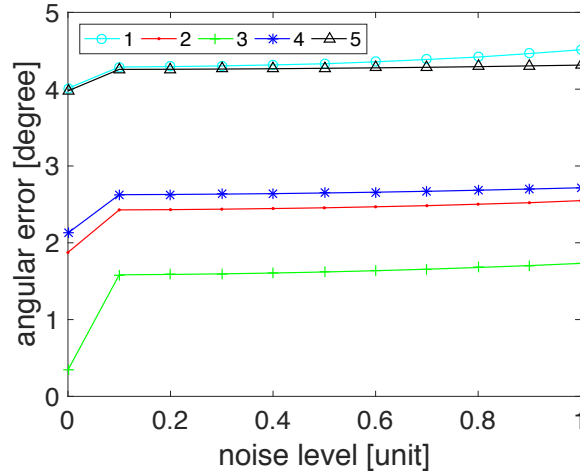


Fig. 2.14 Error of surface normals recovered using the second method for a synthetic thin spherical shell under different thicknesses. The thickness values are shown in the legend, which are the distances between the two sphere centers defined by  $s$  in (2.8).

with a resolution of  $2048 \times 1536$  as the reference plane. We displayed stripe patterns on the iPad for extracting the dense refraction correspondences using the strategy in Section 2.3.3. In order to reconstruct PBCs, the reference plane was placed at two different positions in a water tank. Under each position, we first took one set of images of the sweeping stripe patterns refracted by the object directly. We then filled the tank with water, having a refractive index  $\lambda = 1.33$ , to alter the PBCs and took another set of images. In brief, four sets of images with a resolution of  $3888 \times 2592$  were captured for each object. This yielded dense correspondences (see Table 2.1). The poses of the reference plane relative to the camera were calibrated with Matlab Calibration Toolbox [54].

A pair of PBCs were reconstructed from the extracted refraction correspondences for each image point. These PBCs were triangulated to give an estimate of the FEP. We treated those reconstructed FEPs with a small PBC angle ( $< 1^\circ$ ), or out of the depth range between the camera and reference plane as noise points. The normal for each FEP was then recovered with the knowledge of refractive indices 1.0 and 1.33 for air and water respectively. In Fig. 2.15, we show our reconstructed 3D FEP cloud, angles between the PBCs in a pair, depth map, and surface normals for *hemisphere*

Table 2.1 Statistics for our real experiment with the first method. We show the number of captured encoding pattern images, refraction correspondences, reconstructed FEPs, and reconstructed normals for our dense reconstructions of *hemisphere*, *ornament* and *bottle*, respectively.

	<i>hemisphere</i>	<i>ornament</i>	<i>bottle</i>
Images	2,800	2,200	2,100
Corres.	1,180,300	546,173	483,052
FEPs	1,115,748	519,162	471,874
Normals	1,115,748	519,162	471,874

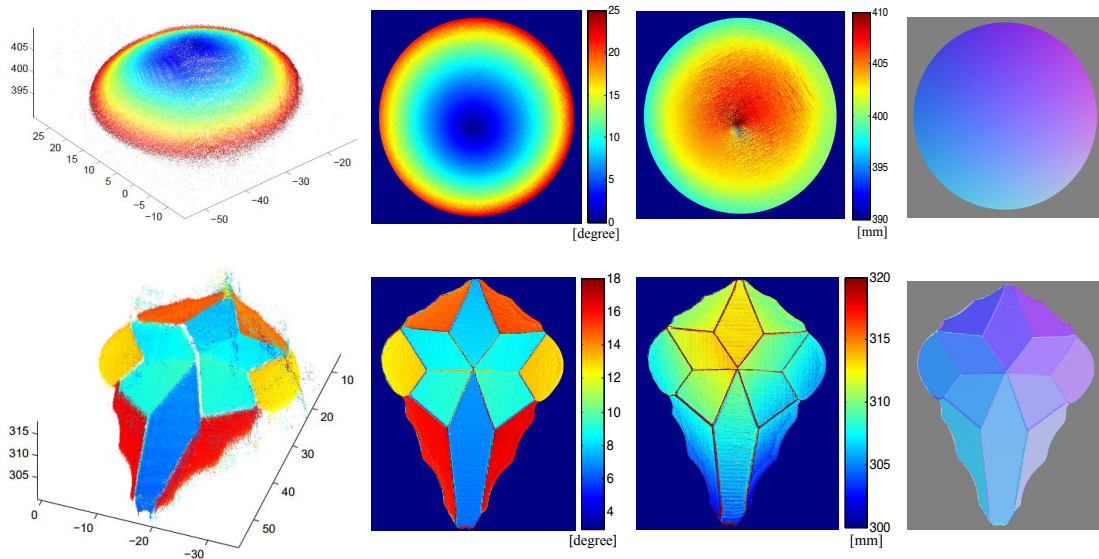


Fig. 2.15 Reconstruction results of the first method on real data. First row: *hemisphere* reconstruction results. Second row: *ornament* reconstruction results. The first column shows the reconstructed FEPs; the second column shows the angle between each PBC pairs; the third column shows the depth map; the fourth column shows the reconstructed normal map.



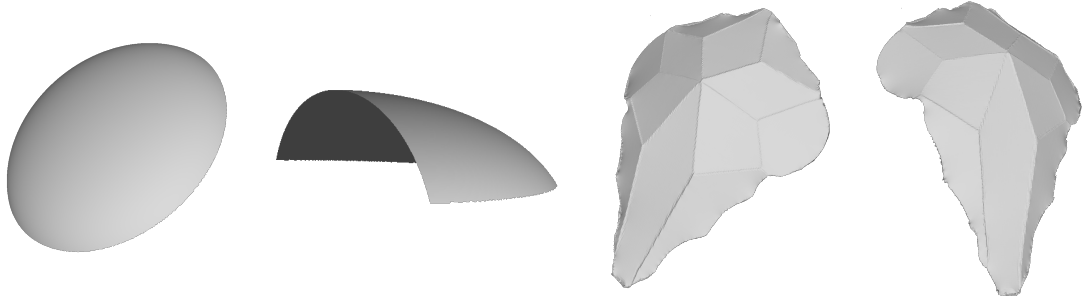


Fig. 2.16 Two views of the reconstructed surface using the first method. First two columns: *hemisphere* reconstruction results (the second view is a cross-section view). Last two columns: *ornament* reconstruction results. Note the shown surfaces are the surfaces touching the water in the experiment.

and *ornament* respectively. Note that large reconstruction errors occur in regions with small PBC angles. We also employed the integration method proposed by Xie *et al.* in [56] to generate surface meshes with our recovered normals. These meshes are shown in Fig. 2.16.

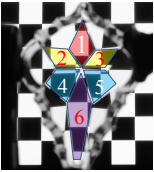
Since no ground truth was available, a sphere was fitted to the FEP cloud to evaluate the reconstruction accuracy for the *hemisphere*. We compared the fitted radius with the physical measurement, which were  $26.95\text{ mm}$  and  $27.99\text{ mm}$ , respectively. The error was  $1.04\text{ mm}$  (i.e., 3.7% compared with the measurement). Table 2.2 shows the reconstruction errors of the *hemisphere* compared against the fitted sphere. The mean and median FEP position errors were  $< 0.6\text{ mm}$ , and the mean and median normal errors were  $< 7.0^\circ$ . This shows a high accuracy of the reconstruction. In order to evaluate the reconstruction of the *ornament*, we first used RANSAC [57] to fit a plane for each facet. The reconstruction error for each facet was measured by the distances from the reconstructed FEPs to the fitted plane, as well as the angles between the reconstructed normals and the normal of the fitted plane. The results shown in Table 2.3 suggest that our proposed approach can accurately reconstruct the piecewise planar *ornament*. The mean and median FEP position errors were  $< 1.0\text{ mm}$ <sup>4</sup> and the mean and median normal errors were  $< 10.0^\circ$ .

<sup>4</sup>Except facet 6 with a mean of 1.0442.

Table 2.2 Reconstruction errors of *hemisphere* using the first method. The position error is defined as the difference between the distance from the fitted center to each FEP and the length of fitted radius. The normal error is defined as the angle between the ray from the fitted center to each FEP and the reconstructed normal for each FEP.

	Position (mm)	Normal (degree)
Mean error	0.5903	6.9665
Median error	0.4179	6.9215

Table 2.3 Reconstruction error measurements of *ornament*. Left figure: shows the labels for each facet of the ornament. Right table: shows the various error metric used in reconstruction error evaluation for *ornament*. Due to its piecewise property, we fitted each facet using RANSAC with an inlier threshold of  $0.5mm$  and then measured the distances from the FEPs to the fitted plane and also the angle difference between the reconstructed normals of each facet region and the fitted facet normal.

	Facet label	1	2	3	4	5	6
	Mean normal error (degree)	6.2654	9.9585	7.5905	9.6871	3.6591	6.8677
	Median normal error (degree)	6.1677	9.5906	7.5109	9.7511	3.4741	6.7908
	Mean position error (mm)	0.7250	0.6814	0.6675	0.6767	0.5881	1.0442
	Median position error (mm)	0.6108	0.5945	0.5755	0.5721	0.5133	0.6333
	RANSAC position inliers (%)	40.33	42.97	44.06	43.38	49.02	40.64

Besides, we also validated our approach on a hollow object by reconstructing a small transparent bottle. The reconstruction result is presented in Fig. 2.17 and the reconstructed surface mesh is shown in Fig. 2.18. The measured height and radius were  $55.65\text{ mm}$  and  $14.18\text{ mm}$  respectively, and the body part (red box region in Fig. 2.17) of the bottle is  $44.73\text{ mm}$  in height. We fitted a cylinder to the reconstructed point cloud using MLESAC [58]. We set the point-to-cylinder distance threshold to  $0.5\text{ mm}$  during fitting, and 41.17% of reconstructed FEPs are inliers. The fitted height and radius were  $46.26\text{ mm}$  and  $15.32\text{ mm}$  respectively. Both the height error and radius error were  $< 2\text{ mm}$ . The position error and normal error are shown in Table 2.4.

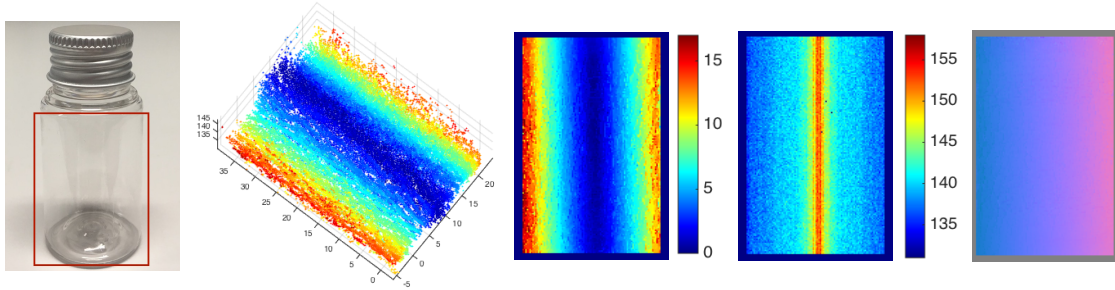


Fig. 2.17 Reconstruction result for *bottle* using the first method. Left to right: real object (red box highlights the region for reconstruction); reconstructed FEPs; angle between the PBCs in a pair; estimated depth map; reconstructed normal map.

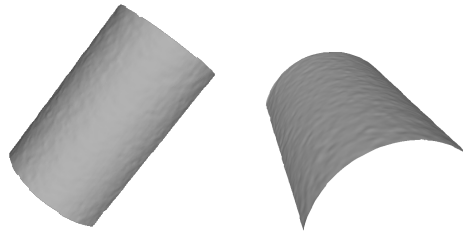


Fig. 2.18 Two views of the *bottle* reconstructed using the first method.

To evaluate our **second method on real data**, we applied our method on two thin glass plates, namely a *circular plate* and a *fish plate* (see the first column in Fig. 2.20). Since the objects were considered thin enough ( $\approx 0.3\text{ cm}$ ), compared with the size of the objects (*circular plate* : diameter =  $17.5\text{ cm}$ , *fish plate*:  $25.6\text{ cm} \times 20.7\text{ cm}$ ) and the distance between the camera and the objects ( $\approx 50\text{ cm}$ ), the light path

Table 2.4 Reconstruction errors of *bottle* using the first method. The position error is defined as the difference between the distance from the fitted cylinder axis to each FEP and the length of fitted radius. The normal error is defined as the angle between the normal computed analytically from the fitted cylinder and the reconstructed one for each FEP.

	Position (mm)	Normal (degree)
Mean error	0.6356	6.1256
Median error	0.6278	5.9183

displacements inside the objects could be ignored. Fig. 2.19 shows our real setup for thin object recovery. We used a 19-inch LCD display with a resolution of  $1280 \times 1024$  as the reference plane. Similar to the experiments done before, we captured four sets of images to establish refraction correspondences for each image point by arranging the display in two different positions. Differently, it was not necessary to immerse the thin object partially in a liquid any more. The simplified setup largely reduced the efforts in taking images.



Fig. 2.19 Real reconstruction setup for thin transparent objects.

We first captured an image sequence of the moving stripe on the reference plane, and then put the object in front of the camera to take another sequence. To take the third sequence, we moved the the reference plane to another position while keeping

the camera and object stationary. We then removed the object and took the last sequence. After reconstructing a PBC and visual ray for each observed surface point, the surface normals were recovered with the known refractive index of glass ( $\lambda = 1.52$ ). Due to the absence of the ground truth, we can only qualitatively evaluate our results. Fig. 2.20 shows the recovered normal map. The angles between rays in a pair are larger for regions with more details as these regions are less planar. The recovered normal maps were consistent with the real objects. The reconstructed surfaces are shown in Fig. 2.21, which can correctly show the details of the real objects.

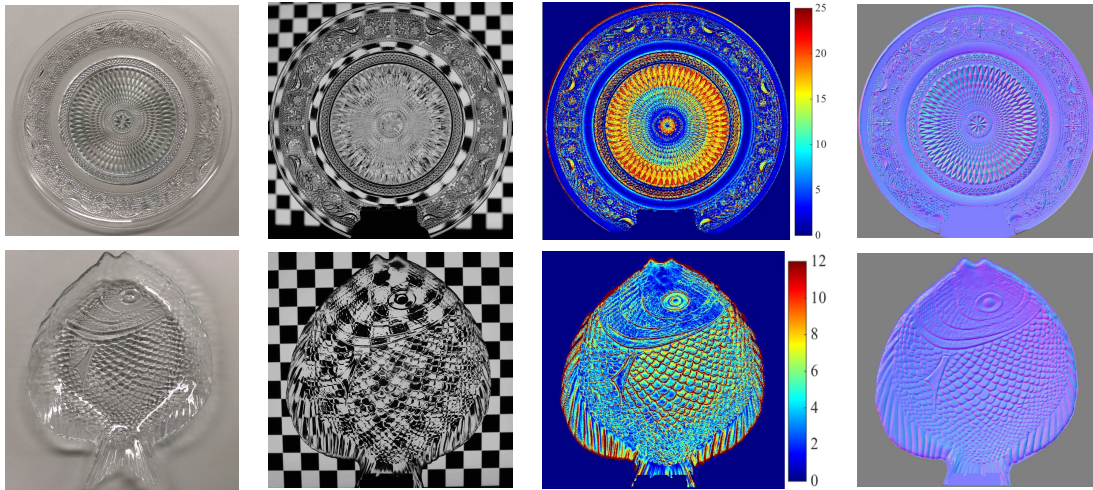


Fig. 2.20 Reconstruction results of the second method on real data. First row: *circular plate* reconstruction results. Second row: *fish plate* reconstruction results. The first column shows the real objects under room illumination; the second column shows the refraction of the pattern caused by the objects; the third column shows the angle between the visual ray and PBC in a pair; the last column shows the reconstructed normal map.

## 2.7 Conclusions

In this chapter, we develop a fixed viewpoint approach to dense surface reconstruction of transparent objects. We introduce a simple setup that allows us to alter the incident light paths by immersing the object partially in a liquid, while keeping the rest of the light paths fixed as light rays travel through the object. This greatly simplifies the

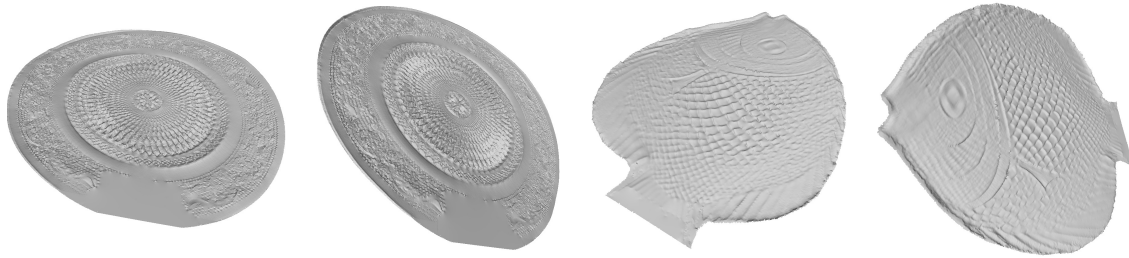


Fig. 2.21 Two views of the surface reconstructed using the second method. First two columns: *circular plate* reconstruction results. Last two columns: *fish plate* reconstruction results.

problem by making it not necessary to model the complex interactions of light inside the object, and allows the object surface to be recovered by triangulating the incident light paths. Our approach can handle transparent objects with a relatively complex structure, with an unknown and inhomogeneous refractive index. The only assumption to the objects is that the light paths should not re-enter the liquid medium once they enter the object. If the refractive index of the liquid is known a priori, our method can also recover the surface normal at each reconstructed surface point. Besides, for thin transparent objects, we show that the acquisition setup can be simplified by adopting a single refraction approximation. Experimental results demonstrate both the feasibility and robustness of our methods.

# Chapter 3

## Single View Mirror Surface Reconstruction

### 3.1 Introduction

3D reconstruction of *diffuse* surfaces has enjoyed tremendous success. Diffuse surfaces reflect light from a single incident ray to many rays in all directions, resulting in a constant appearance regardless of the observer's viewpoint. Methods for diffuse surface reconstruction can therefore rely on the appearance of the object.

This chapter considers mirror surfaces, which exhibit *specular* reflections and whose appearances are a reflection of the surrounding environment. Under specular reflection, an incoming ray is reflected to a single outgoing ray. This special characteristic leads to different appearances of the mirror surface under different viewpoints, and renders diffuse surface reconstruction methods useless. Meanwhile, there exist many objects with a mirror surface in the man-made environment. The study of mirror surface reconstruction is therefore an important problem in computer vision.

In this chapter, we assume the mirror surface reflect a light ray only once, and tackle the mirror surface reconstruction problem by adopting a common approach of introducing motion to the environment. Unlike previous methods which require a fully calibrated camera and known motion, we propose a novel solution based on observing

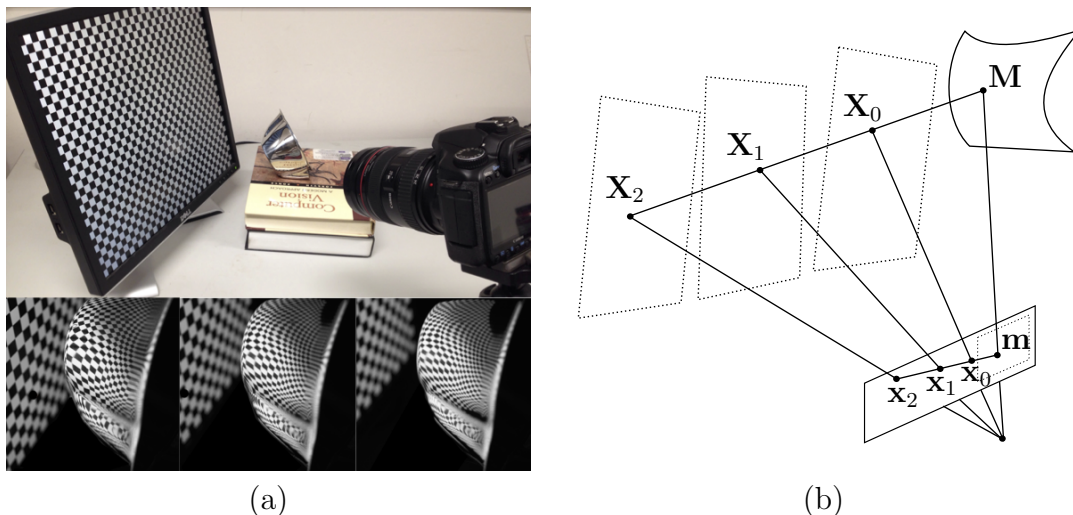


Fig. 3.1 (a) A stationary uncalibrated camera observing the reflections of a reference plane undergoing an unknown motion. (b) Surface points can be recovered using the cross-ratio between a surface point  $\mathbf{M}$  and its reflection correspondences  $\{\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2\}$ .

the reflections of a reference plane undergoing an unknown motion with a stationary uncalibrated camera (see Fig. 3.1(a)).

2D correspondences between the image and the reference plane are established by displaying a sweeping line on the plane (we use a computer screen as the reference plane in practice). The relative poses of the reference plane are then estimated [59], and rays piercing the plane under different poses are determined for each image point on the mirror surface.

Given the rays and their corresponding image points, we first derive an analytical solution to estimate the camera projection matrix through estimating the *line projection matrix*. Such a line projection matrix can then be transformed to a corresponding camera (point) projection matrix [60]. To make our solution more robust to noise, we use this closed-form solution as an initialization and optimize the camera projection matrix by minimizing reprojection errors computed based on a cross-ratio formulation for the mirror surface (see Fig. 3.1(b)). The mirror surface is finally reconstructed based on the optimized cross-ratio constraint.

The key contributions of this work are



- To the best of our knowledge, the first mirror surface reconstruction solution under an unknown motion and an uncalibrated camera.
- A closed-form (linear) solution for estimating the camera projection matrix from reflection correspondences.
- A cross-ratio based nonlinear formulation that allows a robust estimation of the camera projection matrix together with the mirror surface.

## 3.2 Related Work

Great efforts have been devoted to the problem of mirror surface recovery [14, 15, 49]. Based on the assumed prior knowledge, shape recovery methods for mirror surfaces can be classified into those assuming an *unknown distant* environment and those assuming a *known nearby* environment.

Under an *unknown distant* environment, a set of methods referred to as shape from specular flow (SFSF) have been proposed. In [61], Oren and Nayar successfully recovered a 3D curve on the object surface by tracking the trajectory of the reflection of a light source on the mirror surface. However, it is difficult to track a complete trajectory since the reflected feature will be greatly distorted near the occluding boundary of an object. Roth and Black [62] introduced the concept of specular flow and derived its relation with the 3D shape of a mirror surface. Although they only recovered a surface with a parametric representation (e.g., sphere), their work provides a theoretical basis for the later methods. In [63, 64], Adato *et al.* showed that under far-field illumination and large object-environment distance, the observed specular flow can be related to surface shape through a pair of coupled nonlinear partial differential equations (PDEs). Vasilyev *et al.* [65] further suggested that it is possible to reconstruct a smooth surface from one specular flow by inducing integrability constraints on the surface normal field. In [66], Canas *et al.* reparameterized the nonlinear PDEs as linear equations that lead to a more manageable solution.

Although SFSF achieves a theoretical breakthrough in shape recovery of mirror surfaces, the essential issues in tracking dense specular flow and in solving PDEs still hinder their practical use. In [67], Sankaranarayanan *et al.* developed an approach that uses sparse specular reflection correspondences instead of specular flow to recover a mirror surface linearly. Their proposed method is more practical than the traditional SFSF methods. Nevertheless, their method requires quite a number of specular reflection correspondences across different views, which are difficult to obtain due to the distorted reflections on the mirror surface.

Under a *known nearby* environment, a different set of methods for shape recovery of mirror surfaces can be derived. The majority of these methods are based on the smoothness assumption on the mirror surface. Under this assumption, one popular way is to formulate the surface recovery into the problem of solving PDEs. In [68, 69], Savarese and Perona demonstrated that local surface geometry of a mirror surface can be determined by analyzing the local differential properties of the reflections of two calibrated lines. Following the same fashion, Rozenfeld *et al.* [70] explored the 1D homography relationship between the calibrated lines and the reflections using sparse correspondences. Depth and first order local shape are estimated by minimizing a statistically correct measure, and a dense 3D surface is then constructed by performing a constrained interpolation. In [12], Liu *et al.* proved that a smooth mirror surface can be determined up to a two-fold ambiguity from just one reflection view of a calibrated reference plane.

Another way to formulate the mirror surface recovery is by employing normal consistency property to refine visual hull and/or integrate normal field. In [71], Bonfort and Sturm introduced a voxel carving method to reconstruct a mirror surface using a normal consistency criterion derived from the reflections of some calibrated reference planes. In order to get a better view for shape recovery, they further proposed that the camera may not need to face the reference plane, and the shape can be well recovered by using a mirror to calibrate the poses of the reference plane [72, 73]. In [74], Nehab *et al.* formulated the shape recovery as an image matching problem by minimizing

a cost function based on normal consistency. In [75], Weinmann *et al.* employed a turntable setup with multiple cameras and displays, which enables the calculation of the normal field for each reflection view. The 3D surface is then estimated by a robust multi-view normal field integration technique. In [76], Balzer *et al.* deployed a room-sized cube consisting of six walls that encode/decode specular correspondences based on a phase shift method. The surface is then recovered by integration of normal fields.

Another approach is to reconstruct the individual light paths based on the law of reflection. Kutulakos and Steger [31] showed that a point on a mirror surface can be recovered if the positions of two reference points are known in space and reflected to the same image point in a single view, or the positions of two reference points are known and are reflected by the same surface point to two different views. In [59], Liu *et al.* established reflection correspondences on the reference plane under three distinct poses, and derived a method for recovering the relative poses of the plane. Given the camera intrinsics, the camera pose can also be solved and the surface can be recovered by ray triangulation.

Note that calibration plays an important role in all the above methods that assume a known nearby environment. The calibration procedure is often very tedious and laborious for most of the mirror surface reconstruction systems. Normally, multiple images of a known pattern are required, and both the camera and the pattern poses need to be calibrated. Extra efforts may be needed due to the design of different systems. In this work, we neither make assumption on the smoothness of the mirror surface, nor require the calibration of the camera. Our proposed approach can automatically calibrate the setup as well as reconstruct the mirror surface using the observed reflections of the reference plane.

Cross-ratio constraint has been used to estimate mirror position and camera pose for axial non-central catadioptric systems [77], and produce more point correspondences in the context of 3D reconstruction [78]. Our method also relies on a cross-ratio constraint to optimize the camera projection matrix as well as recovering the mirror surface. Unlike existing methods where both the mirror and the reference plane are

simultaneously visible to the camera (e.g., [79]), we tackle a more challenging scenario where only the mirror surface is visible.

### 3.3 Acquisition Setup

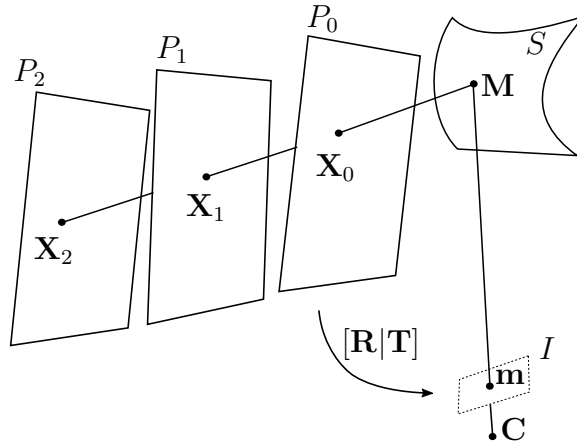


Fig. 3.2 Setup used for mirror surface reconstruction. Refer to Section 3.3 for notations and definitions.

Fig. 3.2 shows the setup used for mirror surface reconstruction. Consider a pinhole camera centered at  $\mathbf{C}$  observing the reflections of a moving reference plane on a mirror surface  $S$ . Let  $\mathbf{X}_0$  be a point on the plane at its initial pose, denoted by  $P_0$ , which is reflected by a point  $\mathbf{M}$  on  $S$  to a point  $\mathbf{m}$  on the image plane  $I$ . Suppose the reference plane undergoes an unknown rigid body motion, and let  $P_1$  and  $P_2$  denote the plane at its two new poses. Let  $\mathbf{X}_1$  and  $\mathbf{X}_2$  be points on  $P_1$  and  $P_2$ , respectively, which are both reflected by  $\mathbf{M}$  on  $S$  to the same image point  $\mathbf{m}$  on  $I$ .  $\mathbf{X}_0$ ,  $\mathbf{X}_1$  and  $\mathbf{X}_2$  are referred to as *reflection correspondences* of the image point  $\mathbf{m}$ .

### 3.4 A Closed-form Solution

In this section, we first briefly review Plücker coordinates and the line projection matrix. We then derive a linear method for obtaining a closed-form solution to the

line projection matrix of a camera from reflection correspondences of the image points.

### 3.4.1 Plücker Coordinates

A 3D line can be described by a skew-symmetric Plücker matrix  $\mathbf{L} = \mathbf{Q}\mathbf{P}^T - \mathbf{P}\mathbf{Q}^T =$

$$\begin{bmatrix} 0 & q_1p_2 - q_2p_1 & q_1p_3 - q_3p_1 & q_1p_4 - q_4p_1 \\ q_2p_1 - q_1p_2 & 0 & q_2p_3 - q_3p_2 & q_2p_4 - q_4p_2 \\ q_3p_1 - q_1p_3 & q_3p_2 - q_2p_3 & 0 & q_3p_4 - q_4p_3 \\ q_4p_1 - q_1p_4 & q_4p_2 - q_2p_4 & q_4p_3 - q_3p_4 & 0 \end{bmatrix}, \quad (3.1)$$

where  $\mathbf{P} = [p_1 \ p_2 \ p_3 \ p_4]^T$  and  $\mathbf{Q} = [q_1 \ q_2 \ q_3 \ q_4]^T$  are the homogeneous representations of two distinct 3D points on the line. Since  $\mathbf{L}$  is skew-symmetric, it can be represented simply by a Plücker vector  $\mathcal{L}$  consisting of its 6 distinct non-zero elements

$$\mathcal{L} = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{bmatrix} = \begin{bmatrix} q_1p_2 - q_2p_1 \\ q_1p_3 - q_3p_1 \\ q_1p_4 - q_4p_1 \\ q_2p_3 - q_3p_2 \\ q_3p_4 - q_4p_3 \\ q_4p_2 - q_2p_4 \end{bmatrix}. \quad (3.2)$$

Dually, a matrix  $\bar{\mathbf{L}}$  can be constructed from two distinct planes with homogeneous representations  $\hat{\mathbf{P}}$  and  $\hat{\mathbf{Q}}$  as  $\bar{\mathbf{L}} = \hat{\mathbf{Q}}\hat{\mathbf{P}}^T - \hat{\mathbf{P}}\hat{\mathbf{Q}}^T$ . The *dual Plücker vector* can be constructed directly from  $\bar{\mathbf{L}}$  or by rearranging the elements of  $\mathcal{L}$  as

$$\bar{\mathcal{L}} = [l_5 \ l_6 \ l_4 \ l_3 \ l_1 \ l_2]^T. \quad (3.3)$$

Let  $\mathbf{A} = [a_1 \ a_2 \ a_3]^T$  and  $\mathbf{B} = [b_1 \ b_2 \ b_3]^T$  be two distinct 3D points in Cartesian coordinates. Geometrically, the line defined by these points can be represented by a direction vector  $\boldsymbol{\omega} = (\mathbf{A} - \mathbf{B}) = [l_3, -l_6, l_5]^T$  and a moment vector  $\boldsymbol{\nu} = (\mathbf{A} \times \mathbf{B}) = [l_4, -l_2, l_1]^T$ , which define the line up to a scalar factor.

Two 3D lines  $\mathcal{L}$  and  $\mathcal{L}'$  can either be skew or coplanar. The geometric requirement for the latter case is that the dot product between the direction vector of the first line and the moment vector of the second line should equal the negative of the dot product between the direction vector of the second line and the moment vector of the first line. Let the two lines have direction vectors  $\boldsymbol{\omega}$ ,  $\boldsymbol{\omega}'$  and moment vectors  $\boldsymbol{\nu}$ ,  $\boldsymbol{\nu}'$ , respectively. They are coplanar (i.e., either coincident or intersect) if and only if

$$\boldsymbol{\omega} \cdot \boldsymbol{\nu}' + \boldsymbol{\nu} \cdot \boldsymbol{\omega}' = 0 \Leftrightarrow \mathcal{L} \cdot \bar{\mathcal{L}}' = 0. \quad (3.4)$$

Note that a Plücker vector is not any arbitrary 6-vector. A valid Plücker vector must always intersect itself, i.e.,

$$\mathcal{L} \cdot \bar{\mathcal{L}} = 0 \Leftrightarrow \det(\mathbf{L}) = 0. \quad (3.5)$$

### 3.4.2 Line Projection Matrix

Using homogeneous coordinates, a linear mapping can be defined for mapping a point  $\mathbf{X}$  in 3D space to a point  $\mathbf{x}$  in a 2D image, i.e.,

$$\mathbf{x} = \mathbf{P}\mathbf{X}, \quad (3.6)$$

where  $\mathbf{P}$  is a  $3 \times 4$  matrix known as the camera (point) projection matrix. Similarly, using Plücker coordinates, a linear mapping can be defined for mapping a line  $\mathcal{L}$  in 3D space to a line  $\mathbf{l}$  (in homogeneous coordinates) in a 2D image, i.e.,

$$\mathbf{l} = \mathcal{P}\bar{\mathcal{L}}, \quad (3.7)$$

where  $\mathcal{P}$  is a  $3 \times 6$  matrix known as the *line projection matrix*. Note that each row  $\mathbf{P}_i^T$  ( $i \in \{1, 2, 3\}$ ) of  $\mathbf{P}$  represents a *plane* (in homogeneous coordinates) that intersects at the optical center. Dually, each row  $\mathcal{P}_i^T$  ( $i \in \{1, 2, 3\}$ ) of  $\mathcal{P}$  represents a *line* that intersects at the optical center (see Fig. 3.3). It follows that a valid line projection

matrix must satisfy

$$\mathcal{P}_i \cdot \bar{\mathcal{P}}_j = 0 \quad \forall i, j \in \{1, 2, 3\} \Leftrightarrow \mathcal{P}\bar{\mathcal{P}}^T = \mathbf{0}_{3,3}, \quad (3.8)$$

where  $\bar{\mathcal{P}}_{\{1,2,3\}}$  denotes the dual Plücker vector (3.3) and  $\bar{\mathcal{P}} = [\bar{\mathcal{P}}_1 \ \bar{\mathcal{P}}_2 \ \bar{\mathcal{P}}_3]^T$ .

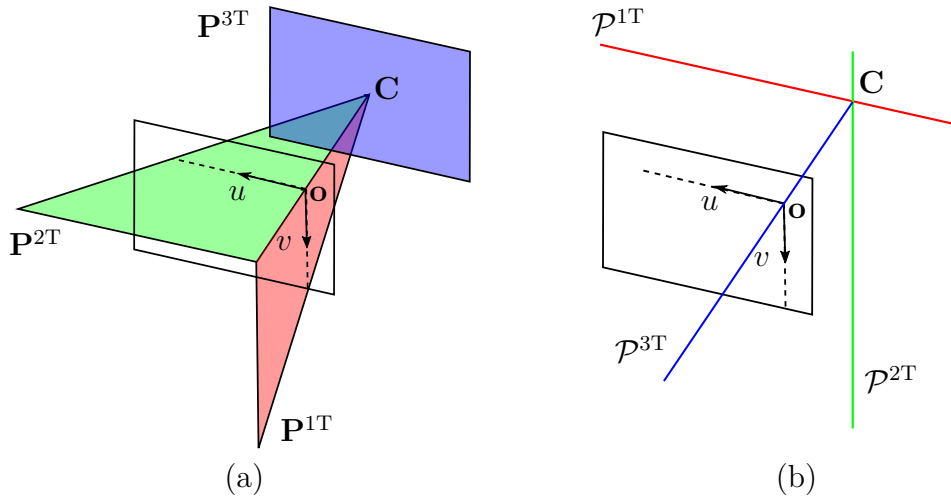


Fig. 3.3 (a) Rows of a point projection matrix represent planes that intersect at the optical center  $\mathbf{C}$  of the camera. (b) Dually, rows of a line projection matrix represent lines that intersect at the optical center.

### 3.4.3 Estimating the Line Projection Matrix

To estimate the line projection matrix of the camera, we first employ the method described in [59] to recover the relative poses of the reference plane under three distinct poses using reflection correspondences established across the images (see Appendix A). We can then form a 3D Plücker line  $\mathcal{L}$  from the reflection correspondences of each observed point  $\mathbf{x}$  in the image. Note that, by construction,  $\mathbf{x}$  must lie on the projection of  $\mathcal{L}$ , i.e.,

$$\mathbf{x}^T \mathcal{P}\bar{\mathcal{L}} = 0. \quad (3.9)$$

Given a set of 3D space lines  $\{\mathcal{L}_1, \dots, \mathcal{L}_n\}$  constructed for a set of image points  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ , the constraint derived in (3.9) can be arranged into

$$\mathbf{A}\vec{\mathcal{P}} = \mathbf{0}, \quad (3.10)$$

where  $\vec{\mathcal{P}} = [\mathcal{P}_1^T \ \mathcal{P}_2^T \ \mathcal{P}_3^T]^T$  and

$$\mathbf{A} = \begin{bmatrix} \mathbf{x}_1^T \otimes \bar{\mathcal{L}}_1^T \\ \vdots \\ \mathbf{x}_n^T \otimes \bar{\mathcal{L}}_n^T \end{bmatrix}.^1 \quad (3.11)$$

The line projection matrix of the camera can then be estimated by solving

$$\operatorname{argmin}_{\vec{\mathcal{P}}} \|\mathbf{A}\vec{\mathcal{P}}\|^2 \quad (3.12)$$

subject to  $\|\vec{\mathcal{P}}\| = 1$ . The line projection matrix obtained thus can be transformed into a point projection matrix and vice versa (see Appendix B). Note that, however, (3.12) minimizes only algebraic errors and does not enforce (3.8). The solution to (3.12) is therefore subject to numerical instability and not robust in the presence of noise.

Instead of solving (3.12), we can minimize the geometric distance from each image point to the projection of the corresponding 3D line. Let  $\mathbf{l} = [a, b, c]^T = \mathcal{P}\bar{\mathcal{L}}$  be the projection of the 3D line  $\mathcal{L}$  corresponding to an image point  $\mathbf{x} = [x_1, x_2, x_3]^T$ .  $\mathcal{P}$  can be estimated by solving

$$\operatorname{argmin}_{\mathcal{P}} \sum_{i=1}^n \frac{(\mathbf{x}_i^T \mathcal{P} \bar{\mathcal{L}}_i)^2}{a_i^2 + b_i^2} \quad (3.13)$$

subject to  $\|\mathcal{P}\| = 1$ , where  $\|\mathcal{P}\|$  is the Frobenius norm of  $\mathcal{P}$ . A straight-forward approach to enforce (3.8) is by incorporating it as a hard constraint in (3.13). However, experiments using a number of state-of-the-art optimization schemes show that such a solution often converges to local minima.

---

<sup>1</sup> $\otimes$  stands for Kronecker product.



### 3.4.4 Enforcing Constraints

Given a proper camera projection matrix, the corresponding line projection matrix will automatically satisfy (3.8). However, given an improper  $3 \times 6$  line projection matrix not satisfying (3.8), the corresponding camera projection matrix cannot be decomposed into one with proper intrinsic and extrinsic parameters. Based on this observation, we propose to enforce (3.8) by enforcing a proper decomposition of the camera projection matrix.

Consider a simplified scenario where the principal point  $(u_0, v_0)$  (which is often located at the image centre) is known. After translating the image origin to the principal point, the camera projection matrix can be expressed as

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \ \mathbf{T}] = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix},$$

and the corresponding line projection matrix can be expressed as

$$\mathcal{P} = \begin{bmatrix} f_y & 0 & 0 \\ 0 & f_x & 0 \\ 0 & 0 & f_x f_y \end{bmatrix} \mathcal{P}', \quad (3.14)$$

where

$$\mathcal{P}'_i{}^\Gamma = \begin{bmatrix} \rho'_{i1} \\ \rho'_{i2} \\ \rho'_{i3} \\ \rho'_{i4} \\ \rho'_{i5} \\ \rho'_{i6} \end{bmatrix} = (-1)^{(i+1)} \begin{bmatrix} r_{j3}t_k - t_j r_{k3} \\ t_j r_{k2} - r_{j2}t_k \\ r_{j2}r_{k3} - r_{j3}r_{k2} \\ r_{j1}t_k - t_j r_{k1} \\ r_{j1}r_{k2} - r_{j2}r_{k1} \\ r_{j1}r_{k3} - r_{j3}r_{k1} \end{bmatrix}, \quad (3.15)$$

with  $i \neq j \neq k \in \{1, 2, 3\}$  and  $j < k$ . (3.10) can then be rewritten as

$$\mathbf{A}\vec{\mathcal{P}} = \mathbf{A}\mathbf{D}\vec{\mathcal{P}}' = \mathbf{A}'\vec{\mathcal{P}}' = 0, \quad (3.16)$$

where  $\mathbf{A}' = \mathbf{A}\mathbf{D}$  and  $\mathbf{D}$  is a  $18 \times 18$  diagonal matrix with  $d_{ii} = f_y$  for  $i \in \{1, \dots, 6\}$ ,  $d_{ii} = f_x$  for  $i \in \{7, \dots, 12\}$ , and  $d_{ii} = f_x f_y$  for  $i \in \{13, \dots, 18\}$ .

With known  $f_x$  and  $f_y$ ,  $\vec{\mathcal{P}}'$  can be estimated by solving (3.16). Since  $\mathcal{P}'$  only depends on the elements of  $\mathbf{R}$  and  $\mathbf{T}$ , it can be converted to a point projection matrix in the form of  $\lambda[\mathbf{R} \ \mathbf{T}]$ . The magnitude of  $\lambda$  is determined by the orthogonality of  $\mathbf{R}$ , and its sign is determined by the sign of  $t_3$ . Hence, given the camera intrinsics, the camera extrinsics can be recovered using the reflection correspondences. [59] also provides another way for estimating  $\mathbf{R}$  and  $\mathbf{T}$  with given camera intrinsics. In Section 3.5, we tackle the problem of unknown camera intrinsics by formulating the problem into a nonlinear optimization by minimizing reprojection errors computed based on a cross-ratio formulation for the mirror surface. For initialization purpose, we assume  $(u_0, v_0)$  being located at the image center, and  $f_x = f_y = f$ . We choose a rough range of  $f$  and for each sample value of  $f$  within the range, we estimate  $\mathbf{R}$  and  $\mathbf{T}$  by solving (3.16). The point to line distance criterion in (3.13) is applied to find the best focal length  $f'$ . A camera projection matrix can then be constructed using  $f'$ ,  $(u_0, v_0)$ ,  $\mathbf{R}$  and  $\mathbf{T}$  that satisfies all the above mentioned constraints.

### 3.5 Cross-ratio Based Formulation

In this section, we obtain the camera projection matrix and the mirror surface by minimizing reprojection errors. We will derive a cross-ratio based formulation for recovering a 3D point on the mirror surface from its reflection correspondences. Note that minimizing point-to-point reprojection errors can provide a stronger geometrical constraint than minimizing the point-to-line distances in (3.13) (see Fig. 3.4).

Consider a point  $\mathbf{M}$  on the mirror surface (see Fig. 3.5). Let  $\mathbf{X}_0$ ,  $\mathbf{X}_1$  and  $\mathbf{X}_2$  be its reflection correspondences on the reference plane under three distinct poses, denoted

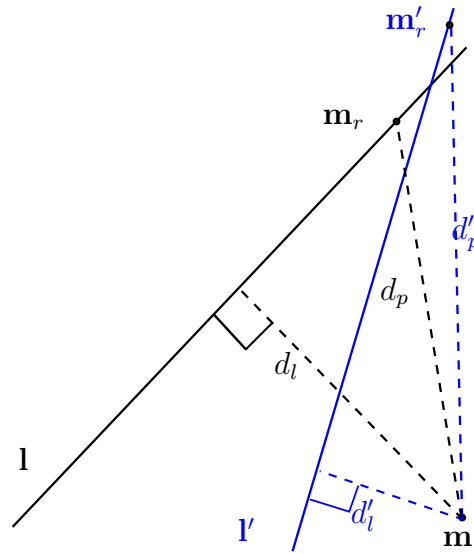


Fig. 3.4 Minimizing point-to-line distance does not guarantee minimizing point-to-point distance. A 3D point  $\mathbf{M}$  and a 3D line  $\mathcal{L}$  passing through it are projected by  $\mathcal{P}$  to a 2D point  $\mathbf{m}_r$  and a 2D line  $\mathbf{l}$ , respectively. Let  $\mathbf{m}$  denote the observation of  $\mathbf{M}$ . The distance between  $\mathbf{m}$  and  $\mathbf{m}_r$  is  $d_p$ , and the distance between  $\mathbf{m}$  and  $\mathbf{l}$  is  $d_l$ . Suppose the same 3D point  $\mathbf{M}$  and 3D line  $\mathcal{L}$  are projected by  $\mathcal{P}'$  to  $\mathbf{m}'_r$  and  $\mathbf{l}'$ , respectively. The distance between  $\mathbf{m}$  and  $\mathbf{m}'_r$  is  $d'_p$ , and the distance between  $\mathbf{m}$  and  $\mathbf{l}'$  is  $d'_l$ . Note that  $d'_l < d_l$ , but  $d'_p > d_p$ .

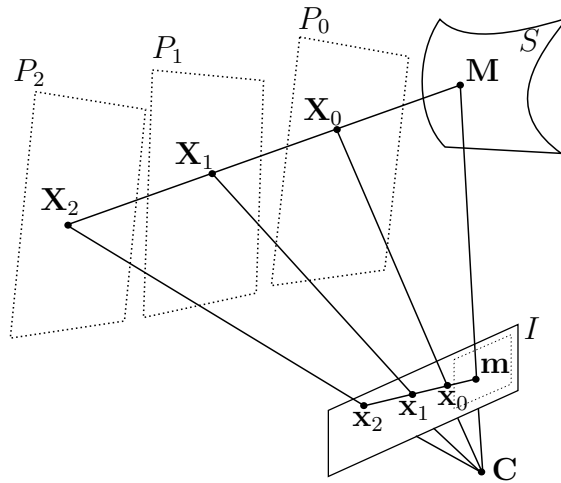


Fig. 3.5 Camera projection matrix and mirror surface points are recovered by minimizing reprojection errors computed from the cross-ratio constraint  $\{\mathbf{M}, \mathbf{X}_0; \mathbf{X}_1, \mathbf{X}_2\} = \{\mathbf{m}, \mathbf{x}_0; \mathbf{x}_1, \mathbf{x}_2\}$ , where  $\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2$  are the correspondences of  $\mathbf{M}$  under three different pattern poses and  $\mathbf{m}, \mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2$  are their projections on image plane. Note that  $\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2$  may not be visible by the camera.

by  $P_0$ ,  $P_1$  and  $P_2$ , respectively. Suppose  $\mathbf{M}$ ,  $\mathbf{X}_0$ ,  $\mathbf{X}_1$  and  $\mathbf{X}_2$  are projected to the image as  $\mathbf{m}$ ,  $\mathbf{x}_0$ ,  $\mathbf{x}_1$  and  $\mathbf{x}_2$  respectively. We observe that the cross-ratios  $\{\mathbf{M}, \mathbf{X}_0; \mathbf{X}_1, \mathbf{X}_2\}$  and  $\{\mathbf{m}, \mathbf{x}_0; \mathbf{x}_1, \mathbf{x}_2\}$  are identical, i.e.,

$$\frac{|\overline{\mathbf{X}_1\mathbf{M}}||\overline{\mathbf{X}_2\mathbf{X}_0}|}{|\overline{\mathbf{X}_1\mathbf{X}_0}||\overline{\mathbf{X}_2\mathbf{M}}|} = \frac{|\overline{\mathbf{x}_1\mathbf{m}}||\overline{\mathbf{x}_2\mathbf{x}_0}|}{|\overline{\mathbf{x}_1\mathbf{x}_0}||\overline{\mathbf{x}_2\mathbf{m}}|}. \quad (3.17)$$

Let  $s$  be the distance between  $\mathbf{X}_2$  and  $\mathbf{M}$  (i.e.,  $s = |\overline{\mathbf{X}_2\mathbf{M}}|$ ), from (3.17)

$$s = \frac{|\overline{\mathbf{X}_2\mathbf{X}_1}||\overline{\mathbf{X}_2\mathbf{X}_0}||\overline{\mathbf{x}_1\mathbf{x}_0}||\overline{\mathbf{x}_2\mathbf{m}}|}{|\overline{\mathbf{X}_2\mathbf{X}_0}||\overline{\mathbf{x}_1\mathbf{x}_0}||\overline{\mathbf{x}_2\mathbf{m}}| - |\overline{\mathbf{X}_1\mathbf{X}_0}||\overline{\mathbf{x}_2\mathbf{x}_0}||\overline{\mathbf{x}_1\mathbf{m}}|}. \quad (3.18)$$

Given the projection matrix,  $\mathbf{x}_0$ ,  $\mathbf{x}_1$ ,  $\mathbf{x}_2$  and  $\mathbf{m}$ , a surface point  $\mathbf{M}$  can be recovered as

$$\mathbf{M} = \mathbf{X}_2 + s \frac{\overrightarrow{\mathbf{X}_2\mathbf{X}_0}}{|\overline{\mathbf{X}_2\mathbf{X}_0}|}, \quad (3.19)$$

where  $\overrightarrow{\mathbf{X}_2\mathbf{X}_0}$  denotes the directed ray from  $\mathbf{X}_2$  to  $\mathbf{X}_0$ .

We optimize the projection matrix by minimizing the reprojection errors, i.e.,

$$\operatorname{argmin}_{\theta} \sum_{i=1}^n (\mathbf{m}_i - \mathbf{m}'_i)^2, \quad (3.20)$$

where  $\theta = [f_x, f_y, u_0, v_0, r_x, r_y, r_z, t_x, t_y, t_z]^T$ <sup>2</sup>,  $\mathbf{m}_i \in \mathbb{R}^2$  is the observation of  $\mathbf{M}_i \in \mathbb{R}^3$ , and  $\mathbf{m}'_i = \mathbf{P}(\theta)\mathbf{M}_i$ . We initialize  $\theta$  using the method proposed in Section 3.4, and solve the optimization problem using the Levenberg-Marquardt method. Given the estimated projection matrix, the mirror surface can be robustly reconstructed by solving (3.17)-(3.19).

## 3.6 Evaluation

To demonstrate the effectiveness of our method, we evaluate it using both synthetic and real data.

<sup>2</sup>We used angle-axis representation for rotation, i.e.,  $[r_x, r_y, r_z]^T = \alpha \mathbf{e}$ , where  $\alpha$  is the rotation angle and  $\mathbf{e}$  is the unit rotation axis.

### 3.6.1 Synthetic Data

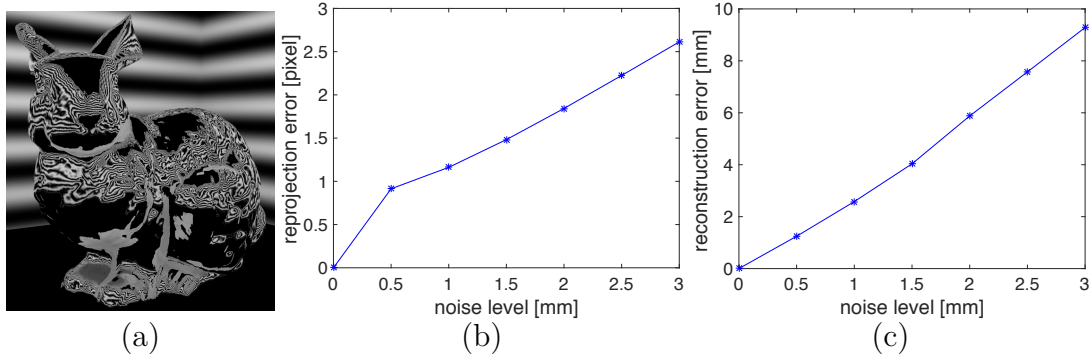


Fig. 3.6 (a) An image of the mirror *Stanford bunny*. (b) RMS reprojection errors (computed against ground truth image points). (c) RMS reconstruction errors (computed against ground truth 3D surface points).

We employed a reflective *Stanford bunny* rendered by Balzer *et al.* [76] to generate our synthetic data. The bunny has a dimension of  $880 \times 680 \times 870 \text{ mm}^3$ , and 208,573 surface points. The images have a resolution of  $960 \times 1280$  pixels. Fig. 3.6(a) shows the reflective appearance of the bunny. In their original data, the bunny is placed in a cubic room, with each side of the room working as a reference plane. The reference pattern has a dimension of  $3048 \times 3048 \text{ mm}^2$ . The center of the room is defined as the world origin. A camera is placed in the room viewing the bunny. Since our method requires reflection correspondences under three distinct poses of a reference plane, we introduced two additional planes for each side of the room and obtained the reflection correspondences through ray tracing.

To evaluate the performance of our method, we added Gaussian noise to the reflection correspondences with standard deviations ranging from 0 to 3.0 *mm*. We initialized the projection matrix using the method described in Section 3.4. The optimized projection matrix together with the 3D surface points were obtained by minimizing reprojection errors computed based on our cross-ratio formulation. Our cross-ratio based formulation can effectively improve the initialization results. An example is given in Table 3.1. We evaluated: (1) linear solution by solving (3.12) in

Section 3.4.3, denoted as  $L$ ; (2) linear solution by solving (3.16) after enforcing the constraints in Section 3.4.4, denoted as  $EL$ ; (3) cross-ratio based non-linear solution by solving (3.20) in Section 3.5, denoted as  $CR$ . The error in the rotation matrix  $\mathbf{R}$  is the angle of the rotation induced by  $\mathbf{R}_{gt}\mathbf{R}^T$ , where  $\mathbf{R}_{gt}$  denotes the ground truth rotation matrix. The error in the translation vector  $\mathbf{T}$  is the angle ( $\mathbf{T}_{deg}$ ) between  $\mathbf{T}$  and  $\mathbf{T}_{gt}$ , where  $\mathbf{T}_{gt}$  denotes the ground truth translation vector. In addition, we obtain  $\mathbf{T}_{scale} = \|\mathbf{T}_{gt} - \mathbf{T}\|$  to estimate the error in  $\mathbf{T}$ . After applying the cross-ratio based non-linear optimization, the accuracy of the camera projection matrix can be effectively improved.

Table 3.1 Estimation error under noise lv  $\sigma = 2.0$  [mm] on *bunny*.  $L$ : linear solution of Section 3.4.3;  $EL$ : constrained linear solution with strategy in Section 3.4.4;  $CR$ : estimation using cross-ratio formulation initialized with  $EL$ .

	$f_u$	$f_v$	$u_0$	$v_0$	$\mathbf{R}[\circ]$	$\mathbf{T}_{deg}[\circ]$	$\mathbf{T}_{scale}$
$L$	1.39%	1.78%	1.76%	2.39%	0.61	0.55	0.98%
$EL$	0.94%	0.94%	0.07%	0.10%	0.12	0.11	0.28%
$CR$	0.11%	0.11%	0.18%	0.25%	0.08	0.07	0.13%

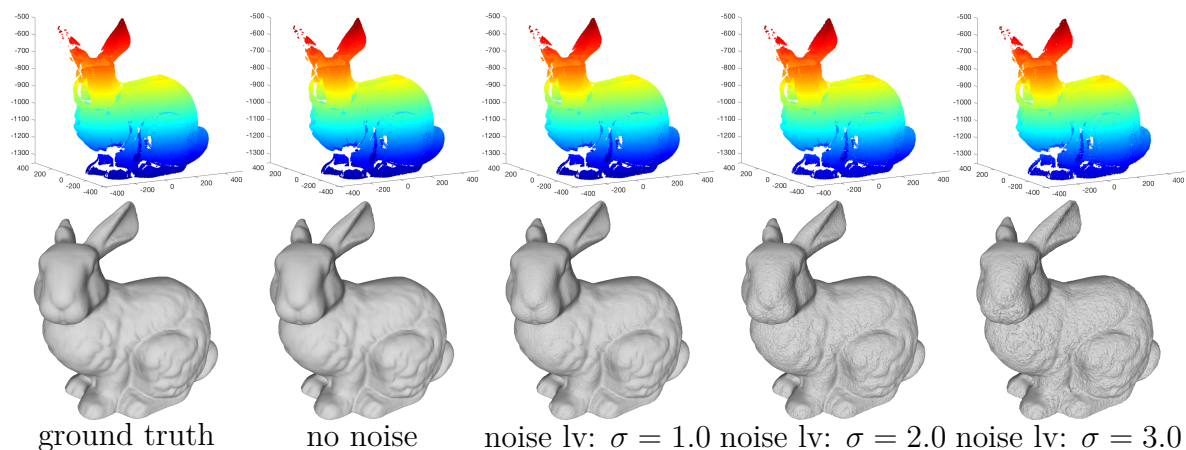


Fig. 3.7 Top row: reconstructed point clouds under different noise levels. Coordinates are w.r.t world and colors are rendered w.r.t  $z$  coordinates. Note that the missing regions are due to the lack of correspondences in the original data set. Bottom row: surfaces generated using screened Poisson surface reconstruction method [80].

Fig. 3.6(b) and (c) depict the root mean square (RMS) reprojection errors and reconstruction errors, respectively, under different noise levels. It can be seen that

Table 3.2 Camera intrinsic and extrinsic estimation error under different noise level  $\sigma$  for the *Stanford bunny* dataset. The ground truth for the intrinsic parameters are  $f_u = 1400$ ,  $f_v = 1400$ , and  $(u_0, v_0) = (639.5, 479.5)$ .

	$f_u$ [pixel]	$f_v$ [pixel]	$u_0$ [pixel]	$v_0$ [pixel]	$\mathbf{R}$ [°]	$\mathbf{T}_{deg}$ [°]	$\mathbf{T}_{scale}$ [mm]
$\sigma = 0.5$	0.22(0.02%)	0.22(0.02%)	0.41(0.06%)	0.07(0.01%)	0.02	0.02	0.53(0.03%)
$\sigma = 1.0$	0.33(0.02%)	0.33(0.02%)	0.41(0.06%)	0.05(0.01%)	0.02	0.02	0.52(0.03%)
$\sigma = 1.5$	0.50(0.04%)	0.51(0.04%)	0.56(0.09%)	0.90(0.20%)	0.04	0.05	1.74(0.10%)
$\sigma = 2.0$	1.52(0.11%)	1.52(0.11%)	1.15(0.18%)	1.22(0.25%)	0.08	0.07	2.42(0.13%)
$\sigma = 2.5$	4.36(0.31%)	4.36(0.31%)	2.11(0.32%)	2.36(0.49%)	0.37	0.56	7.33(0.40%)
$\sigma = 3.0$	10.15(0.73%)	10.11(0.73%)	5.76(0.90%)	3.08(0.64%)	0.85	0.73	13.29(0.73%)

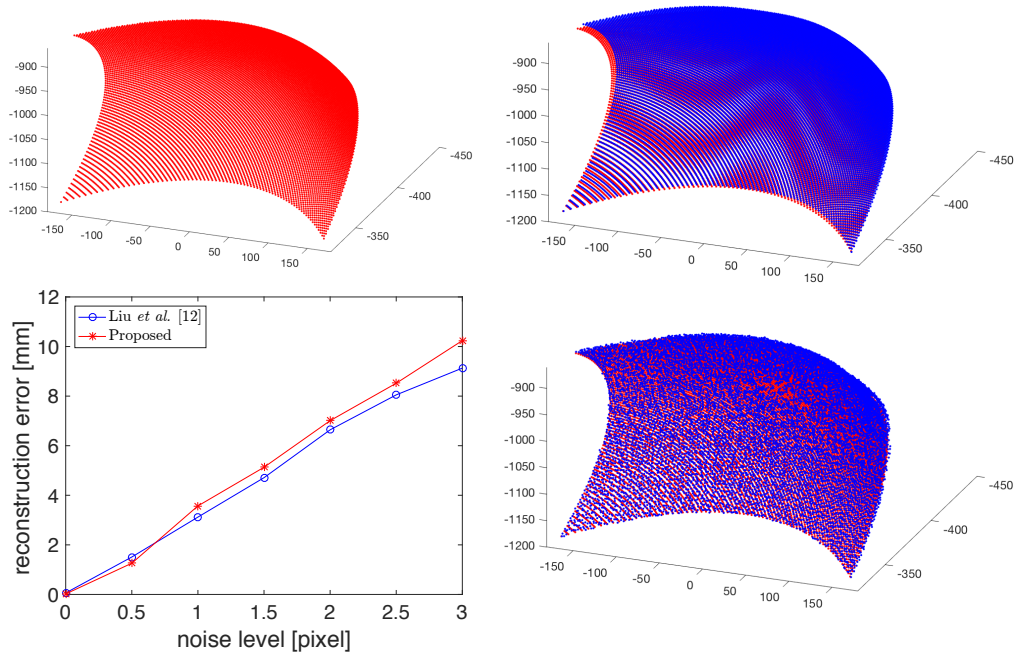


Fig. 3.8 Comparison with a fully calibrated method [12]. Upper left: ground truth. Lower left: RMS reconstruction errors. Upper right ([12]) & lower right (ours): reconstruction (blue) against ground truth (red) under  $\sigma = 2.0$ . Our uncalibrated approach achieves comparable accuracy with that of the fully calibrated method [12].

the reprojection errors and the reconstruction errors increase linearly with the noise level. The magnitude of the reconstruction errors is relatively small compared to the size of the object. Fig. 3.7 shows the reconstructed point clouds and surfaces. Table 3.2 shows a quantitative comparison of our estimated projection matrices w.r.t the ground truth. Among all noise levels, the errors are below 1% for  $f_u$ ,  $f_v$ ,  $u_0$ ,  $v_0$  and  $\mathbf{T}_{scale}$ , and angular errors are below  $1^\circ$  for  $\mathbf{R}$  and  $\mathbf{T}$ .

Besides, we compared our method with state-of-the-art mirror surface reconstruction method [12] under smooth surface assumption and calibrated setup. Note that [12] assumes the mirror surface is  $C^2$  continuous. In order to make fair comparison, we perform the experiment on a *sphere* patch under the same setup with the *bunny* dataset. Fig. 3.8 depicts the comparison between fully calibrated [12] and uncalibrated (proposed) methods. The overall reconstruction accuracy is similar. While our result is not as smooth as that from [12] due to our point-wise reconstruction, their result shows a global reconstruction bias due to the B-spline parameterization for the surface (see Fig. 3.8).

### 3.6.2 Real Data



Fig. 3.9 Top row: *sauce boat* and *two spheres* in real experiments. Bottom row: a sweeping line is reflected by *two spheres* under three distinct positions of the LCD monitor while the camera and mirror surfaces are stationary.

We evaluated our method on a *sauce boat* and *two spheres* respectively (see Fig. 3.9). We captured images using a *Canon EOS 40D* digital camera with a 24-70 *mm* lens.



A 19 *inch* LCD monitor was used as a reference plane and was placed at three different positions. For each position, we captured an image sequence of a thin bright stripe sweeping across the screen in vertical direction and then in horizontal direction [22, 31]. For each direction, we examined the intensity value sequence for each image point, and established the reflection correspondence by identifying the image in which the intensity attained a peak value. To improve the accuracy, quadratic approximation was applied to the intensity profile in the neighborhood of the peak value.

After establishing reflection correspondences, we first estimated the relative poses of the reference plane using the method in [59]. We then formed 3D lines from the reflection correspondences on the reference plane under the two poses that are furthest apart (e.g.,  $P_0$  and  $P_2$  in Fig. 3.5). These 3D lines were used to obtain a preliminary solution of projection matrix using the method in Section 3.4, which was then used to initialize the nonlinear optimization described in Section 3.5.

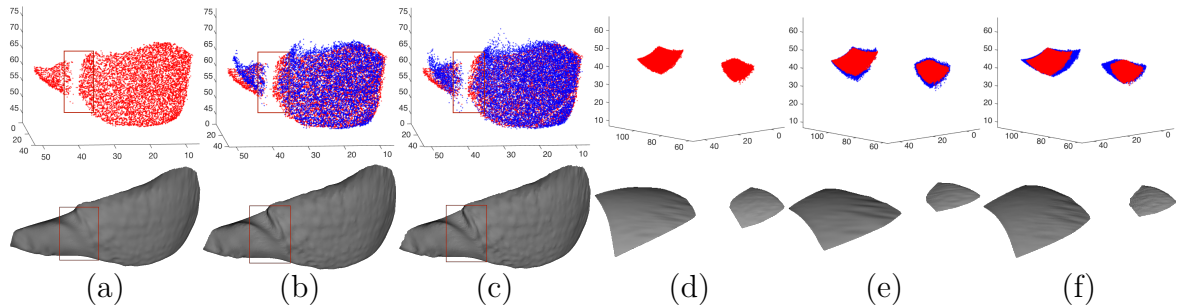


Fig. 3.10 (a)-(c): reconstructions of *sauce boat*. Results are obtained under (a) calibrated camera with calibrated plane poses (this result is treated as ground truth and overlaid in (b) and (c) for comparison (red)); (b) uncalibrated camera with calibrated plane poses (blue); (c) uncalibrated camera with uncalibrated plane poses (ours, blue). Note the missing regions (in red rectangle) in the reconstructed point clouds are filled by the mesh generation algorithm and should be ignored in comparing the surface meshes. (d)-(f): reconstructions of *two spheres*.

To evaluate our method, we calibrated the camera and reference plane poses using [81]. We used the calibration result to estimate the surface and treated it as the ground truth, due to the absence of the ground-truth surface. This result was compared against the result obtained using uncalibrated camera but calibrated plane poses, and

our result using uncalibrated camera and uncalibrated plane poses. Fig. 3.10 shows the reconstructed surfaces and Table 3.3 shows the numerical errors. We aligned each estimated surface with the ground truth by a rigid body transformation before computing the reconstruction error [82]. The RMS reconstruction errors are below 3 mm.  $f_u$  and  $f_v$  errors are below 2%.  $u_0$ ,  $v_0$  and  $\mathbf{T}_{scale}$  errors are below 10%. The angular errors are below  $10^\circ$  for  $\mathbf{R}$  and below  $3^\circ$  for  $\mathbf{T}$ . The errors in intrinsics and extrinsics are larger than those in the synthetic experiments. This is reasonable since accurate specular correspondences in real case are difficult to obtain due to the large and complex distortion caused by the mirror surface and varying lighting condition. The quality of specular correspondences is also a key factor for existing mirror surface reconstruction methods. However, existing methods are designed under certain assumptions (e.g., convex,  $C^n$  continuity, etc), and their setups are carefully tailored or require special equipments. Besides, there is no public available dataset that can serve as input for existing methods. As a result, it is challenging to make a fair comparison with existing methods on real dataset. Therefore, we didn't include comparison with other existing methods on real data.

Table 3.3 Real experiments evaluation.  $B$  and  $S$  denote the results of *sauce boat* and *two spheres*, respectively. The subscripts  $uc$  and  $uu$  stand for experiments under an uncalibrated camera with calibrated plane poses and under an uncalibrated camera with uncalibrated plane poses, respectively. The ground truth for the intrinsic parameters are  $f_u = 5812.86$ ,  $f_v = 5812.82$ , and  $(u_0, v_0) = (1971.95, 1230.02)$ .  $S_{rms}$  stands for the RMS reconstruction error.

	$f_u$ [pixel]	$f_v$ [pixel]	$u_0$ [pixel]	$v_0$ [pixel]	$\mathbf{R}$ [ $^\circ$ ]	$\mathbf{T}_{deg}$ [ $^\circ$ ]	$\mathbf{T}_{scale}$ [mm]	$S_{rms}$ [mm]
$B_{uc}$	36.70(0.63%)	21.99(0.38%)	99.10(5.03%)	100.00(8.13%)	9.12	1.00	19.16(8.23%)	2.55
$B_{uu}$	101.70(1.75%)	86.90(1.49%)	112.10(5.69%)	113.00(9.19%)	9.86	1.99	17.02(7.34%)	2.71
$S_{uc}$	63.38(1.09%)	68.01(1.17%)	61.49(3.18%)	42.7(3.47%)	6.67	1.78	33.83(8.96%)	1.78
$S_{uu}$	81.38(1.40%)	86.02(1.48%)	81.67(4.14%)	56.70(4.61%)	7.17	2.13	37.69(9.98%)	2.03

## 3.7 Discussions and Conclusions

A novel method is introduced for mirror surface reconstruction. Our method works under an uncalibrated setup and can recover the camera intrinsics and extrinsics, along with the surface. We first proposed an analytical solution for camera projection matrix estimation, and then derived a cross-ratio based formulation to achieve a robust estimation. Our cross-ratio based formulation does not encounter degeneracy. However, degenerate cases (e.g., a planar mirror, a spherical mirror, etc) may occur to the system due to the application of [59] to estimate relative poses of the reference plane. Employing methods without degeneracy to estimate the relative poses will help handle these cases.

The proposed method only needs reflection correspondences as input and removes the restrictive assumptions of known motions,  $C^m$  continuity of the surface, and calibrated camera(s) that are being used by other existing methods. This greatly simplifies the challenging problem of mirror surface recovery. We believe our work can provide a meaningful insight towards solving this problem. In the future, we would like to extend the proposed method to recover complete surfaces and investigate inter-reflection cases.



# Chapter 4

## Single View Diffuse Surface Reconstruction

### 4.1 Introduction

3D reconstruction has always been a hot topic in the field of computer vision. Tremendous efforts have been devoted to this problem in the past decades. In particular, multi-view stereo (MVS) has been one of the most popular and successful approaches in solving this problem, and numerous state-of-the-art MVS methods have been proposed (e.g., [83]). Single view approach, on the other hand, has received much less attention compared with MVS. Despite its potential, single view approach is relatively less studied in the literature.

The working principle of most single view methods is based on observing multiple light paths to the same scene point. This is often accomplished by introducing one or more mirrors into the scene and observing the reflection(s) of the scene on the mirror(s). Both planar and spherical mirrors have been employed by such methods. Planar mirrors do not introduce any distortion in the reflected images, but provide only a very limited field of view (FOV) (e.g., [84–86]). Spherical mirrors, on the other hand, can provide a much wider FOV for 3D reconstruction (e.g., [87–90]). However, distortions do exist in the reflected images, and these make the correspondence problem

more difficult.

Most of the existing mirror based methods for single view 3D reconstruction require a fully calibrated setup, in which the intrinsic parameters of both the camera and mirrors, as well as the positions and orientations of the mirrors (relative to the camera) are assumed to be known. This often requires tedious calibration that hinders the application of these methods. In [16], Chen *et al.* showed that, with an internally calibrated camera, it is possible to recover the position of a mirror sphere from its image up to a scale determined by its radius. This allows a 3D reconstruction up to an unknown scale under an unknown radius of the mirror sphere.

In this chapter, we revisit the problem studied in [16], which is single view 3D reconstruction using an unknown mirror sphere, but with an uncalibrated camera. We reconstruct the surface of a diffuse object using its reflection on the unknown mirror sphere. During data acquisition, the camera and the object are fixed while the mirror sphere is placed at one or more locations. It is well known that, under perspective projection, the image of a sphere would be a conic [60]. Based on eigen decomposition of the matrix representing the conic image and enforcing a repeated eigenvalue constraint, we derive an analytical solution for recovering the focal length of the camera given its principal point. Based on this analytical solution, we develop two robust algorithms for estimating both the principal point and focal length of the camera from multiple images as well as from just one image of the mirror sphere. With the estimated camera intrinsic parameters, the position(s) of the sphere can be readily retrieved from the eigen decomposition(s) as in [16], and a scaled 3D reconstruction follows. The key contributions of this work include

- To the best of our knowledge, the first single view 3D reconstruction method that works under an uncalibrated camera and an unknown mirror sphere (i.e., one with both its position and radius being unknown).
- An analytical solution for recovering the focal length of a camera from an image of an unknown sphere given the principal point of the camera.

- A robust method for estimating both the principal point and focal length of a camera from multiple images of an unknown sphere placed at different positions.
- A novel method for estimating both the principal point and focal length of a camera from just one image of an unknown sphere.

The rest of this chapter is organized as follows. Section 4.2 gives a brief literature review. Section 4.3 provides the theoretical background of this work. The proposed algorithms for estimating both the principal point and focal length of a camera are introduced in Section 4.4, followed by a brief description of the reconstruction method in Section 4.5. Experimental results are presented in Section 4.6. Section 4.7 discusses and concludes this work.

## 4.2 Related Work

Imaging systems consisting of a camera observing one or more mirrors are referred to as a catadioptric imaging system. They have many applications in both computer graphics and computer vision [15], including panoramic imaging [91], stereo vision [84, 86, 87, 92, 93], light field imaging [94], recognition [95], etc. Both planar mirrors and spherical mirrors are commonly used in constructing a catadioptric imaging system.

A planar mirror provides a very cheap way of constructing a new viewpoint, and is the simplest device for building a stereo vision system from a single camera (e.g., [85]). It is useful in applications like 3D reconstruction [96, 97] and light field imaging [98]. In [99], Mitsumoto *et al.* described the single planar mirror geometric constraints for 3D reconstruction. They showed that it is possible to recover a large coverage of an object by moving the planar mirror or by placing multiple planar mirrors around the object. In [86], Gluckman and Nayar used multiple planar mirrors to build a complex imaging geometry. A planar mirror has the advantage of not introducing any distortion in the reflected image. However, its small FOV greatly hinders its use in practice.

A spherical mirror can provide a much wider FOV than a planar mirror, and may even reflect the entire surrounding environment. This makes it more commonly used in catadioptric systems. Existing methods often assume an internally calibrated camera, and consider only extrinsic calibration (e.g., [73, 77, 100, 101]). In [93], Nayar presented the spherio system for scene depth recovery. It consists of a calibrated camera looking at two specular spheres, with both the radii and positions of the spheres being known a priori. In [102], Powell *et al.* recovered light source positions from specular highlights observed on surfaces of spherical mirrors. In [89], Lanman *et al.* built a catadioptric system using a perspective camera and many identical spherical mirrors. They recovered the parameters of the spherical mirrors by a tailored calibration method. In [92], Kanbara *et al.* attached a color marker around the camera lens to estimate the visual ray passing through a spherical mirror with a known radius. In [103, 104], Wong *et al.* estimated the camera poses and light source directions from a sphere with unknown radius and position using specular highlights observed on the sphere and the silhouettes of the sphere. In [105] and [106], the corneas were approximated as spheres, and exploited for reconstructing a display from its reflections on the corneas. In [16], Chen *et al.* introduced a method to reconstruct a 3D object using a moving spherical mirror. They showed that without knowing the radius of the spherical mirror, a 3D surface can be recovered up to an unknown scale. Note that all the aforementioned methods assume an internally calibrated camera and/or a known sphere.

Other than using mirrors to build a catadioptric system, there is also much effort aiming at recovering the mirror surface itself from reflections observed on its surface (e.g., [13, 14, 49]). These, however, are out of the scope of this chapter. Besides, diffuse spheres have also been explored in camera calibration (e.g., [107–111]). These methods normally require multiple images and multiple spheres.



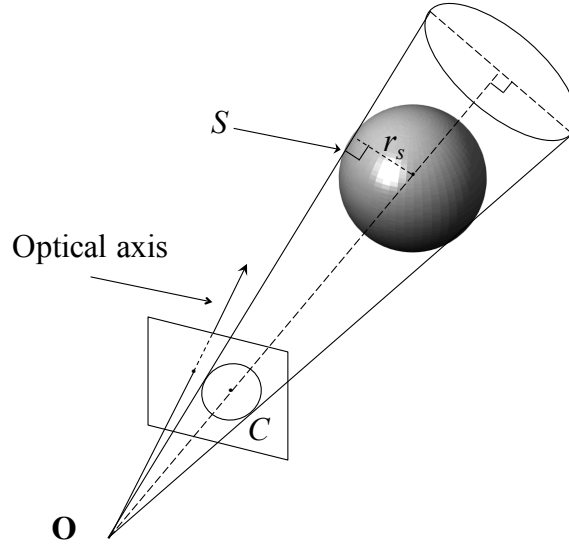


Fig. 4.1 A perspective camera located at  $\mathbf{O}$  observes a sphere  $S$  of radius  $r_s$ . The image of the sphere  $S$  is a conic  $C$ , which can be represented by a  $3 \times 3$  symmetric matrix  $\mathbf{C}$ .  $\mathbf{O}$  and  $C$  define a right circular cone tangent to  $S$ . The axis of this cone pierces the center of  $S$ . By construction, this axis is the  $Z$ -axis of the world coordinate, and the distance between  $\mathbf{O}$  and the center of  $S$  is  $d$ .

### 4.3 Theoretical Background

Without loss of generality, consider a pinhole camera with its optical centre located at the origin  $\mathbf{O}$  of a world coordinate system, and a sphere of radius  $r_s$  with its centre located at a distance  $d$  from  $\mathbf{O}$  along the positive  $Z$ -axis of the world coordinate system (see Fig. 4.1). The projection matrix of this camera can be written as  $\mathbf{P} = \mathbf{K}[\mathbf{R} \ \mathbf{0}]$ , where  $\mathbf{K}$  is the  $3 \times 3$  camera calibration matrix composed of the camera intrinsic parameters, and  $\mathbf{R}$  and  $\mathbf{0}$  are the  $3 \times 3$  rotation matrix and the translation vector  $[0 \ 0 \ 0]^T$ , respectively, which define the rigid body transformation of the camera with respect to the world coordinate system. The sphere will project onto the image as a conic. This conic can be represented by a  $3 \times 3$  symmetric matrix  $\mathbf{C}$  such that each point  $\mathbf{x}$  (in homogeneous pixel coordinates) lying on this conic satisfies  $\mathbf{x}^T \mathbf{C} \mathbf{x} = 0$ . Such a conic can be easily obtained from the image by applying a robust conic fitting algorithm [60].

By removing the effect of  $\mathbf{K}$ , the conic  $\mathbf{C}$  will transform into

$$\hat{\mathbf{C}} = \mathbf{K}^T \mathbf{C} \mathbf{K} \quad (4.1)$$

which is a conic expressed in image plane coordinates. It has been shown in [104] that, by eigen decomposition,  $\hat{\mathbf{C}}$  can be factorized into

$$\hat{\mathbf{C}} = \mathbf{R} \mathbf{D} \mathbf{R}^T, \quad (4.2)$$

where  $\mathbf{R}$  is the rotation matrix of the camera, and  $\mathbf{D} = \text{diag}(\lambda_1, \lambda_1, \lambda_2)$  is a diagonal matrix composed of the eigenvalues of  $\hat{\mathbf{C}}$ . Note that  $\mathbf{D}$  represents a circle of radius  $r_c = \sqrt{-\frac{\lambda_1}{\lambda_2}}$  centred at the image plane origin. It corresponds to the image of the sphere when the camera has its optical axis aligned with the  $Z$ -axis of the world coordinate system (i.e., when  $\mathbf{R} = \mathbf{I}$ ). The sphere centre, expressed in camera coordinates, can be recovered as  $d\mathbf{r}_3$ , where  $\mathbf{r}_3$  is the third column of  $\mathbf{R}$  and  $d = r_s \sqrt{\frac{1+r_c^2}{r_c}}$ .

With a calibrated camera and a sphere with known radius, the position of the sphere can be uniquely recovered from its image. The scene can be reconstructed from its reflections on the sphere placed at two distinct positions. When the radius of the sphere is not known, the position of the sphere can still be recovered up to an unknown scale determined by this unknown radius, and this results in a reconstruction up to the same unknown scale.

## 4.4 Estimating Camera Intrinsic Parameters

In this section, we first derive an analytical solution for recovering the focal length of a camera from an image of a sphere under a known principal point of the camera. Based on this analytical solution, we introduce two robust algorithms for estimating both the principal point and focal length of the camera from multiple images of the sphere as well as from just one image of the sphere.

### 4.4.1 Focal Length

Assume the camera has unit aspect ratio, and let  $f$  and  $(u_0, v_0)$  be its focal length and principal point respectively. The camera calibration matrix  $\mathbf{K}$  can be factorized into

$$\mathbf{K} = \mathbf{T}\mathbf{F}, \quad (4.3)$$

where

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & u_0 \\ 0 & 1 & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ and } \mathbf{F} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.4)$$

Substituting (4.3) into (4.1) gives

$$\hat{\mathbf{C}} = \mathbf{F}^T \mathbf{T}^T \mathbf{C} \mathbf{T} \mathbf{F}. \quad (4.5)$$

Now suppose the principal point of the camera is known (e.g., by assuming the principal point is at the image centre). The effect of  $\mathbf{T}$  can be removed by translating all the points by  $(-u_0, -v_0)$ . After the translation, the conic  $\mathbf{C}$  will transform into

$$\bar{\mathbf{C}} = \mathbf{T}^T \mathbf{C} \mathbf{T} \quad (4.6)$$

which is a conic having the same shape as  $\mathbf{C}$  (see Fig. 4.2).

Substituting (4.6) into (4.5) gives

$$\hat{\mathbf{C}} = \mathbf{F}^T \bar{\mathbf{C}} \mathbf{F}. \quad (4.7)$$

Now let

$$\hat{\mathbf{C}} = \begin{bmatrix} \hat{\omega}_{11} & \hat{\omega}_{12} & \hat{\omega}_{13} \\ \hat{\omega}_{21} & \hat{\omega}_{22} & \hat{\omega}_{23} \\ \hat{\omega}_{31} & \hat{\omega}_{32} & \hat{\omega}_{33} \end{bmatrix} \text{ and } \bar{\mathbf{C}} = \begin{bmatrix} \bar{\omega}_{11} & \bar{\omega}_{12} & \bar{\omega}_{13} \\ \bar{\omega}_{21} & \bar{\omega}_{22} & \bar{\omega}_{23} \\ \bar{\omega}_{31} & \bar{\omega}_{32} & \bar{\omega}_{33} \end{bmatrix},$$

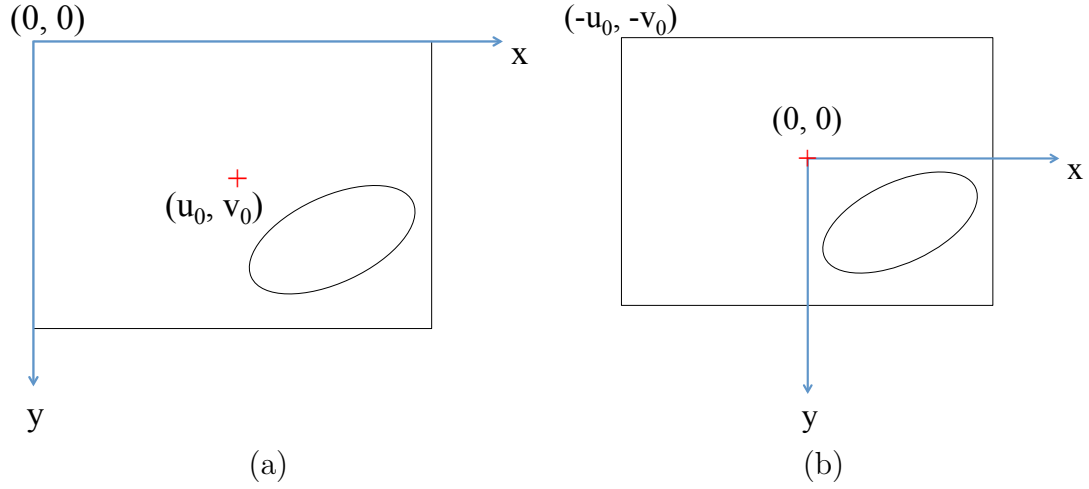


Fig. 4.2 (a)  $\mathbf{C}$ : before removing the effect of  $\mathbf{T}$ . (b)  $\bar{\mathbf{C}}$ : after removing the effect of  $\mathbf{T}$ .

and from (4.7) we have

$$\begin{bmatrix} \hat{\omega}_{11} & \hat{\omega}_{12} & \hat{\omega}_{13} \\ \hat{\omega}_{21} & \hat{\omega}_{22} & \hat{\omega}_{23} \\ \hat{\omega}_{31} & \hat{\omega}_{32} & \hat{\omega}_{33} \end{bmatrix} = \begin{bmatrix} \bar{\omega}_{11}f^2 & \bar{\omega}_{12}f^2 & \bar{\omega}_{13}f \\ \bar{\omega}_{21}f^2 & \bar{\omega}_{22}f^2 & \bar{\omega}_{23}f \\ \bar{\omega}_{31}f & \bar{\omega}_{32}f & \bar{\omega}_{33} \end{bmatrix} \quad (4.8)$$

Recall that  $\hat{\mathbf{C}}$  can be factorized into  $\hat{\mathbf{C}} = \mathbf{RDR}^T$ , where  $\mathbf{D} = \text{diag}(\lambda_1, \lambda_1, \lambda_2)$  is a diagonal matrix composed of the eigenvalues of  $\hat{\mathbf{C}}$ . The eigenvalues of  $\hat{\mathbf{C}}$  can be obtained by solving the characteristic equation

$$\det(\hat{\mathbf{C}} - \lambda\mathbf{I}) = 0. \quad (4.9)$$

Note that (4.9) gives a cubic polynomial

$$\lambda^3 - \beta\lambda^2 + \gamma\lambda + \delta = 0, \quad (4.10)$$

where

$$\begin{aligned}
\beta &= \hat{\omega}_{11} + \hat{\omega}_{22} + \hat{\omega}_{33} \\
\gamma &= \hat{\omega}_{11}\hat{\omega}_{22} + \hat{\omega}_{11}\hat{\omega}_{33} + \hat{\omega}_{22}\hat{\omega}_{33} - \hat{\omega}_{12}^2 - \hat{\omega}_{13}^2 - \hat{\omega}_{23}^2 \\
\delta &= \hat{\omega}_{11}\hat{\omega}_{23}^2 + \hat{\omega}_{22}\hat{\omega}_{13}^2 + \hat{\omega}_{33}\hat{\omega}_{12}^2 - \hat{\omega}_{11}\hat{\omega}_{22}\hat{\omega}_{33} - 2\hat{\omega}_{12}\hat{\omega}_{13}\hat{\omega}_{23}
\end{aligned} \tag{4.11}$$

Since  $\hat{\mathbf{C}}$  has at least two identical eigenvalues, (4.10) must have at least two equal roots. Hence we have [112]

$$\mu^2 - 4\nu^3 = 0, \tag{4.12}$$

where

$$\begin{aligned}
\mu &= \beta^2 - 3\gamma \\
&= \frac{1}{2}[(\hat{\omega}_{11} - \hat{\omega}_{22})^2 + (\hat{\omega}_{11} - \hat{\omega}_{33})^2 + (\hat{\omega}_{22} - \hat{\omega}_{33})^2] + 3(\hat{\omega}_{12}^2 + \hat{\omega}_{13}^2 + \hat{\omega}_{23}^2) \\
\nu &= 2\beta^2 - 9\beta\gamma - 27\delta \\
&= 18(\hat{\omega}_{11}\hat{\omega}_{22}\hat{\omega}_{33} + 3\hat{\omega}_{12}\hat{\omega}_{13}\hat{\omega}_{23}) + 2(\hat{\omega}_{11}^2 + \hat{\omega}_{22}^2 + \hat{\omega}_{33}^2) \\
&\quad + 9(\hat{\omega}_{11} + \hat{\omega}_{22} + \hat{\omega}_{33})(\hat{\omega}_{12}^2 + \hat{\omega}_{13}^2 + \hat{\omega}_{23}^2) - 3(\hat{\omega}_{11} + \hat{\omega}_{22})(\hat{\omega}_{11} + \hat{\omega}_{33})(\hat{\omega}_{22} + \hat{\omega}_{33}) \\
&\quad - 27(\hat{\omega}_{11}\hat{\omega}_{23}^2 + \hat{\omega}_{22}\hat{\omega}_{13}^2 + \hat{\omega}_{33}\hat{\omega}_{12}^2)
\end{aligned} \tag{4.13}$$

Given the conic  $\mathbf{C}$  and the principal point  $(u_0, v_0)$ , the only unknown in (4.12) is  $f$ . Solving (4.12) leads to 12 solutions. We observe that among these 12 solutions, 8 of them are zeros, and the remaining 4 have an identical absolute value, which gives the solution of  $f$ . Under noisy data, the solutions to (4.12) might be complex numbers. The 8 zero solutions become pure imaginary numbers, and the remaining 4 become two pairs of conjugate complex numbers, with opposite signs for their real and imaginary parts (i.e.,  $a + bi$ ,  $a - bi$ ,  $-a + bi$ ,  $-a - bi$ ). The absolute value of their real parts gives us an estimate of the focal length, i.e.,  $f = \|a\|$ .

#### 4.4.2 Principal Point

As discussed above, a unique focal length  $f$  can be obtained by solving (4.12) when the principal point of the camera is known or assumed to locate at the image centre.

In practice, however, the principal point is often not known a priori. Besides, due to manufacture reasons, the principal point usually does not locate exactly at the image centre, but somewhere close to it.

We notice that given the ground truth principal point  $(u_0, v_0)$ , the focal length estimated by (4.12) should be identical to the ground truth focal length  $f_{gt}$ . On the other hand, if we solve (4.12) using points close to the ground truth principal point as the principal point, the estimated focal length will become a complex number. The real part of this complex number is close to  $f_{gt}$ , and its imaginary part is a relatively small value to compensate the errors.

Another observation is that, given the conic images of the mirror sphere at two distinct positions, we can obtain two estimates of the focal length, denoted as  $f_1$  and  $f_2$  respectively, using the same assumed principal point.  $f_1 = f_2 = f_{gt}$  holds when the assumed principal point equals the ground truth principal point.

Based on the above observations, we propose to estimate the principal point by minimizing the difference between the focal lengths estimated from the conic images of the mirror sphere at two distinct positions, subject to the principal point lying within a small window centred at the image centre. The difference between the estimated focal lengths is measured as the sum of the magnitudes of the differences in their real and imaginary parts respectively

$$error = ||\text{real}(f_1)| - |\text{real}(f_2)|| + ||\text{imag}(f_1)| - |\text{imag}(f_2)||. \quad (4.14)$$

The proposed approach is summarized in Algorithm 1.

The above algorithm requires two conic images of the mirror sphere. When only one image of the sphere is available, one simple strategy is to assume the principal point is at the image centre and then estimate  $f$  by solving (4.12). This, however, often does not give an optimal solution as the principal point, as mentioned before, usually does not locate exactly at the image centre. Based on the observation that the error in the estimated focal length is highly correlated with the error in the position

---

**Algorithm 1:** Estimation of the principal point and focal length from two conic images of a mirror sphere.

---

**Input** : Image centre  $(u_c, v_c)$ , conic images  $\mathbf{C}_1, \mathbf{C}_2$   
**Output:** Principal point  $(u_0, v_0)$ , focal length  $f$   
Initialization: Set offsets along  $u, v$  directions as  $w = \text{const1}, h = \text{const2}$ ; set step size as  $s = \text{const3}$ ; set error as  $\text{error}_{\min} = \text{large const}$ ;  
**for**  $u_p \leftarrow u_c - w$  **to**  $u_c + w$  **step**  $s$  **do**  
    **for**  $v_p \leftarrow v_c - h$  **to**  $v_c + h$  **step**  $s$  **do**  
        Construct  $\mathbf{T}$  using (4.4) with  $(u_p, v_p)$ ;  
        Compute  $\bar{\mathbf{C}}_1, \bar{\mathbf{C}}_2$  using (4.6);  
        Construct  $\hat{\mathbf{C}}_1, \hat{\mathbf{C}}_2$  from  $\bar{\mathbf{C}}_1, \bar{\mathbf{C}}_2$  using (4.7);  
        Solve (4.12) for  $\hat{\mathbf{C}}_1, \hat{\mathbf{C}}_2$  to obtain  $f_1, f_2$ ;  
        Compute current error,  $\text{error}_{\text{cur}}$ , by (4.14);  
        **if**  $\text{error}_{\text{cur}} < \text{error}_{\min}$  **then**  
             $u_0 \leftarrow u_p$   
             $v_0 \leftarrow v_p$   
             $f \leftarrow \frac{|\text{real}(f_1)| + |\text{real}(f_2)|}{2}$   $\text{error}_{\min} \leftarrow \text{error}_{\text{cur}}$   
        **end**  
    **end**  
**end**

---

of the principal point, we propose a novel approach for estimating both the principal point and focal length of the camera from just one image of the mirror sphere. We first sample points evenly within a small window centred at the image centre, and estimate a focal length  $f$  using each sample point as the principal point. We then calculate the mean of these estimated values and this gives us a final estimate of the focal length. Next, we identify the sample point that leads to an estimated focal length closest to the mean value as the principal point. Fig. 4.3 gives an example of estimating the focal length using points sampled around the image centre as the principal point. We can see that when the sampled principal point is closer to the ground truth principal point, the estimated  $f$  will be closer to the ground truth focal length  $f_{gt}$ .

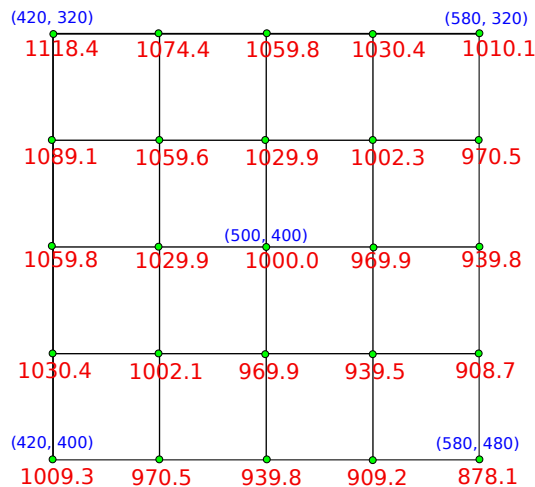


Fig. 4.3 An example of camera intrinsics estimation from the conic of a single sphere. The ground truth intrinsic parameters are  $f = 1000$ ,  $(u_0, v_0) = (500, 400)$ . Given the conic of the sphere, we estimated the focal length by setting different image points as the principal point. The coordinates (blue) stand for the pixel coordinates, and the number (red) close to a pixel stands for the estimated  $f$  by setting that pixel as the principal point. The mean of the estimated values is  $f_m = 998.1$ .  $(500, 400)$  is the sample point that leads that to an estimated  $f$  closest to  $f_m$ .

## 4.5 Shape Recovery

After estimating the principal point  $(u_0, v_0)$  and focal length  $f$  of the camera using the methods described in the Section 4.4, we can obtain the camera calibration matrix  $\mathbf{K}$  using (4.3) and (4.4). We use  $\mathbf{K}$  to transform the conic image  $\mathbf{C}$  into  $\hat{\mathbf{C}}$  using (4.1), and factorize it into  $\hat{\mathbf{C}} = \mathbf{RDR}^T$  by eigen decomposition. The centre of the sphere, expressed in camera coordinates, can be recovered as  $d\mathbf{r}_3$  where  $\mathbf{r}_3$  denotes the third column of  $\mathbf{R}$  resulting from the eigen decomposition, and  $d = r_s \sqrt{\frac{1+r_s^2}{r_c}}$  denotes the distance of the sphere centre from the camera centre. With the radius  $r_s$  of the sphere being unknown, we can simply set  $r_s$  to 1 and the resulting reconstruction will then be scaled by  $\frac{1}{r_s}$ .

Consider a scene point  $\mathbf{P}$ , and let  $\mathbf{q}_1$  and  $\mathbf{q}_2$  be its reflections observed on the surface of a mirror sphere placed at  $\mathbf{S}_1$  and  $\mathbf{S}_2$ , respectively (see Fig. 4.4). To reconstruct  $\mathbf{P}$  (in camera coordinates), we first construct the visual rays  $\mathbf{V}(\mathbf{q}_1)$  and  $\mathbf{V}(\mathbf{q}_2)$  for  $\mathbf{q}_1$  and  $\mathbf{q}_2$ ,



respectively, using the formula  $\mathbf{V}(\mathbf{q}) = \mathbf{K}^{-1}\mathbf{q}$ . We solve for the point of intersection  $\mathbf{Q}_1$  between  $\mathbf{V}(\mathbf{q}_1)$  and the sphere at  $\mathbf{S}_1$ , and the point of intersection  $\mathbf{Q}_2$  between  $\mathbf{V}(\mathbf{q}_2)$  and the sphere at  $\mathbf{S}_2$ , respectively. Based on the law of reflection, the incident rays at  $\mathbf{Q}_1$  and  $\mathbf{Q}_2$  can be constructed, and  $\mathbf{P}$  can be recovered by triangulating these incident rays. In the case where  $\mathbf{P}$  can be directly observed by the camera, only one reflection of  $\mathbf{P}$  on the surface of a mirror sphere is sufficient to recover  $\mathbf{P}$  by triangulating the visual ray of the direct observation with the incident ray of the reflection observed.

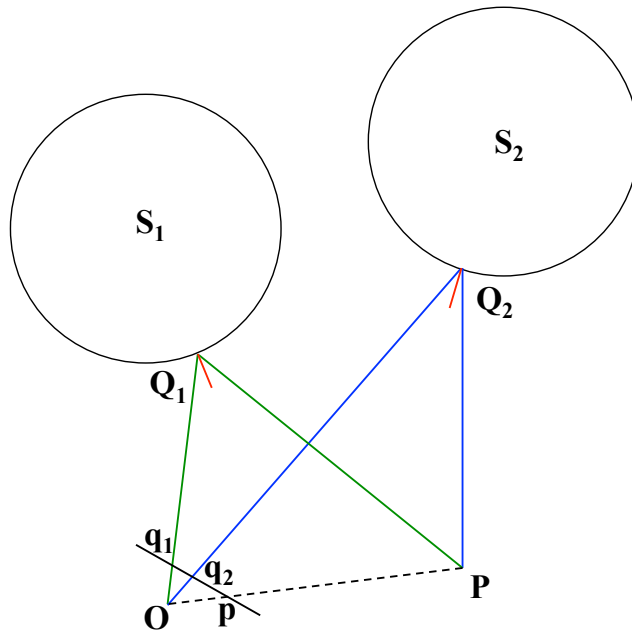


Fig. 4.4 A perspective camera located at  $\mathbf{O}$  observes the reflections of a scene point  $\mathbf{P}$  at  $\mathbf{Q}_1$  and  $\mathbf{Q}_2$  on the surface of a mirror sphere placed at  $\mathbf{S}_1$  and  $\mathbf{S}_2$ , respectively.  $\mathbf{P}$  can be reconstructed by triangulating: (a)  $\mathbf{Q}_1\mathbf{P}$  with  $\mathbf{Q}_2\mathbf{P}$ , if  $\mathbf{P}$  is not visible; (b)  $\mathbf{OP}$  with  $\mathbf{Q}_1\mathbf{P}$  (or  $\mathbf{Q}_2\mathbf{P}$ ), if  $\mathbf{P}$  is visible.

## 4.6 Experimental Results

We evaluate our proposed methods on both synthetic and real data. We compare our uncalibrated approach against the method in [16] which assumes an internally calibration camera.

### 4.6.1 Synthetic Data

We used the same synthetic data set as in [16] to evaluate our approach and compare against their method. A synthetic perspective camera was employed to observe a mirror sphere with radius  $1.5mm$  placed at two distinct positions. Four 3D points  $\mathbf{P}_{\{1,2,3,4\}}$  were reflected at  $\mathbf{Q}_{\{11,12,13,14\}}$  on the surface of the sphere when it was placed at  $\mathbf{S}_1$ , and at  $\mathbf{Q}_{\{21,22,23,24\}}$  on the surface of the sphere when it was placed at  $\mathbf{S}_2$ .  $\mathbf{P}_1\mathbf{P}_2$ ,  $\mathbf{P}_1\mathbf{P}_3$  and  $\mathbf{P}_1\mathbf{P}_4$  were mutually orthogonal to each other, and they were of the same lengths ( $5\text{ mm}$ ).

We applied SVD to fit conics to the edges extracted from the images of the sphere [60]. To evaluate the robustness of our approach, we added Gaussian noise to the pixel coordinates of the edges with noise level ranging from 0 to 3.0 pixels. With conics fitted to two images of the mirror sphere, the camera calibration matrix  $\mathbf{K}$  can be estimated using our method described in Section 4.4.

With an unknown radius of the sphere, we can only reconstruct the scene up to an unknown scale. Hence, it is more meaningful to compare the reconstruction error in terms of length ratios and angles rather than absolute distance errors. Let  $l_1$ ,  $l_2$ , and  $l_3$  denote  $\|\mathbf{P}_1\mathbf{P}_2\|$ ,  $\|\mathbf{P}_1\mathbf{P}_3\|$  and  $\|\mathbf{P}_1\mathbf{P}_4\|$ , respectively. We measured the errors in the length ratios  $l_1/l_2$ ,  $l_2/l_3$  and  $l_3/l_1$ , respectively, as their deviations from 1. Similarly, let  $\alpha$ ,  $\beta$  and  $\theta$  denote  $\angle\mathbf{P}_2\mathbf{P}_1\mathbf{P}_3$ ,  $\angle\mathbf{P}_3\mathbf{P}_1\mathbf{P}_4$  and  $\angle\mathbf{P}_4\mathbf{P}_1\mathbf{P}_2$ , respectively. The errors in  $\alpha$ ,  $\beta$  and  $\theta$  were measured as their deviations from  $90^\circ$ . We performed 500 independent trials and our reconstruction results as well as the comparison with [16] are presented in Fig. 4.5. Generally, the errors increase linearly with the noise level. Algorithm 1 performed slightly better than the method using only a single image of the sphere, hence we used the intrinsics estimated by Algorithm 1 for 3D reconstruction in our experiment. Our length ratio errors and angle errors are quite close to that of [16]. Fig. 4.6 shows our reconstruction under the noise level  $\sigma = 2.0$ . The results demonstrate the accuracy and robustness of our proposed approach.

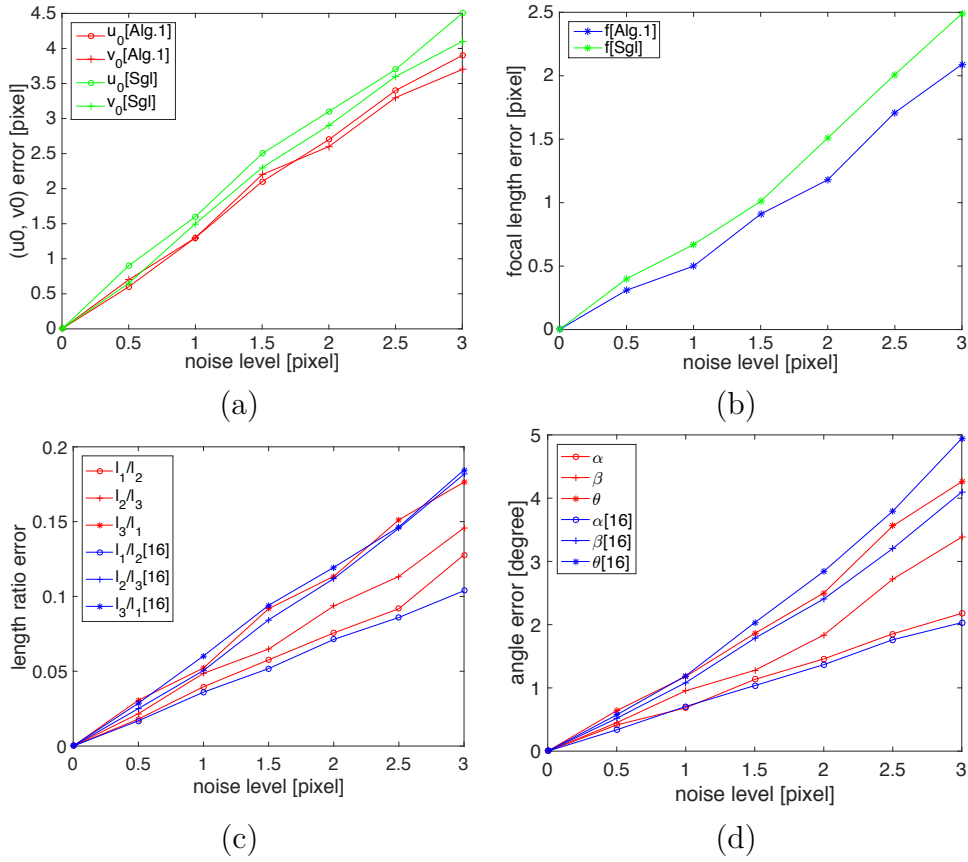


Fig. 4.5 Synthetic experiment results under different noise levels. Ground truth value:  $f = 1200$ ,  $u_0 = 495$ ,  $v_0 = 395$ . (a) Reconstructed principal point  $(u_0, v_0)$  errors by Algorithm 1 [Alg.1], and by the method using only a single image of the sphere [Sgl]. (b) Focal length errors; (c) Length ratio errors compared with [16]; (d) Angle errors of the reconstructed rays compared with [16].

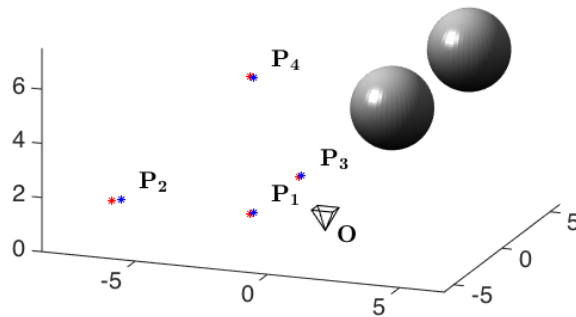


Fig. 4.6 Reconstruction under noise level  $\sigma = 2.0$ . Blue points: ground truth; Red points: reconstruction.

### 4.6.2 Real Data

To evaluate our approach on real data set, we performed an experiment on a rectangular box with a dimension of  $6\text{ cm} \times 6\text{ cm} \times 8\text{ cm}$ . In the experiment, the box was reflected by a mirror sphere of radius  $40\text{ mm}$  placed at four different positions. The reflections on the sphere surface were captured using a *Canon EOS 40D* equipped with a  $24\text{-}70\text{ mm}$  lens. Fig. 4.7 shows our experiment setup.

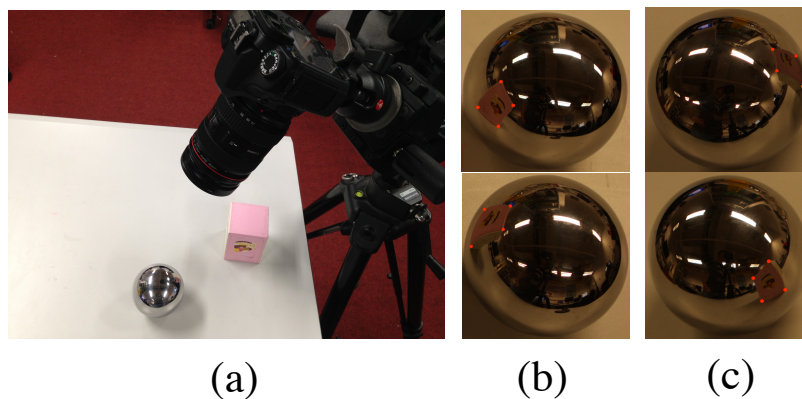


Fig. 4.7 Real experiment setup (a) and reflections on a sphere at four distinct positions (b-c). Four corners of a box are reconstructed using the reflections in (b) and the other four corners are reconstructed using the reflections in (c). The correspondences are marked with red dots in the images. Note that the box is not visible by the camera.

After fitting conics to the images of the sphere using SVD, we first estimated the principal point and focal length of the camera using the methods introduced in Section 4.4. Table 4.1 shows our estimation result. It can be seen that the estimated camera intrinsic parameters are very close to those obtained by camera calibration using a calibration pattern [81]. We used the intrinsics estimated by Algorithm 1 to reconstruct the corners of the box, and measured 24 angles and 12 length ratios around them. We compared these measurements against the ground truth values ( $90^\circ$  for angle and 1.25 for length ratio). We also compared our results with that of [16], which works under an internally calibrated camera. The RMS errors are given in Table 4.2. Fig. 4.8 shows the reconstructed 3D corner points of our method and that of [16]. The recovered surfaces are presented in Fig. 4.9. Our approach achieved a

high accuracy which is very close to that using a calibrated camera.

Table 4.1 Estimation of camera intrinsic parameters. **[Centre]**: results by setting image centre as the principal point; **[Alg.1]**: results by Algorithm 1; **[Sgl]**: results by the method using only a single image of the sphere.

	$f$	$u_0$	$v_0$
<i>Calibration</i> [81]	4435.36	1963.0	1277.0
<i>Estimation</i> [Centre]	4117.49	1944.0	1296.0
<i>Estimation</i> [Alg.1]	4386.06	1955.0	1285.0
<i>Estimation</i> [Sgl]	4301.02	1980.0	1290.0
<i>Error</i> [Centre]	7.17%	0.97%	1.49%
<i>Error</i> [Alg.1]	<b>1.11%</b>	<b>0.41%</b>	<b>0.63%</b>
<i>Error</i> [Sgl]	3.06%	0.87%	1.02%

Table 4.2 RMS angle error and length ratio error of the recovered box. The ground truth angle and length ratio are  $90^\circ$  and 1.25, respectively.

	<i>angle</i>	<i>length ratio</i>
<i>Calibrated</i> [16]	1.08	0.03
<i>Ours</i>	1.05	0.03

## 4.7 Discussions and Conclusions

This chapter addresses the problem of single view 3D reconstruction under an uncalibrated camera and an unknown mirror sphere. We derive an analytical solution to solve the camera focal length given the principal point by enforcing a repeated eigenvalue constraint for the conic image of the mirror sphere. Based on this analytical solution, we introduce a robust algorithm to estimate both the principal point and focal length of the camera by minimizing the difference between focal lengths estimated from multiple images of the sphere. Besides, we also present a novel approach to estimate both the principal point and focal length of the camera in the case of just one image of the sphere. With the estimated camera intrinsic parameters, we can recover the sphere position(s) by eigen decomposition, and reconstruct the scene up to an unknown scale determined by the radius of the sphere. Experimental results on

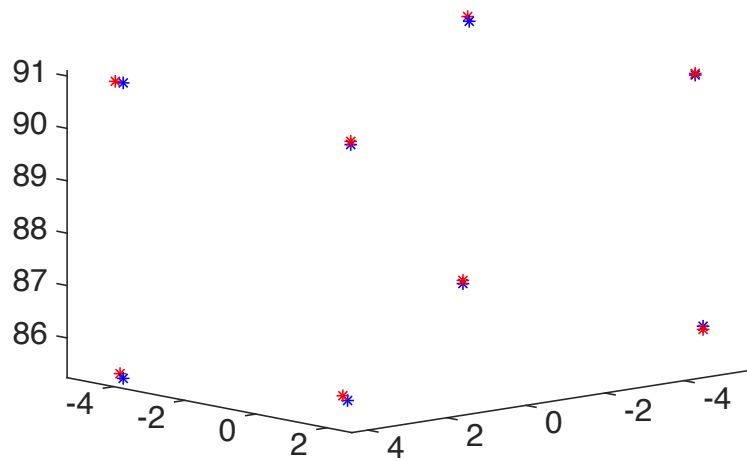


Fig. 4.8 Reconstructed corners of a box. Blue: using a calibrated camera [16]; Red: using an uncalibrated camera (ours).

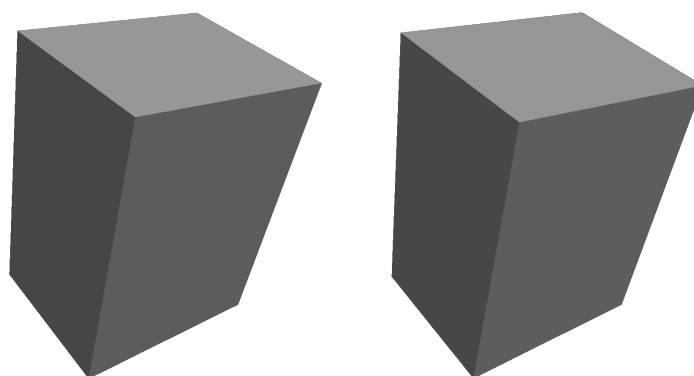


Fig. 4.9 Recovered surfaces. Left: using a calibrated camera [16]; Right: using an uncalibrated camera (ours).

---

both synthetic and real data demonstrate the feasibility and accuracy of our proposed approach. One limitation of our approach is the difficulty in establishing dense and high quality reflection correspondences due to the distortions exhibited in the reflected images. As a result, the reconstructed point cloud is sparse. Note that this is in fact the limitation of all 3D reconstruction methods based on mirror spheres. In the future, we would like to extend our work to achieve dense 3D reconstruction.





# Chapter 5

## Learning Semantic Correspondence

### 5.1 Introduction

Our goal in this chapter is to establish *semantic correspondences* across images that contain different instances of the same object or scene category. Such images feature much larger changes in appearance and spatial layout than pictures of the *same* scene used in *stereo vision*, which we take here to include broadly not only classical (narrow-baseline) stereo fusion (e.g., [113, 114]), but also optical flow computation (e.g., [115–117]) and wide-baseline matching (e.g., [118, 119]). Due to such a large degree of variations, the problem of semantic correspondence remains very challenging. Most previous approaches to semantic correspondence [119–124] focus on combining an effective spatial regularizer with hand-crafted features such as SIFT [18], DAISY [125] or HOG [19]. With the remarkable successes of deep learning approaches in visual recognition, several learning-based methods have also been proposed for both stereo vision [126–129] and semantic correspondence [130–132]. Yet, none of these methods exploits geometric consistency constraint that has proven to be a key factor to the success of their hand-crafted counterparts. Geometric regularization, if any, often occurs during post-processing but not during learning (e.g., [128, 129]).

In this chapter we propose a convolutional neural network (CNN) architecture, called *SCNet*, for learning geometrically plausible semantic correspondence (Fig. 5.1).

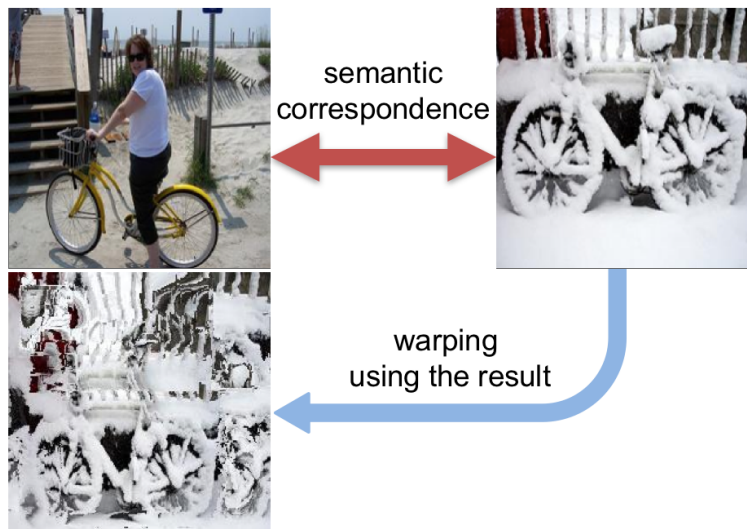


Fig. 5.1 Learning semantic correspondence. We propose a convolutional neural network, SCNet, to learn semantic correspondence using both appearance and geometry. This allows us to handle a large degree of intra-class and scene variations. This figure shows a pair of input images (top) and a warped image (bottom) using its semantic correspondence by our method.

Following the *proposal flow* approach to semantic correspondence of Ham *et al.* [133], we use object proposals [134–136] as matching primitives, and explicitly incorporate the geometric consistency of these proposals in our loss function. Unlike [133] with its hand-crafted features, however, we train our system in an end-to-end manner using image pairs extracted from the PASCAL VOC 2007 keypoint dataset [137]. A comparative evaluation on several standard benchmarks demonstrates that the proposed approach substantially outperforms both recent deep architectures and previous methods based on hand-crafted features.

Our main contributions can be summarized as follows:

- We introduce a simple and efficient model for learning to match regions using both appearance and geometry.
- We propose a convolutional neural network, SCNet, to learn semantic correspondence with region proposals.
- We achieve state-of-the-art results on several benchmarks, clearly demonstrating

the advantage of learning both appearance and geometric terms.

## 5.2 Related Work

Here we briefly describe representative approaches related to semantic correspondence.

**Semantic correspondence.** SIFT Flow [123] extends classical optical flow to establish correspondences across similar but different scenes. It uses dense SIFT descriptors to capture semantic information beyond naive color values, and leverages a hierarchical optimization technique in a coarse-to-fine pipeline for efficiency. Kim *et al.* [122] and Hur *et al.* [121] proposed more efficient generalizations of SIFT Flow. Instead of using SIFT features, Yang *et al.* [119] used DAISY [125] for an efficient descriptor extraction. Inspired by an exemplar-LDA approach [138], Bristow *et al.* [120] used whitened SIFT descriptors, making semantic correspondence robust to background clutter. Recently, Ham *et al.* [133] introduced proposal flow that uses object proposals as matching elements for semantic correspondence that are robust to scale and clutter. This work shows that the HOG descriptor gives better matching performance than deep learning features [139, 140]. Taniar *et al.* [124] also used HOG descriptors, and showed that jointly performing cosegmentation and establishing dense correspondence are helpful in both tasks. Despite differences in feature descriptors and optimization schemes, these semantic correspondence approaches use a spatial regularizer to ensure flow smoothness on top of hand-crafted or pre-trained features.

**Deep learning for correspondence.** Recently, CNNs have been applied to classical dense correspondence problems such as optical flow and stereo matching to learn feature descriptors [128, 129, 141] or similarity functions [127, 128, 141]. FlowNet [126] uses an end-to-end scheme to learn optical flow with a synthetic dataset, and several recent approaches also use supervision from reconstructed 3D scenes and stereo pairs [127–129, 141] to learn correspondence. MC-CNN [128] and its efficient extension [129] train CNN models to predict how well two image patches match and use this

information to compute the stereo matching cost. DeepCompare [141] learns a similarity function for patches directly from images of a 3D scene, which allows for various types of geometric and photometric transformations (e.g., rotation and illumination changes). These approaches are inherently limited to matching images of the same physical object/scene. In contrast, Long *et al.* [142] used CNN features pre-trained on the ImageNet classification tasks (due to a lack of available datasets for learning semantic correspondence) for semantic correspondence estimation and achieved performance comparable to SIFT flow. To overcome the difficulty in obtaining ground truth for semantic correspondence, Zhou *et al.* [132] leveraged 3D models, and used flow consistency between 3D models and 2D images as a supervisory signal to train a CNN. Another approach to generating ground truth is to directly augment the data by densifying sparse keypoint annotations using warping [133, 143]. The universal correspondence network (UCN) of Choy *et al.* [130] learns semantic correspondence using an architecture similar to [129], but adds a convolutional spatial transformer network for improved robustness to rotation and scale changes. Kim *et al.* [131] introduced a convolutional descriptor using self-similarity, called fully convolutional self-similarity (FCSS), and combined the learned semantic descriptors with the proposal flow [133] framework. These approaches to learning semantic correspondence [130, 132] or semantic descriptors [131] typically perform better than traditional hand-crafted ones. Unlike our method, however, they do not incorporate geometric consistency between regions or object parts in the learning process.

### 5.3 Our Approach

We consider the problem of learning to match regions with arbitrary positions and sizes in pairs of images. This setting is general enough to cover all cases of region sampling used in semantic correspondence, including sampling a dense set of regular local regions as in typical dense correspondence [120, 122, 123, 144] as well as employing multi-scale object proposals [134–136, 145, 146]. In this work, following proposal flow [133], we

focus on establishing correspondences between object proposal boxes.

### 5.3.1 Model

Our basic model for matching is based on the probabilistic Hough matching (PHM) approach [133, 147]. Given a potential match  $m$  between two regions, and the supporting data  $D$  (a set of potential matches), the PHM model can be written as

$$\begin{aligned} P(m|D) &= \sum_x P(m|x, D)P(x|D) \\ &= P_a(m) \sum_x P_g(m|x)P(x|D), \end{aligned} \quad (5.1)$$

where  $x$  is the offset (*e.g.*, position and scale change) between two regions  $r$  and  $s$  in a potential match  $m = [r, s]$  in  $D$ .  $P_a(m)$  and  $P_g(m|x)$  are the probabilities that the match is correct based on appearance only, and based on geometry computed using the offset  $x$  only, respectively.<sup>1</sup> PHM computes a matching score by replacing geometry prior  $P(x|D)$  with the Hough voting  $h(x|D)$  [147]:

$$h(x|D) = \sum_{m' \in D} P_a(m')P_g(m'|x). \quad (5.2)$$

This turns out to be an effective spatial matching model that combines appearance similarity with global geometric consistency measured by letting all matches vote on the potential offset  $x$  [133, 147].

In our learning framework, we consider similarities rather than probabilities, and rewrite the PHM score for the match  $m$  as

$$\begin{aligned} z(m, w) &= f(m, w) \sum_x g(m, x) \sum_{m' \in D} f(m', w)g(m', x) \\ &= f(m, w) \sum_{m' \in D} [\sum_x g(m, x)g(m', x)]f(m', w), \end{aligned} \quad (5.3)$$

where  $f(m, w)$  is a parameterized appearance similarity function between the two

---

<sup>1</sup>We suppose that appearance matching is independent of geometry matching and the offset.

regions in the potential match  $m$ ,  $x$  is as before an offset variable (position plus scale), and  $g(m, x)$  measures the geometric compatibility between the match  $m$  and the offset  $x$ .

Now assuming that we have a total number of  $n$  potential matches, and identifying matches with their indices, we can rewrite this score as

$$z(m, w) = f(m, w) \sum_{m'} K_{mm'} f(m', w), \quad (5.4)$$

where  $K_{mm'} = \sum_x g(m, x)g(m', x)$ , and the  $n \times n$  matrix  $K$  is the *kernel matrix* associated with the feature vector  $\varphi(m) = [g(m, x_1), \dots, g(m, x_s)]^T$ , where  $x_1$  to  $x_s$  form the finite set of values that the offset variable  $x$  runs over: indeed  $K_{mm'} = \varphi(m) \cdot \varphi(m')$ .<sup>2</sup>

Given training pairs of images with associated true and false matches, we can learn our similarity function by minimizing with respect to  $w$

$$E(w) = \sum_{m=1}^n l[y_m, z(m, w)] + \lambda \Omega(w), \quad (5.5)$$

where  $l$  is a loss function,  $y_m$  is the the ground-truth label (either 1 [true] or 0 [false]) for the match  $m$ , and  $\Omega$  is a regularizer (e.g.,  $\Omega(w) = \|w\|^2$ ). We use the hinge loss and  $L_2$  regularizer in this work. Finally, at test time, we associate any region  $r$  with the region  $s$  maximizing  $z([r, s], w^*)$ , where  $w^*$  is the set of learned parameters.

### 5.3.2 Similarity Function and Geometry Kernel

There are many possible choices for the function  $f$  that computes the appearance similarity of the two regions  $r$  and  $s$  making up a match  $m$ . Here we assume a trainable embedding function  $c$  (as will be shown later,  $c$  will be the output of a CNN in our case) that outputs a  $L_2$  normalized feature vector. For the appearance similarity

---

<sup>2</sup>Putting it all together in an  $n$ -vector of scores, this can also be rewritten as  $z(w) = f(w) \odot K f(w)$ , where  $z(w) = (z(1, w), \dots, z(n, w))^T$ , “ $\odot$ ” stands for the elementwise product between vectors, and  $f(w) = (f(1, w), \dots, f(n, w))^T$ .

between two regions  $r$  and  $s$ , we then use a rectified cosine similarity:

$$f(m, w) = \max(0, c(r, w) \cdot c(s, w)), \quad (5.6)$$

that sets all negative similarity values to zero, thus making the similarity function sparser as well as insensitive to negative matches during training, with the additional benefit of giving nonnegative weights in (5.3).

Our geometry kernel  $K_{mm'}$  records the fact that two matches (roughly) correspond to the same offset: Concretely, we discretize the set of all possible offsets into bins. Let us denote by  $h$  the function mapping a match  $m$  onto the corresponding bin  $x$ , we now define  $g$  by

$$g(m, x) = \begin{cases} 1, & \text{if } h(m) = x \\ 0, & \text{otherwise.} \end{cases} \quad (5.7)$$

Thus, the kernel  $K_{mm'}$  simply measures whether two matches share the same offset bin or not:

$$K_{mm'} = \begin{cases} 1, & \text{if } h(m) = h(m') \\ 0, & \text{otherwise.} \end{cases} \quad (5.8)$$

In practice,  $x$  runs over a grid of predefined offset values, and  $h(m)$  assigns match  $m$  to the nearest offset point. Our kernel is sparse, which greatly simplifies the computation of the score function in (5.4): Indeed, let  $B_x$  denote the set of matches associated with the bin  $x$ , the score function  $z$  reduces to

$$z(m, w) = f(m, w) \sum_{m' \in B_{h(m)}} f(m', w). \quad (5.9)$$

This trainable form of the PHM model from [133, 147] can be used with (5.5).

Note that since our simple geometry kernel is only dependent on matches' offsets, we obtain the same geometry term value of  $\sum_{m' \in B_{h(m)}} f(m', w)$  for any match  $m$  that falls into the same bin  $h(m)$ . This allows us to compute this geometry term value only once for each non-empty bin  $x$  and then share it for multiple matches in the same bin.

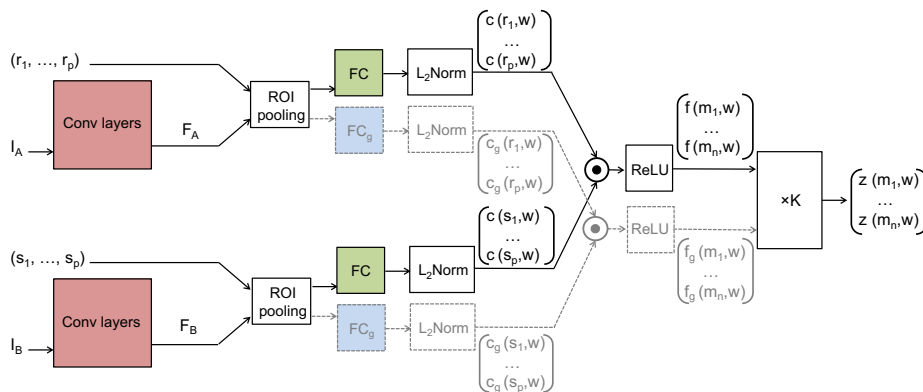


Fig. 5.2 The SCNet architectures. Three variants are proposed: SCNet-AG, SCNet-A, and SCNet-AG+. The basic architecture, SCNet-AG, is drawn in solid lines. Colored boxes represent layers with learning parameters and the boxes with the same color share the same parameters. “ $\times K$ ” denotes the voting layer for geometric scoring. A simplified variant, SCNet-A, learns appearance information only by making the voting layer an identity function. An extended variant, SCNet-AG+, contains an additional stream drawn in dashed lines. SCNet-AG learns a single embedding  $c$  for both appearance and geometry, whereas SCNet-AG+ learns an additional and separate embedding  $c_g$  for geometry.

This sharing makes computing  $z$  several times faster in practice.<sup>3</sup>

### 5.3.3 Gradient-based Learning

The feature embedding function  $c(m, w)$  in the model above can be implemented by any differentiable architecture, for example a CNN-based one, and the score function  $z$  can be learned using stochastic gradient descent. Let us now consider the problem of minimizing the objective function  $E(w)$  defined by (5.5).<sup>4</sup> This requires computing the gradient with respect to  $w$  of the function  $z$ :

$$\nabla z(m, w) = \left[ \sum_{m' \in D} K_{mm'} f(m', w) \right] \nabla f(m, w) + f(m, w) \sum_{m' \in D} K_{mm'} \nabla f(m', w).$$

<sup>3</sup>If the geometry kernel is dependent on something other than offsets, e.g., matches’ absolute position or their neighborhood structure, this sharing is not possible.

<sup>4</sup>We take  $\Omega(w) = 0$  for simplicity in this section, but tackling a nonzero regularizer is easy.



Denoting by  $n$  the size of  $D$ , this involves  $n$  evaluations of both  $f$  and  $\nabla f$ . Computing the full gradient of  $E$  thus requires at most  $n^2$  evaluations of both  $f$  and  $\nabla f$ , which becomes computationally intractable when  $n$  is large enough. The score function of (5.9) with the sparse kernel of (5.8), however, greatly reduces the gradient computation:

$$\nabla z(m, w) = \left[ \sum_{m' \in B_{h(m)}} f(m', w) \right] \nabla f(m, w) + f(m, w) \sum_{m' \in B_{h(m)}} \nabla f(m', w).$$

Note that computing the gradient for match  $m$  involves only a small set of matches falling into the same offset bin  $h(m)$ . More details can be found in Appendix C.

## 5.4 SCNet Architecture

Among many possible architectures implementing the proposed model, we propose using a convolutional neural network (CNN), dubbed *SCNet*, that efficiently processes regions and learns our matching model. Three variants, SCNet-AG, SCNet-A, SCNet-AG+, are illustrated in Fig. 5.2.

In each case, SCNet takes as input two images  $I_A$  and  $I_B$ , and maps them onto feature maps  $F_A$  and  $F_B$  by CNN layers. Given region proposals  $(r_1, \dots, r_p)$  and  $(s_1, \dots, s_p)$  for the two images, parallel ROI pooling layers [148, 149] extract feature maps of the same size for each proposal. This is an efficient architecture that shares convolutional layers over all region proposals.

**SCNet-AG.** The proposal features are fed into a fully-connected layer, mapped onto feature embedding vectors, and normalized into unit feature vectors  $c(r_i, w)$  and  $c(s_j, w)$ , associated with the regions  $r_i$  and  $s_j$  of  $I_A$  and  $I_B$ , respectively. The value of  $f(m, w)$  for the match  $m$  associated with regions  $r_i$  and  $s_j$  is computed as the rectified dot product of  $c(r_i)$  and  $c(s_j)$  (see (5.6)), which defines the appearance similarity  $f(m, w)$  for match  $m$ . Geometric consistency is enforced with the kernel described in Sec. 5.3.2, using a voting layer, denoted as “ $\times K$ ”, that computes score  $z(m, w)$  from the appearance similarity and geometric consensus of proposals. Finally, matching

is performed by identifying the maximal  $z(m, w)$  scores, using both appearance and geometric similarities.

**SCNet-A.** We also evaluate a similar architecture without the geometry term. This architecture drops the voting layer (denoted by  $\times K$  in Fig. 5.2) in SCNet-AG, directly using  $f(m, w)$  as a score function. This is similar to the universal correspondence network (UCN) [130]. The main differences are the use of object proposals and the use of a different loss function.

**SCNet-AG+.** Unlike SCNet-AG, which learns a single embedding  $c$  for both appearance and geometry, SCNet-AG+ learns an additional and separate embedding  $c_g$  for geometry that is implemented by an additional stream in the SCNet architecture (dashed lines in Fig. 5.2). This corresponds to a variant of (5.9), as follows:

$$z^+(m, w) = f(m, w) \sum_{m' \in B_h(m)} f_g(m', w), \quad (5.10)$$

where  $f_g$  is the rectified cosine similarity computed by  $c_g$ . Compared to the original score function, this variant allows the geometry term to learn a separate embedding function for geometric scoring. This may be beneficial particularly when a match’s contribution to the geometric score needs to be different from the appearance score. For example, a match of rigid object parts (wheels of cars) may contribute more to the geometric score than that of deformable object parts (legs of horses). The separate similarity function  $f_g$  allows more flexibility in learning the geometric term.

**Implementation details.** We use the VGG16 [150] model that consists of a set of convolutional layers with  $3 \times 3$  filters, a ReLU layer and a pooling layer.<sup>5</sup> We find that taking the first 4 convolutional layers is a good trade-off for our semantic feature extraction purpose without losing localization accuracy. These layers output features with 512 channels. For example, if the net takes input of  $224 \times 224 \times 3$  images,

<sup>5</sup>Other CNN models can also be adopted.

the convolutional layers produce features with the size of  $14 \times 14 \times 512$ . For the ROI pooling layer, we choose a  $7 \times 7$  filter following the fast R-CNN architecture [148], which produces a feature map with a size of  $7 \times 7 \times 512$  for each proposal. To transform the feature map for each proposal into a feature vector, we use the *FC* layer with a size of  $7 \times 7 \times 512 \times 2048$ . The 2048 dimensional feature vector associated with each proposal are then fed into the  $L_2$  normalization layer, followed by the dot product layer, ReLU, our geometric voting layer, and loss layer. The convolutional layers are initialized by the pretrained weights of VGG16 and the fully connected layers have random initialization. We train our SCNet by mini-batch SGD, with a learning rate of 0.001, and a weight decay of 0.0005. During training, each mini-batch arises from a pair of images associated with a number of proposals. In our implementation, we generated 500 proposals for each image, which leads to  $500 \times 500$  potential matches.

For each mini-batch, we sample matches for training as follows. (1) Positive sampling: For a proposal  $r_i$  in  $I_A$ , we are given its ground truth match  $r'_i$  in  $I_B$ . We pick all the proposals  $s_j$  in  $I_B$  with  $\text{IoU}(s_j, r'_i) > T_{pos}$  to be positive matches for  $r_i$ . (2) Negative sampling: Assume we obtain  $k$  positive pairs w.r.t  $r_i$ . We also need to have  $k$  negative pairs w.r.t  $r_i$ . To achieve this, we first find the proposals  $s_t$  in  $I_B$  with  $\text{IoU}(s_t, r'_i) < T_{neg}$ . Assuming  $p$  proposals satisfying the IoU constraint, we find the proposals with top  $k$  appearance similarity with  $r_i$  among those  $p$  proposals. In our experiment, we set  $T_{pos} = 0.6$ , and  $T_{neg} = 0.4$ .

## 5.5 Experimental Evaluation

In this section we present experimental results and analysis.

### 5.5.1 Experimental Details

**Dataset.** We use the PF-PASCAL dataset that consists of 1300 image pairs selected from PASCAL-Berkeley keypoint annotations<sup>6</sup> of 20 object classes. Each pair of im-

<sup>6</sup><http://www.di.ens.fr/willow/research/proposalflow/>

ages in PF-PASCAL share the same set of non-occluded keypoints. We divide the dataset into 700 training pairs, 300 validation pairs, and 300 testing pairs. The image pairs for training/validation/testing are distributed proportionally to the number of image pairs of each object class. In training, we augment the data into a total of 1400 pairs by horizontal mirroring. We also test our trained models with the PF-WILLOW dataset [133], Caltech-101 [151] and PASCAL Parts [152] to further validate generalization of our models.

**Region proposal.** Unless stated otherwise, we choose to use the method of Manen *et al.* (RP) [134]. The use of RP proposals is motivated by the superior result reported in [133], which is verified once more by our evaluation. In testing we use 1000 proposals for each image as in [133], while in training we use 500 proposals for efficiency.

**Evaluation metric.** We use three metrics to compare the results of SCNet with other methods. First, we use the probability of correct keypoint (PCK) [153], which measures the precision of dense flow at sparse keypoints of semantic relevance. It is calculated on the Euclidean distance  $d(\phi(p), p^*)$  between a warped keypoint  $\phi(p)$  and ground-truth one  $p^*$ <sup>7</sup>. Second, we use the probability of correct regions (PCR) introduced in [133] as an equivalence of the the PCK for region based correspondence. PCR measures the precision of a region matching between region  $r$  and its correspondent  $r^*$  on the intersection over union (IoU) score, which is defined as  $1 - \text{IoU}(\phi(r), r^*)$  [133]. Both metrics are computed against a threshold  $\tau$  in  $[0, 1]$  and we measure  $\text{PCK}@_\tau$  and  $\text{PCR}@_\tau$  as the percentage correct below  $\tau$ . Third, we capture the quality of matching proposals by the mean IoU of the top  $k$  matches ( $\text{mIoU}@k$ ). Note that these metrics are used to evaluate two different types of correspondence. Indeed, PCK is an evaluation metric for dense flow field, whereas PCR and  $\text{mIoU}@k$  are used to evaluate region-based correspondences [133].

---

<sup>7</sup>To better take into account the different sizes of images, we normalize the distance by dividing by the diagonal of the warped image, as in [130].

### 5.5.2 Proposal Flow Components

We use the PF-PASCAL dataset to evaluate region matching performance. This setting allows our method to be tested against three other methods in [133]: NAM, PHM and LOM. NAM finds correspondences using handcrafted features only. PHM and LOM additionally consider global and local geometric consistency, respectively, between region matchings. We also compare our SCNet-learned feature against whitened HOG [19], the best performing handcraft feature of [133].

**Quantitative comparison.** Fig. 5.3(a) compares SCNet methods with the proposal flow methods [133] on the PF-PASCAL dataset. Note that IoU threshold  $\tau$  of PCR was compared against the score of  $1 - \text{IoU}$  following [133], as described in Section 5.5.1. Our SCNet models outperformed the other methods that use the HOG feature. Our geometric models (SCNet-AG, SCNet-AG+) substantially outperformed the appearance-only model (SCNet-A), and SCNet-AG+ slightly outperformed SCNet-AG. This can also be seen from the area under curve (AuC) presented in the legend (in square bracket). This clearly show the effectiveness of deep learned features as well as geometric matching. In this comparison, we fix the VGG16 layer and only learn the FC layers. In our experiment, we also learned all layers including VGG 16 and the FC layers in our model (fully finetuned), but the improvement over the partially learned model was marginal. Fig. 5.3(b) shows the performance of NAM, PHM, LOM matching when replacing HOG feature with our learned feature in SCNet-A. We see that SCNet features greatly improves all the matching methods. Interestingly, LOM using SCNet feature outperformed our best performing SCNet model, SCNet-AG+. However, the LOM method is more than 10 times slower than SCNet-AG+: on average the method took 0.21s for SCNet-A feature extraction and 3.15s for the actual matching process, whereas our SCNet-AG+ only took 0.33s in total. Most of the time in LOM was spent in computing its geometric consistency term. We further evaluated three additional baselines using ImageNet-trained VGG (see Fig. 5.3(c)), namely, VGG, VGG-L2 and VGG-L2-FC. For VGG, we directly use

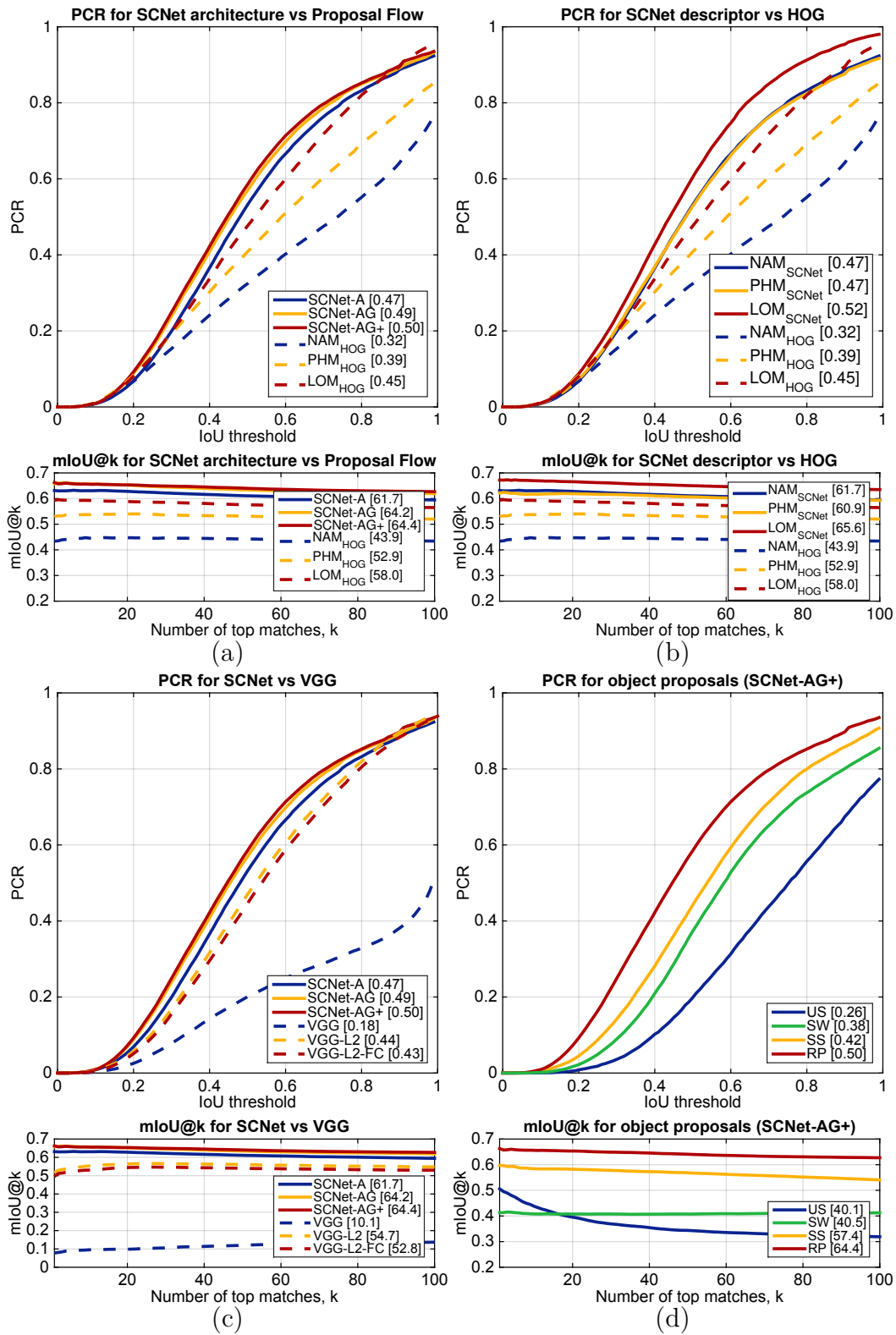


Fig. 5.3 (a) Performance of SCNet on PF-PASCAL, compared to Proposal Flow methods [133]. (b) Performance of SCNet and HOG descriptors on PF-PASCAL, evaluated using Proposal Flow methods [133]. (c) Comparison to ImageNet-trained baselines. (d) Comparison of different proposals. PCR and mIoU@k plots are shown at the top and bottom, respectively. AuC is shown in the legend.

the features from ImageNet-trained VGG, followed by ROI-pooling to make the features for each proposal of the same size ( $7 \times 7 \times 512$ ). We then flatten the features into vectors of dimension 175616. For VGG-L2, we l2-normalized the flattened feature of VGG. For VGG-L2-FC, we performed a random projection from VGG-L2 to a feature of dimension 2048 (the same dimension with SCNet, 12.25 times smaller than VGG and VGG-L2) by adding a randomly initialized FC layer on top of VGG-L2. Note that this is equivalent to SCNet-A without training on the target dataset. The results show that our SCNet approach significantly outperformed the ImageNet-trained baselines.

**Results with different object proposals.** SCNet can be combined with any region proposal methods. In this experiment, we trained and evaluated SCNet-AG+ on PF-PASCAL with four region proposal methods: randomized prim (RP) [134], selective search (SS) [154], random uniform sampling (US), and sliding window (SW). US and SW were extracted using the work of [146], and SW was similar to regular grid sampling used in other popular methods [116, 122, 123]. Fig. 5.3(d) compares the matching performance of using the different proposals in terms of PCR and mIoU@ $k$ . RP performed best, and US performed worst with a large margin. This shows that the region proposal process is an important factor for the matching performance.

**Runtime analysis** Our current (un-optimized) MATLAB implementation takes on average 0.23, 0.32, 0.33 seconds for SCNet-A, SCNet-AG and SCNet-AG+, respectively, on a GeForce GTX TITAN GPU. Table 5.1 shows runtime comparisons.

Table 5.1 Runtime comparison. The time is the mean time cost (second) for each image pair in testing set of PF-PASCAL-RP-1000.

Method	Feature	Match	Total
NAM <sub>HOG</sub> [133]	2.32	0.32	2.64
PHM <sub>HOG</sub> [133]	2.32	1.08	3.40
LOM <sub>HOG</sub> [133]	2.32	3.15	5.47
SCNet-A	-	-	0.23
SCNet-AG	-	-	0.32
SCNet-AG+	-	-	0.33

**Qualitative comparison.** Region matching results for NAM, SCNet-A, and SCNet-AG+ are shown in Fig. 5.4. In this example, at the IoU threshold 0.5, the numbers of correct matches are shown for all methods. We can see that SCNet models performed significantly better than NAM with HOG feature, and SCNet-A was outperformed by SCNet-AG+ that took the geometric consistency term into account in learning.

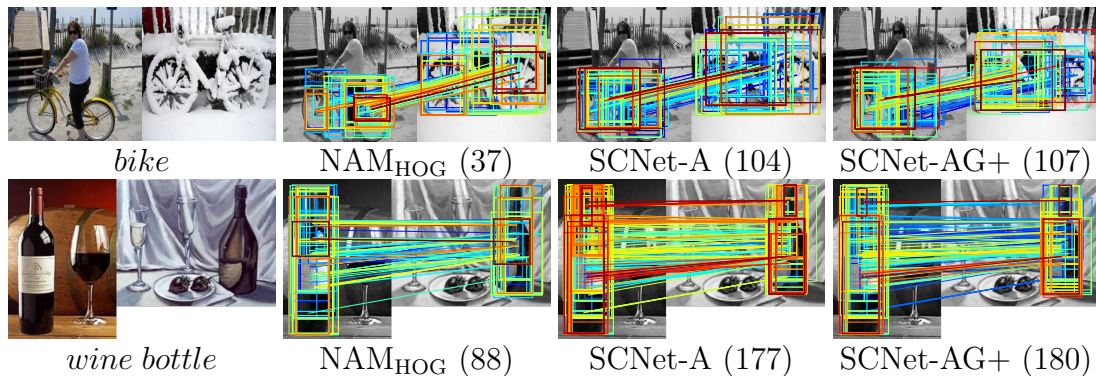


Fig. 5.4 Region matching examples. Numbers beside methods stand for numbers of correct matches.

### 5.5.3 Flow Field

Given a sparse region matching result and its corresponding scores, we generate dense semantic flow using a densifying technique presented in [133]. In brief, we select out a region match with the highest score, and assign dense correspondences to the pixels within the matched regions by linear interpolation. This process is repeated until we assign dense correspondences to all pixels in the source image. The results were evaluated on PF-PASCAL dataset. To evaluate transferability performance of the models, we also tested them on other datasets such as PF-WILLOW [133], Caltech-101 [151] and PASCAL Parts [152] datasets, and compared with state-of-the-art results on these datasets. In these cases direct comparison between learning-based methods may not be fair in the sense that they are trained on different datasets.

**Results on PF-PASCAL.** We compared SCNet with Proposal Flow [133] and UCN [130] on the PF-PASCAL dataset, and summarized the result in Table 5.2.



The UCN was retrained using the code provided by the authors on the PF-PASCAL dataset for fair comparison. Using the raw network of [130] trained on a different subset of PASCAL yielded, as expected, lower performance, with a mean PCK of 36.0 as opposed to the 55.6 obtained for the retrained network. The three variants of SCNet performed consistently better than UCN as well as all methods in [133], with a PCK of 66.3 or above. They also performed better than other methods in the per-class PCK, except sheep and train classes, where the pose variations in the images are smaller than other classes. UCN performed better than SCNet in these two subclasses, since it was trained in a pixel-wise manner considering only appearance similarity. Among all the methods, SCNet-AG+ performed the best with a PCK of 72.2. Fig. 5.5 presents two examples of dense matching for PF-PASCAL. After establishing the dense semantic correspondences, we warped one image to another image using the flow field. We also visualized the errors of (sparse) keypoints in the dense semantic flow. The ground-truth keypoints are presented as circles and the predicted keypoints are presented as crosses. We observe a better performance of SCNet-AG and SCNet-AG+.

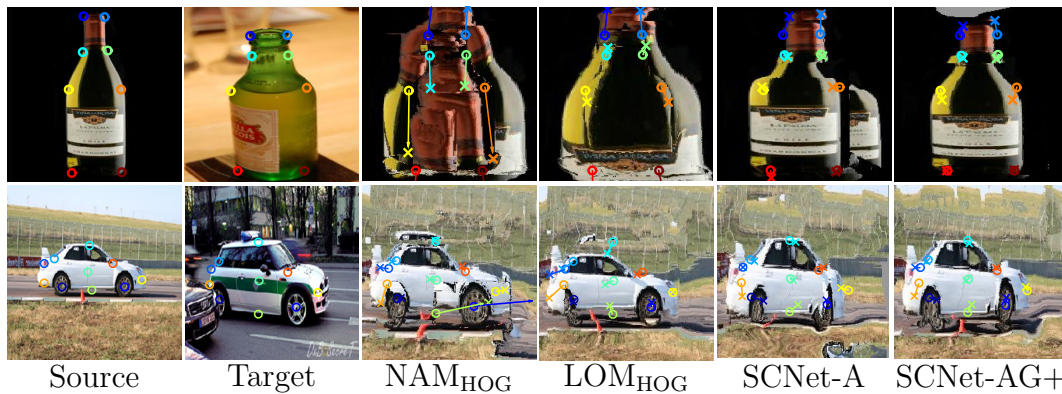


Fig. 5.5 Quantitative comparison of dense correspondence. The source image is warped to the target image using the estimated semantic flow field. The ground-truth keypoints are shown in circles and the predicted keypoints are shown in crosses. The vectors depict the matching errors.

**Results on PF-WILLOW.** For evaluating transferability, we tested (PF-PASCAL trained) SCNet and UCN on the PF-WILLOW dataset [133] and compared the results

Table 5.2 Per-class PCK (in percentage form) on PF-PASCAL at  $\tau = 0.1$ . For all methods using object proposals, we use 1000 RP proposals [134].

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	d.table	dog	horse	moto	person	plant	sheep	sofa	train	tv	mean
NAN <sub>HOG</sub> [133]	72.9	73.6	31.5	52.2	37.9	71.7	71.6	34.7	26.7	48.7	28.3	34.0	50.5	61.9	26.7	51.7	66.9	48.2	47.8	59.0	52.5
PHM <sub>HOG</sub> [133]	78.3	76.8	48.5	46.7	45.9	72.5	72.1	47.9	49.0	84.0	37.2	46.5	51.3	72.7	38.4	53.6	67.2	50.9	60.0	63.4	60.3
LOM <sub>HOG</sub> [133]	73.3	74.4	54.4	50.9	49.6	73.8	72.9	63.6	46.1	79.8	42.5	48.0	68.3	66.3	42.1	62.1	65.2	57.1	64.4	58.0	62.5
UCN [130]	64.8	58.7	42.8	59.6	47.0	42.2	61.0	45.6	49.9	52.0	48.5	49.5	53.2	72.7	53.0	41.4	<b>83.3</b>	49.0	<b>73.0</b>	66.0	55.6
SCNet-A	67.6	72.9	69.3	59.7	<b>74.5</b>	72.7	73.2	59.5	51.4	78.2	39.4	50.1	67.0	62.1	<b>69.3</b>	<b>68.5</b>	78.2	63.3	57.7	59.8	66.3
SCNet-AG	83.9	81.4	<b>70.6</b>	62.5	60.6	81.3	81.2	59.5	53.1	81.2	<b>62.0</b>	58.7	65.5	73.3	51.2	58.3	60.0	69.3	61.5	<b>80.0</b>	69.7
SCNet-AG+	<b>85.5</b>	<b>84.4</b>	66.3	<b>70.8</b>	57.4	<b>82.7</b>	<b>82.3</b>	<b>71.6</b>	<b>54.3</b>	<b>95.8</b>	55.2	<b>59.5</b>	<b>68.6</b>	<b>75.0</b>	56.3	60.4	60.0	<b>73.7</b>	66.5	76.7	<b>72.2</b>

with recent methods in Table 5.3 where PCK was averaged over all classes. The postfix ‘w/SF’ and ‘w/PF’ represent that matching is performed by SIFT Flow [123] and Proposal Flow [133], respectively. On this dataset where the data has a different distribution, SCNet-AG slightly outperformed the A and AG+ variants (PCK@0.05). We observe that all SCNet models significantly outperformed UCN, which is trained on the same dataset with the SCNet models, as well as other methods using hand-crafted features [119, 123, 155] and learned features [127, 131, 140, 141, 150, 156].

Table 5.3 Fixed-threshold PCK on PF-WILLOW. The threshold  $\tau$  is set to be 0.01, 0.1, and 0.15, respectively. PCK is averaged over all classes.

Method	PCK@0.05	PCK@0.1	PCK@0.15
SIFT Flow [123]	0.247	0.380	0.504
DAISY w/SF [119]	0.324	0.456	0.555
DeepC w/SF [141]	0.212	0.364	0.518
LIFT w/SF [156]	0.224	0.346	0.489
VGG w/SF [150]	0.224	0.388	0.555
FCSS w/SF [131]	0.354	0.532	0.681
FCSS w/PF [131]	0.295	0.584	0.715
LOM <sub>HOG</sub> [133]	0.284	0.568	0.682
UCN [130]	0.291	0.417	0.513
SCNet-A	0.390	<b>0.725</b>	<b>0.873</b>
SCNet-AG	<b>0.394</b>	0.721	0.871
SCNet-AG+	0.386	0.704	0.853

**Results on Caltech-101.** We also evaluated our approach on the Caltech-101 dataset [151]. Following the experimental protocol in [122], we randomly selected 15 pairs of images for each object class, and evaluated matching accuracy with three metrics: Label transfer accuracy (LT-ACC) [157], the IoU metric, and the localization error (LOC-ERR) of corresponding pixel positions. Both LT-ACC and IoU measure the overlap between the wrapped source images and the target images, while IoU only considers foreground region and LT-ACC considers both foreground and background regions. LOC-ERR considers pixels inside the annotated object bounding boxes only. Table 5.4 shows that SCNet achieves comparable results with the state of the art and

performs better than other existing methods using hand-crafted features. Note that, the best performer, FCSS [131], was trained on images from the same Caltech-101 dataset, while SCNet models were not.

Table 5.4 Results on Caltech-101. Three metrics, namely, LT-ACC, IoU and LOC-ERR are evaluated.

Methods	LT-ACC	IoU	LOC-ERR
NAM <sub>HOG</sub> [133]	0.70	0.44	0.39
PHM <sub>HOG</sub> [133]	0.75	0.48	0.31
LOM <sub>HOG</sub> [133]	0.78	0.50	0.26
DeepFlow [116]	0.74	0.40	0.34
SIFT Flow [123]	0.75	0.48	0.32
DSP [122]	0.77	0.47	0.35
FCSS w/SF [131]	0.80	0.50	<b>0.21</b>
FCSS w/PF [131]	<b>0.83</b>	<b>0.52</b>	0.22
SCNet-A	0.78	0.50	0.28
SCNet-AG	0.78	0.50	0.27
SCNet-AG+	0.79	0.51	0.25

**Results on PASCAL Parts.** Following [133], we used the dataset provided by [152] where the images are sampled from the PASCAL part dataset [158]. For this experiment, we measured the weighted IoU score between transferred segments and the ground truth, with weights determined by the pixel area of each part. To evaluate alignment accuracy, we measured the PCK metric ( $\tau = 0.05$ ) using keypoint annotations for the PASCAL classes. Following [133] once again, we used selective search (SS) to generate proposals for SCNet in this experiment. The results are summarized in Table 5.5. SCNet models outperformed all other results on the dataset in IoU, and SCNet-AG+ performed the best among them. FCSS w/PF [131] performed better in terms of PCK on this dataset. This is reasonable since SCNet was trained on a region-wise manner and the dense matching is achieved by post-processing, while FCSS was trained on a pixel-wise manner. Better post-processing method may help improve the dense matching results of SCNet.

Table 5.5 Results on PASCAL Parts. The PCK threshold is set as  $\tau = 0.05$ .

Methods	IoU	PCK
NAM <sub>HOG</sub> [133]	0.35	0.13
PHM <sub>HOG</sub> [133]	0.39	0.17
LOM <sub>HOG</sub> [133]	0.41	0.17
Congealing [159]	0.38	0.11
RASL [160]	0.39	0.16
CollectionFlow [161]	0.38	0.12
DSP [122]	0.39	0.17
FCSS w/SF [131]	0.44	0.28
FCSS w/PF [131]	0.46	<b>0.29</b>
SCNet-A	0.47	0.17
SCNet-AG	0.47	0.17
SCNet-AG+	<b>0.48</b>	0.18

These results verify that SCNet models have successfully learned semantic correspondence.

## 5.6 Conclusions

We have introduced a novel model for learning semantic correspondence, and proposed the corresponding CNN architecture that uses object proposals as matching primitives and learns matching in terms of appearance and geometry. The proposed method substantially outperforms both recent deep learning architectures and previous methods based on hand-crafted features. The result clearly demonstrates the effectiveness of learning geometric matching for semantic correspondence.



# Chapter 6

## Conclusions

### 6.1 Summary

This thesis has presented theoretical and practical solutions to

- single view transparent object reconstruction by altering incident light paths through refraction (Chapter 2),
- single view mirror surface reconstruction under an unknown motion of a reference pattern and an uncalibrated camera (Chapter 3),
- single view diffuse surface reconstruction with an uncalibrated camera and an unknown mirror sphere (Chapter 4), and
- semantic correspondence estimation across images containing instances of the same category (Chapter 5).

A brief summary of the algorithms and techniques introduced is given below.

The problem of transparent object reconstruction under a fixed viewpoint was addressed in Chapter 2. We proposed an approach to dense reconstruction of transparent objects based on refraction of light. We introduced a simple setup that allows us to alter the incident light paths before light rays enter the object, and recovered the object surface by reconstructing and triangulating such incident light paths. Compared

with existing methods, our approach does not need to model the complex interactions of light as it travels through the object, neither does it assume any parametric form for the object shape nor the exact number of refractions and reflections taken place along the light paths. It can handle a transparent object with a complex structure, with an unknown and even inhomogeneous refractive index. Moreover, the proposed experimental setup is simple and cheap.

A fixed viewpoint method for reconstructing mirror surface under an unknown motion of a reference pattern and an uncalibrated camera was introduced in Chapter 3. We derived a closed-form solution for estimating the camera projection matrix from reflection correspondences. The camera projection matrix was then optimized by minimizing reprojection errors computed based on a cross-ratio based nonlinear formulation. The mirror surface was then recovered based on the optimized cross-ratio constraint. The proposed method only needs reflection correspondences as input and removes the restrictive assumptions of known motions,  $C^n$  continuity of the surface, and calibrated camera that are being used by other existing methods. This greatly simplifies the challenging problem of mirror surface recovery.

The problem of single view diffuse surface reconstruction using a mirror sphere was investigated in Chapter 4. Unlike existing methods that require the intrinsic parameters of the camera and the position and radius of the sphere to be known, we tackle the challenging scenario that neither the camera is calibrated nor the mirror sphere is known. Based on eigen decomposition of the matrix representing the conic image of the sphere and enforcing a repeated eigenvalue constraint, an analytical solution was derived to recover the focal length of the camera given its principal point. Based on this analytical solution, we developed two robust algorithms for estimating both the principal point and focal length of the camera. One algorithm estimates the camera intrinsics from multiple images of the mirror sphere, and the other needs only a single image. With the estimated camera intrinsic parameters, the sphere position and a scaled 3D scene object can be obtained.

A novel approach was introduced in Chapter 5 to establish semantic correspon-



dence across images containing different instances of the same object or scene category. These images feature much larger changes in appearance and spatial layout than the images of the same scene used in stereo vision (e.g., images for multi-view 3D reconstruction). Most previous approaches to semantic correspondence focus on combining an effective spatial regularizer with hand-crafted features, or learning a correspondence model for appearance only. We proposed a convolutional neural network architecture, called SCNet, for learning a geometrically plausible model for semantic correspondence. SCNet uses region proposals as matching primitives, and explicitly incorporates geometric consistency in its loss function. State-of-the-art results have been achieved on several benchmarks.

The accuracy of our single view reconstruction of transparent, mirror and diffuse surfaces is mainly restricted by the quality of refraction and reflection correspondences. To obtain the reflection and refraction correspondences, we need the predefined pattern (i.e., a sweeping line in our experiments) and a number of images of the object. This is not a highly efficient way in practice, especially when the dimension of the object is very huge. Semantic correspondence estimation approaches that can handle the reflection and refraction distortion will drastically reduce the efforts, by taking only a pair of images as input and producing the correspondences in a fast feed-forward pass. This can make our single view reconstruction methods much more efficient.

## 6.2 Future Work

Though the methods in this thesis are novel and very practical, there are certainly rooms for improvements:

- Complete surface reconstruction

Our transparent and mirror surface reconstruction methods can reconstruct one side of an object each time. If a complete 3D model of the object is needed, we have to reconstruct the object part by part. The complete 3D model can then

be obtained by merging the 3D parts together. However, this procedure is tedious and registration difficulty will also hinder their wide application, given the viewpoint variation, shape variation, and noise in correspondences. Therefore, our methods can be improved for complete surface reconstruction.

- Dense reflection correspondence for single view diffuse surface reconstruction with a mirror sphere

The reflections on the spherical mirror suffer severe distortion, which makes it difficult to establish dense reflection correspondences. As a result, our current method can only recover sparse 3D scene points. Therefore, it will be very useful to discover solutions to dense reflection correspondence for spherical mirror based single view 3D reconstruction.

- Local spatial constraints for dense semantic correspondence

SCNet can establish semantic correspondence using proposals as primitives. Global geometry consistency is incorporated in the model. However, local geometry information is also proved to be useful in semantic matching [133], especially when strong clutter occurs in an image. Therefore, it will be beneficial to consider local geometry in SCNet in the future. Besides, dense semantic correspondences are established by interpolation from region matching results of SCNet. An end-to-end network is desired to establish dense correspondences while preserving the geometry consistency.

# Appendix A

## Pose Estimation with Reflection Correspondences

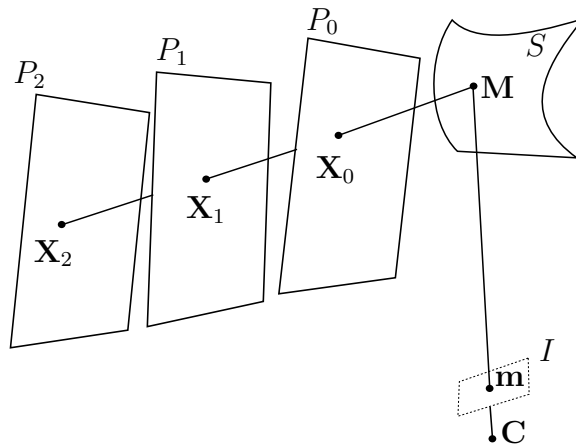


Fig. A.1 A camera centered at  $\mathbf{C}$  is viewing a mirror surface  $S$ , which is reflecting a reference plane. The reference plane is placed at three different locations, denoted as  $P_0$ ,  $P_1$  and  $P_2$ , respectively.  $\mathbf{X}_0$ ,  $\mathbf{X}_1$ ,  $\mathbf{X}_2$  are reflection correspondences defining the incident ray.

Here we briefly summarize the approach of [59] that estimates the relative poses of a reference plane under different locations. The plane is required to be placed at three different locations, namely,  $P_0$ ,  $P_1$ , and  $P_2$  (see Fig. A.1). The relative poses of  $(P_0, P_1)$  and  $(P_0, P_2)$  can be represented by  $(\mathbf{R}^1, \mathbf{T}^1)$  and  $(\mathbf{R}^2, \mathbf{T}^2)$  respectively,

where  $\mathbf{R}^i$  and  $\mathbf{T}^i$ ,  $i \in \{1, 2\}$ , denote the rigid body motion, i.e., rotation (matrix) and translation (vector). Let  $X_0$ ,  $X_1$ , and  $X_3$  stand for points on the reference plane that pass through the same incident light path. The 2D coordinates of  $X_i$  on the plane are  $\mathbf{X}_i^p = (x_i^p, y_i^p, 0)^T$ , where  $i \in \{0, 1, 2\}$ . Their 3D coordinates,  $\mathbf{X}_i = (x_i, y_i, z_i)^T$ ,  $i \in \{0, 1, 2\}$ , w.r.t  $P_0$  can be written as

$$\begin{aligned}\mathbf{X}_0 &= \mathbf{X}_0^p = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \\ \mathbf{X}_1 &= \mathbf{R}^1 \mathbf{X}_1^p + \mathbf{T}^1 \\ &= \mathbf{M} \bar{\mathbf{X}}_1^p, \\ \mathbf{X}_2 &= \mathbf{R}^2 \mathbf{X}_2^p + \mathbf{T}^2 \\ &= \mathbf{N} \bar{\mathbf{X}}_2^p,\end{aligned}$$

where  $\mathbf{M} = (\mathbf{R}_1^1, \mathbf{R}_2^1, \mathbf{T}^1)$ ,  $\mathbf{N} = (\mathbf{R}_1^2, \mathbf{R}_2^2, \mathbf{T}^2)$ ,  $\bar{\mathbf{X}}_i^p = (x_i^p, y_i^p, 1)^T$ , and  $\mathbf{R}_j^i$  denotes the  $j$ th column of  $\mathbf{R}^i$ ,  $i \in \{1, 2\}$ ,  $j \in \{1, 2\}$ . The unknown motion parameters are now embedded in  $\mathbf{M}$  and  $\mathbf{N}$ . Since  $\mathbf{X}_0$ ,  $\mathbf{X}_1$  and  $\mathbf{X}_2$  are colinear, it follows that

$$\begin{aligned}\frac{x_1 - x_0}{x_2 - x_0} &= \frac{y_1 - y_0}{y_2 - y_0} = \frac{z_1 - z_0}{z_2 - z_0}, \\ \frac{\mathbf{M}_1^T \bar{\mathbf{X}}_1^p - x_0}{\mathbf{N}_1^T \bar{\mathbf{X}}_2^p - x_0} &= \frac{\mathbf{M}_2^T \bar{\mathbf{X}}_1^p - y_0}{\mathbf{N}_2^T \bar{\mathbf{X}}_2^p - y_0} = \frac{\mathbf{M}_3^T \bar{\mathbf{X}}_1^p}{\mathbf{N}_3^T \bar{\mathbf{X}}_2^p},\end{aligned}\tag{A.1}$$

where  $\mathbf{M}_i^T$  and  $\mathbf{N}_i^T$  denote the  $i$ th row of  $\mathbf{M}$  and  $\mathbf{N}$  respectively. (A.1) gives two constraints as follows:

$$\begin{cases} (\bar{\mathbf{X}}_2^p)^T \mathbf{A} \bar{\mathbf{X}}_1^p - x_0 (\bar{\mathbf{X}}_2^p)^T \mathbf{N}_3 + x_0 (\bar{\mathbf{X}}_1^p)^T \mathbf{M}_3 = 0, \\ (\bar{\mathbf{X}}_2^p)^T \mathbf{B} \bar{\mathbf{X}}_1^p - y_0 (\bar{\mathbf{X}}_2^p)^T \mathbf{N}_3 + y_0 (\bar{\mathbf{X}}_1^p)^T \mathbf{M}_3 = 0, \end{cases}\tag{A.2}$$

where

$$\begin{aligned}\mathbf{A} &= \mathbf{N}_3 \mathbf{M}_1^T - \mathbf{N}_1 \mathbf{M}_3^T, \\ \mathbf{B} &= \mathbf{N}_3 \mathbf{M}_2^T - \mathbf{N}_2 \mathbf{M}_3^T.\end{aligned}$$

Given  $3 \times m$  points  $\bar{\mathbf{X}}_{ij}^p = (x_{ij}^p, y_{ij}^p, 1)^T$ , where  $0 \leq i \leq 2$  and  $1 \leq j \leq m$ , we can formulate the problem as solving a linear system

$$\mathbf{E} \mathbf{W} = \mathbf{0}, \quad (\text{A.3})$$

where

$$\mathbf{E} = \begin{pmatrix} (\bar{\mathbf{X}}_{21}^p)^T \otimes (\bar{\mathbf{X}}_{11}^p)^T & \mathbf{0}^T & -x_{01}^p (\bar{\mathbf{X}}_{21}^p)^T & -x_{01}^p (\bar{\mathbf{X}}_{11}^p)^T \\ \mathbf{0}^T & (\bar{\mathbf{X}}_{21}^p)^T \otimes (\bar{\mathbf{X}}_{11}^p)^T & -y_{01}^p (\bar{\mathbf{X}}_{21}^p)^T & -y_{01}^p (\bar{\mathbf{X}}_{11}^p)^T \\ \vdots & \vdots & \vdots & \vdots \\ (\bar{\mathbf{X}}_{2m}^p)^T \otimes (\bar{\mathbf{X}}_{1m}^p)^T & \mathbf{0}^T & -x_{0m}^p (\bar{\mathbf{X}}_{2m}^p)^T & -x_{0m}^p (\bar{\mathbf{X}}_{1m}^p)^T \\ \mathbf{0}^T & (\bar{\mathbf{X}}_{2m}^p)^T \otimes (\bar{\mathbf{X}}_{1m}^p)^T & -y_{0m}^p (\bar{\mathbf{X}}_{2m}^p)^T & -y_{0m}^p (\bar{\mathbf{X}}_{1m}^p)^T \end{pmatrix}, \quad (\text{A.4})$$

$$\mathbf{W} = (\mathbf{A}_1^T \ \mathbf{A}_2^T \ \mathbf{A}_3^T \ \mathbf{B}_1^T \ \mathbf{B}_2^T \ \mathbf{B}_3^T \ \mathbf{N}_3^T \ \mathbf{M}_3^T)^T, \quad (\text{A.5})$$

and  $\mathbf{A}_i^T$  and  $\mathbf{B}_i^T$  denote the  $i$ th row for  $\mathbf{A}$  and  $\mathbf{B}$  respectively.  $\otimes$  denotes kronecker tensor product.  $\mathbf{W}$  contains 24 unknowns in total. At least 12 incident rays (i.e.,  $3 \times 12$  reflection correspondences) are needed to solve all the unknowns, since each incident ray provides two constraints.

The nullity of  $\mathbf{E}$  is two for non-zero solutions, as the 21st and 24th columns are identical. Therefore, we first apply SVD to get a solution space spanned by two solution basis vectors,  $\mathbf{d}_1$  and  $\mathbf{d}_2$ .  $\mathbf{W}$  is then parameterized as

$$\mathbf{W} = \alpha(\mathbf{d}_1 + \beta \mathbf{d}_2), \quad (\text{A.6})$$

where  $\alpha$  and  $\beta$  are scale parameters. Now there are 26 unknowns in total. By enforcing the element-wise equality of (A.5), and (A.6), we have 18 bilinear and 6 linear equations to solve  $\mathbf{M}$ ,  $\mathbf{N}$ ,  $\alpha$  and  $\beta$ . Besides, the orthonormality property of the rotation matrices encoded in  $\mathbf{M}$  and  $\mathbf{N}$  will provide 6 more constraints. A closed-form solution for the

unknown motion parameters and the two scale parameters can then be obtained by solving these equations. We use the symbolic Math Toolbox in Matlab to solve them.

# Appendix B

## Line Projection Matrix and Camera Projection Matrix

After achieving the line projection matrix, the following task is to convert to its corresponding camera (point) projection matrix, so that we can know the intrinsic and extrinsic parameters of the camera.

First, consider the case of transforming a point projection matrix to its equivalent line projection matrix. It can be seen in Fig. 3.3 that the plane  $\mathbf{P}_1^T \mathbf{X} = \mathbf{0}$  is described by the camera center and the line  $u = 0$  in the image plane. Similarly,  $\mathbf{P}_2^T \mathbf{X} = \mathbf{0}$  is described by the camera center and the line  $v = 0$  in the image plane. Finally, the plane equation  $\mathbf{P}_3^T \mathbf{X} = \mathbf{0}$  holds for all points with pixel coordinates  $s = 0$ . We can obtain the  $i$ -th row of  $\mathcal{P}$  by the intersection of rows  $j$  and  $k$  of  $\mathbf{P}$ , i.e.

$$\mathcal{P}_i^T = \begin{bmatrix} \rho_{i1} \\ \rho_{i2} \\ \rho_{i3} \\ \rho_{i4} \\ \rho_{i5} \\ \rho_{i6} \end{bmatrix} = (-1)^{(i+1)} \begin{bmatrix} p_{j3}p_{k4} - p_{j4}p_{k3} \\ p_{j4}p_{k2} - p_{j2}p_{k4} \\ p_{j2}p_{k3} - p_{j3}p_{k2} \\ p_{j1}p_{k4} - p_{j4}p_{k1} \\ p_{j1}p_{k2} - p_{j2}p_{k1} \\ p_{j1}p_{k3} - p_{j3}p_{k1} \end{bmatrix},$$

where  $\mathbf{P}_i$  is the  $i$ -th row of  $\mathbf{P}$ ,  $\mathcal{P}_i$  is the  $i$ -th row of  $\mathcal{P}$ ,  $i \neq j \neq k \in \{1, 2, 3\}$  and  $j < k$ .

Note that  $\mathcal{P}_i^T$  is the dual Plücker vector of  $(-1)^{i+1}(\mathbf{P}_j \wedge \mathbf{P}_k)$ , i.e. the intersection of the  $j$ -th with the  $k$ -th row of  $\mathbf{P}$ . The sign here controls the order of intersection, i.e.  $(\mathbf{P}_j \wedge \mathbf{P}_k) = -(\mathbf{P}_k \wedge \mathbf{P}_j)$ . Dually, we can obtain the  $i$ -th row of  $\mathbf{P}$  by the intersection of rows  $j$  and  $k$  of  $\mathcal{P}$ , which results in the homogeneous plane

$$\begin{aligned} \mathbf{P}_i^T &= (-1)^{(i+1)} \begin{bmatrix} \omega_j \times \omega_k \\ \nu_j \cdot \omega_k \end{bmatrix} \\ &= (-1)^{(i+1)} \begin{bmatrix} \rho_{j5}\rho_{k6} - \rho_{j6}\rho_{k5} \\ \rho_{j5}\rho_{k3} - \rho_{j3}\rho_{k5} \\ \rho_{j6}\rho_{k3} - \rho_{j3}\rho_{k6} \\ \rho_{j4}\rho_{k3} + \rho_{j2}\rho_{k6} + \rho_{j1}\rho_{k5} \end{bmatrix}, \end{aligned}$$

where again  $i \neq j \neq k \in \{1, 2, 3\}$  with  $j < k$ ,  $\omega_j$  is the direction vector of  $\mathcal{P}_j^T$  and  $\nu_j$  is the moment vector of  $\mathcal{P}_j^T$ .



# Appendix C

## Back-propagation for Hough Voting

Given two images associated with  $p$  and  $q$  proposals respectively. The appearance similarity matrix  $F$  is obtained by considering all possible matches between the two:

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} & \dots & f_{1q} \\ f_{21} & f_{22} & f_{23} & \dots & f_{2q} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{p1} & f_{p2} & f_{p3} & \dots & f_{pq} \end{bmatrix}. \quad (\text{C.1})$$

The geometric similarity matrix  $V$  contains geometric scores assigned to the matches:

$$V = \begin{bmatrix} v_{11} & v_{12} & v_{13} & \dots & v_{1q} \\ v_{21} & v_{22} & v_{23} & \dots & v_{2q} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{p1} & v_{p2} & v_{p3} & \dots & v_{pq} \end{bmatrix}, \quad (\text{C.2})$$

$$v_{ij} = \text{vec}(F)^T g_{ij} \quad (\text{C.3})$$

where  $\text{vec}$  denotes column-wise vectorization, and the geometry voting vector  $g_{ij}$  for a match  $m$  is a  $(p \times q) \times 1$  vector with each element being 0 or 1, such that  $v_{ij}$  is a partial sum of the elements in  $F$ .

To learn SCNet, we need to back-propagate for the voting. To achieve this, we

need to first calculate  $dL/dV$ , which is a  $p \times q$  matrix, and  $L$  stands for loss.

According to the chain rule, we have  $dL/dF = dL/dV \cdot dV/dF$ , and  $dV/dF$  is a  $(p \times q) \times (p \times q)$  matrix.

$$\begin{aligned}
dL/df_{ij} &= dL/dV \cdot dV/df_{ij} \\
&= [dL/dv_{11}, dL/dv_{21}, \dots, dL/dv_{pq}] \\
&\quad [dv_{11}/df_{ij}, dv_{21}/df_{ij}, \dots, dv_{pq}/df_{ij}]^T \\
&= [dL/dv_{11}, dL/dv_{21}, \dots, dL/dv_{pq}]g_{ij}
\end{aligned} \tag{C.4}$$

By vectorizing  $dL/dV$ , we simplify the gradient estimation for  $dL/dF$  (index follows Matlab fashion) as

$$\begin{aligned}
dL/dF &= [dL/dv_{11}, dL/dv_{21}, \dots, dL/dv_{pq}] \\
&\quad \begin{bmatrix} dv_{11}/df_{11} & dv_{11}/df_{21} & \dots & dv_{11}/df_{pq} \\ dv_{21}/df_{11} & dv_{21}/df_{21} & \dots & dv_{21}/df_{pq} \\ \vdots & \vdots & \ddots & \vdots \\ dv_{pq}/df_{11} & dv_{pq}/df_{21} & \dots & dv_{pq}/df_{pq} \end{bmatrix} \\
&= [dL/dv_{11}, dL/dv_{21}, \dots, dL/dv_{pq}] \\
&\quad [g_{11}, g_{21}, g_{31}, \dots, g_{pq}]
\end{aligned} \tag{C.5}$$

For efficiency, we first compute the offset matrix  $H$  for all candidate matches.

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \dots & h_{1p} \\ h_{21} & h_{22} & h_{23} & \dots & h_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{p1} & h_{p2} & h_{p3} & \dots & h_{pq} \end{bmatrix}, \tag{C.6}$$

where  $h_{ij}$  is the offset bin index corresponding to the match  $m = (r_i, r_j)$ , that is a match between  $r_i$  in image  $I_A$  and  $r_j$  in image  $I_B$ . Note that there are only  $t$  unique numbers in  $H$ , which means that there are  $t$  different offsets among the  $p \times q$  pairs of

possible matches.

For example,  $H$  can be in the form of

$$H = \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_4 \\ x_2 & x_1 & x_2 & \dots & x_4 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1 & x_4 & x_4 & \dots & x_3 \end{bmatrix}$$

In this example, since  $h_{11} = h_{n1} = h_{22} = x_1$ , then  $g$  vector for offset  $x_1$  is

$$g_{x_1} = \text{vec} \left( \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} \right)$$

Since matches  $m = (r_i, r_j)$  with the same offset index have the same gradient  $dL/df_{ij}$ ,  $dL/df_{ij}$  can be computed only once by (C.4) and shared among them.



# References

- [1] H. C. Longuet-Higgins, “A computer algorithm for reconstructing a scene from two projections,” *Nature*, vol. 293, pp. 133–135, 1981.
- [2] A. Chiuso, R. Brockett, , and S. Soatto, “Optimal structure from motion: Local ambiguities and global estimates,” *International Journal of Computer Vision (IJCV)*, vol. 39, no. 3, pp. 195–228, 2000.
- [3] J. Oliensis, “A critique of structure-from-motion algorithms,” *International Journal of Computer Vision (IJCV)*, vol. 80, no. 2, pp. 172–214, 2000.
- [4] A. Laurentini, “The visual hull concept for silhouette-based image understanding,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 16, no. 2, pp. 150–162, 1994.
- [5] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan, “Image-based visual hulls,” in *Proc. SIGGRAPH*, 2000, pp. 369–374.
- [6] A. Tankus, N. Sochen, and Y. Yeshurun, “Shape-from-shading under perspective projection,” *International Journal of Computer Vision (IJCV)*, vol. 63, no. 1, pp. 21–43, 2005.
- [7] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, “Shape from shading: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 21, no. 8, pp. 690–706, 1999.
- [8] M. Daum and G. Dudekb, “Out of the dark: Using shadows to reconstruct 3d surfaces,” in *Asian Conference on Computer Vision (ACCV)*, 1998, pp. 72–79.
- [9] —, “On 3-d surface reconstruction using shape from shadows,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998, pp. 461–468.
- [10] M. G. Jens Ackermann, “A survey of photometric stereo techniques,” *Foundations and Trends in Computer Graphics and Vision*, vol. 9, no. 3–4, pp. 149–254, 2015.
- [11] R. J. Woodham, “Photometric method for determining surface orientation from multiple images,” *Optical Engineering*, vol. 19, no. 1, pp. 139–144, 1980.

- 
- [12] M. Liu, R. Hartley, and M. Salzmann, “Mirror surface reconstruction from a single image,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 37, no. 4, pp. 760–773, 2015.
- [13] I. Ihrke, K. N. Kutulakos, H. P. A. Lensch, M. Magnor, and W. Heidrich, “State of the art in transparent and specular object reconstruction,” in *Eurographics STAR*, 2008, pp. 87–108.
- [14] —, “Transparent and specular object reconstruction,” *Computer Graphics Forum*, vol. 29, no. 8, pp. 2400–2426, 2010.
- [15] I. Reshetouski and I. Ihrke, “Mirrors in computer graphics, computer vision and time-of-flight imaging,” *Lecture Notes in Computer Science*, vol. 8200, pp. 77–104, 2013.
- [16] Z. Chen, K.-Y. K. Wong, M. Liu, and D. Schnieders, “Single-view reconstruction from an unknown spherical mirror,” in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 2733–2736.
- [17] Z. Chen, K.-Y. K. Wong, Y. Matsushita, X. Zhu, and M. Liu, “Self-calibrating depth from refraction,” in *Proc. International Conference on Computer Vision (ICCV)*, 2011, pp. 635–642.
- [18] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.
- [19] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 886–893.
- [20] S. Inokuchi, K. Sato, and F. Matsuda, “Range-imaging for 3-d object recognition,” in *Proc. IEEE International Conference on Pattern Recognition (ICPR)*, 1984, pp. 806–808.
- [21] C. Wust and D. W. Capson, “Surface profile measurement using color fringe projection,” *Machine Vision and Applications*, vol. 4, no. 3, pp. 193–203, 1991.
- [22] K. Han, K.-Y. K. Wong, and M. Liu, “A fixed viewpoint approach for dense reconstruction of transparent objects,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4001–4008.
- [23] —, “Dense reconstruction of transparent objects by altering incident light paths through refraction,” *International Journal of Computer Vision (IJCV)*, 2017.
- [24] K. Han, K.-Y. K. Wong, D. Schnieders, and M. Liu, “Mirror surface reconstruction under an uncalibrated camera,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1772–1780.

- [25] K. Han, K.-Y. K. Wong, and X. Tan, “Single view 3d reconstruction under an uncalibrated camera and an unknown mirror spher,” in *Proc. International Conference on 3D Vision (3DV)*, 2016, pp. 408–416.
- [26] K. Han, R. S. Rezende, B. Ham, K.-Y. K. Wong, M. Cho, C. Schmid, and J. Ponce, “Scnet: Learning semantic correspondence,” in *Proc. International Conference on Computer Vision (ICCV)*, 2017.
- [27] H. Murase, “Surface shape reconstruction of an undulating transparent object,” in *Proc. International Conference on Computer Vision (ICCV)*, 1990, pp. 313–317.
- [28] —, “Surface shape reconstruction of a nonrigid transparent object using refraction and motion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 14, no. 10, pp. 1045–1052, 1992.
- [29] Q. Shan, S. Agarwal, and B. Curless, “Refractive height fields from single and multiple images,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 286–293.
- [30] K. N. Kutulakos and E. Steger, “A theory of refractive and specular 3D shape by light-path triangulation,” in *Proc. International Conference on Computer Vision (ICCV)*, vol. 2, 2005, pp. 1448–1455.
- [31] —, “A theory of refractive and specular 3D shape by light-path triangulation,” *International Journal of Computer Vision (IJCV)*, vol. 76, no. 1, pp. 13–29, 2008.
- [32] C. Tsai, A. Veeraraghavan, and A. C. Sankaranarayanan, “What does a single light-ray reveal about a transparent object?” in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 606–610.
- [33] N. J. W. Morris and K. N. Kutulakos, “Dynamic refraction stereo,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33, no. 10, pp. 1518–1531, 2011.
- [34] S. Hata, Y. Saitoh, S. Kumamura, and K. Kaida, “Shape extraction of transparent object using genetic algorithm,” in *Proc. IEEE International Conference on Pattern Recognition (ICPR)*, vol. 4, 1996, pp. 684–688.
- [35] M. Ben-Ezra and S. K. Nayar, “What does motion reveal about transparency?” in *Proc. International Conference on Computer Vision (ICCV)*, vol. 2, 2003, pp. 1025–1032.
- [36] X. Zuo, C. Du, S. Wang, J. Zheng, and R. Yang, “Interactive visual hull refinement for specular and transparent object surface reconstruction,” in *Proc. International Conference on Computer Vision (ICCV)*, 2015, pp. 2237–2245.

- 
- [37] Y. Qian, M. Gong, and Y.-H. Yang, “3d reconstruction of transparent objects with position-normal consistency,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4369–4377.
- [38] G. Wetzstein, D. Roodnick, W. Heidrich, and R. Raskar, “Refractive shape from light field distortion,” in *Proc. International Conference on Computer Vision (ICCV)*, 2011, pp. 1180–1186.
- [39] Y. Ding, F. Li, Y. Ji, and J. Yu, “Dynamic fluid surface acquisition using a camera array,” in *Proc. International Conference on Computer Vision (ICCV)*, 2011, pp. 2478–2485.
- [40] G. Eren, O. Aubreton, F. Meriaudeau, L. A. S. Secades, D. Fofi, A. T. Naskali, F. Truchetet, and A. Ercil, “Scanning from heating: 3D shape estimation of transparent objects from local surface heating,” *Optics Express*, vol. 17, no. 14, pp. 11 457–11 468, 2009.
- [41] I. Ihrke, B. Goidluecke, and M. Magnor, “Reconstructing the geometry of flowing water,” in *Proc. International Conference on Computer Vision (ICCV)*, vol. 2, 2005, pp. 1055–1060.
- [42] D. Miyazaki and K. Ikeuchi, “Inverse polarization raytracing: estimating surface shapes of transparent objects,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005, pp. 910–917.
- [43] B. Trifonov, D. Bradley, and W. Heidrich, “Tomographic reconstruction of transparent objects,” in *SIGGRAPH 2006 Sketches*, 2006, pp. 51–60.
- [44] M. B. Hullin, M. Fuchs, I. Ihrke, H.-P. Seidel, and H. P. A. Lensch, “Fluorescent immersion range scanning,” in *Proc. SIGGRAPH*, 2008, pp. 87:1–87:10.
- [45] S. G. Narasimhan, S. K. Nayar, B. Sun, and S. J. Koppal, “Structured light in scattering media,” in *Proc. International Conference on Computer Vision (ICCV)*, 2005, pp. 420–427.
- [46] M. O’Toole, J. Mather, and K. N. Kutulakos, “3d shape and indirect appearance by structured light transport,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3246–3253.
- [47] C. Ma, X. Lin, J. Suo, Q. Dai, and G. Wetzstein, “Transparent object reconstruction via coded transport of intensity,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3238–3245.
- [48] Y. Ji, J. Ye, and J. Yu, “Reconstructing gas flows using light-path approximation,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2507–2514.



- [49] J. Balzer and S. Werling, “Principles of shape from specular reflection,” *Measurement*, vol. 43, no. 10, pp. 1305–1317, 2010.
- [50] N. J. W. Morris and K. N. Kutulakos, “Reconstructing the surface of inhomogeneous transparent scenes by scatter-trace photography,” in *Proc. International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.
- [51] 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 2513–2520.
- [52] V. Chari and P. Sturm, “A theory of refractive photo-light-path triangulation,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1438–1445.
- [53] D. Liu, X. Chen, and Y.-H. Yang, “Frequency-based 3d reconstruction of transparent and specular objects,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 660–667.
- [54] J.-Y. Bouguet, “Camera calibration toolbox for matlab,” [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/).
- [55] J. Batlle, E. Mouaddib, and J. Salvi, “Recent progress in coded structured light as a technique to solve the correspondence problem: a survey,” *Pattern Recognition*, vol. 31, no. 7, pp. 963–982, 1998.
- [56] W. Xie, Y. Zhang, C. C. L. Wang, , and R. C.-K. Chung, “Surface-from-gradients: An approach based on discrete geometry processing,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 2203–2210.
- [57] 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.
- [58] P. H. S. Torr and A. Zisserman, “MLE-SAC: A new robust estimator with application to estimating image geometry,” *Computer Vision and Image Understanding (CVIU)*, vol. 78, no. 1, pp. 138–156, 2000.
- [59] M. Liu, K.-Y. K. Wong, Z. Dai, and Z. Chen, “Pose estimation from reflections for specular surface recovery,” in *Proc. International Conference on Computer Vision (ICCV)*, 2011, pp. 579–586.
- [60] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

- 
- [61] M. Oren and S. K. Nayar, “A theory of specular surface geometry,” *International Journal of Computer Vision (IJCV)*, vol. 24, no. 2, pp. 105–124, 1996.
- [62] S. Roth and M. J. Black, “Specular flow and the recovery of surface structure,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 1869–1876.
- [63] Y. Adato, Y. Vasilyev, O. Ben-Shahar, and T. Zickler, “Toward a theory of shape from specular flow,” in *Proc. International Conference on Computer Vision (ICCV)*, 2007, pp. 1–8.
- [64] Y. Adato, Y. Vasilyev, T. Zickler, and O. Ben-Shahar, “Shape from specular flow,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 32, no. 11, pp. 2054–2070, 2010.
- [65] Y. Vasilyev, T. Zickler, S. Gortler, and O. Ben-Shahar, “Shape from specular flow: Is one flow enough?” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 2561–2568.
- [66] G. D. Canas, Y. Vasilyev, Y. Adato, T. Zickler, S. Gortler, and O. Ben-Shahar, “A linear formulation of shape from specular flow,” in *Proc. International Conference on Computer Vision (ICCV)*, 2009, pp. 191–198.
- [67] A. Sankaranarayanan, A. Veeraraghavan, O. Tuzel, and A. Agrawal, “Specular surface reconstruction from sparse reflection correspondences,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1245–1252.
- [68] S. Savarese and P. Perona, “Local analysis for 3d reconstruction of specular surfaces.” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001, pp. 738–745.
- [69] ———, “Local analysis for 3d reconstruction of specular surfaces – part ii,” in *Proc. European Conference on Computer Vision (ECCV)*, 2002, pp. 759–774.
- [70] S. Rozenfeld, I. Shimshoni, and M. Lindenbaum, “Dense mirroring surface recovery from 1d homographies and sparse correspondences,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33, no. 2, pp. 325–337, 2011.
- [71] T. Bonfort and P. Sturm, “Voxel carving for specular surfaces,” in *Proc. International Conference on Computer Vision (ICCV)*, 2003, pp. 691–696.
- [72] T. Bonfort, P. Sturm, and P. Gargallo, “General specular surface triangulation,” in *Asian Conference on Computer Vision (ACCV)*, 2006, pp. 872–881.
- [73] P. Sturm and T. Bonfort, “How to compute the pose of an object without a direct view,” in *Asian Conference on Computer Vision (ACCV)*, 2006, pp. 21–31.

- [74] D. Nehab, T. Weyrich, and S. Rusinkiewicz, “Dense 3d reconstruction from specular consistency,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.
- [75] M. Weinmann, A. Osep, R. Ruiters, and R. Klein, “Multi-view normal field integration for 3d reconstruction of mirroring objects,” in *Proc. International Conference on Computer Vision (ICCV)*, 2013, pp. 2504–2511.
- [76] J. Balzer, D. Acevedo-Feliz, S. Soatto, S. Höfer, M. Hadwiger, and J. Beyerer, “Cavlectometry: Towards holistic reconstruction of large mirror objects,” in *Proc. International Conference on 3D Vision (3DV)*, 2014, pp. 448–455.
- [77] L. Perdigoto and H. Araujo, “Calibration of mirror position and extrinsic parameters in axial non-central catadioptric systems,” *Computer Vision and Image Understanding (CVIU)*, vol. 117, pp. 909–921, 2013.
- [78] S. Ramalingam, M. Antunes, D. Snow, G. Hee Lee, and S. Pillai, “Line-sweep: Cross-ratio for wide-baseline matching and 3d reconstruction,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1238–1246.
- [79] S. Ramalingam, P. Sturm, and S. K. Lodha, “Towards complete generic camera calibration,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 1093–1098.
- [80] M. Kazhdan and H. Hoppe, “Screened Poisson surface reconstruction,” *ACM Transactions on Graphics (TOG)*, vol. 32, pp. 29:1–29:13, 2013.
- [81] J.-Y. Bouguet, “Camera calibration toolbox for matlab,” [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/).
- [82] P. J. Besl and H. D. McKay, “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 14, no. 2, pp. 239–256, 1992.
- [83] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs, “Large scale multi-view stereopsis evaluation,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 406–413.
- [84] J. Gluckman and S. K. Nayar, “Planar catadioptric stereo: Geometry and calibration,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 1999, pp. 22–28.
- [85] —, “Catadioptric stereo using planar mirrors,” *International Journal of Computer Vision (IJCV)*, vol. 44, no. 1, pp. 65–79, 2001.
- [86] —, “Rectified catadioptric stereo sensors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 24, no. 2, pp. 224–236, 2002.

- [87] Y. Ding, J. Yu, and P. Sturm, “Multi-perspective stereo matching and volumetric reconstruction,” in *Proc. International Conference on Computer Vision (ICCV)*, 2009, pp. 1827–1834.
- [88] G. Jang, S. Kim, and I. Kweon, “Single camera catadioptric stereo system,” in *Proc. IEEE Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras (OMNIVIS)*, 2005, pp. 1–7.
- [89] D. Lanman, D. Crispell, M. Wachs, and G. Taubin, “Spherical catadioptric arrays: Construction, multiview geometry, and calibration,” in *Proc. International Conference on 3D Data Processing, Modeling, Processing, Visualization and Transmission (3DPVT)*, 2006, pp. 81–88.
- [90] D. Lanman, M. Wachs, G. Taubin, and F. Cukierman, “Reconstructing a 3d line from a single catadioptric image,” in *Proc. International Conference on 3D Data Processing, Modeling, Processing, Visualization and Transmission (3DPVT)*, 2006, pp. 1–8.
- [91] K. Tan, H. Hua, and N. Ahuja, “Multiview panoramic cameras using mirror pyramids,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 26, no. 7, pp. 941–946, 2004.
- [92] M. Kanbara, N. Ukita, M. Kidode, and N. Yokoya, “3d scene reconstruction from reflection images in a spherical mirror,” in *Proc. IEEE International Conference on Pattern Recognition (ICPR)*, 2006, pp. 874–879.
- [93] S. Nayar, “Sphereo: Determining depth using two specular spheres and a single camera,” in *SPIE Conf. Optics, Illumination, and Image Sensing for Machine Vision III*, 1988, pp. 245—254.
- [94] Y. Taguchi, A. Agrawal, A. Veeraraghavan, S. Ramalingam, and R. Raskar, “Axial-cones: Modeling spherical catadioptric cameras for wide-angle light field rendering,” *ACM Transactions on Graphics (TOG)*, vol. 29, no. 6, pp. 172:1–172:8, 2010.
- [95] R. Sagawa, N. Kurita, T. Echigo, and T. Yagi, “Compound catadioptric stereo sensor for omnidirectional object detection,” in *Proc. International Conference on Intelligent Robots and Systems (IROS)*, 2004, pp. 2612–2617.
- [96] B. Hu, “It’s all done with mirrors: Calibration-and-correspondence-free 3d reconstruction,” in *Proc. Canadian Conference on Computer and Robot Vision (CRV)*, 2009, pp. 148–154.
- [97] H. Zhong, W. F. Sze, and Y. S. Hung, “Reconstruction from plane mirror reflection,” in *Proc. IEEE International Conference on Pattern Recognition (ICPR)*, 2006, pp. 715–718.

- [98] I. Ihrke, T. Stich, H. Gottschlich, M. Magnor, and H. Seidel, “Fast incident light field acquisition and rendering,” in *Journal of WSCG*, vol. 16, 2008, pp. 25–32.
- [99] H. Mitsumoto, S. Tamura, K. Okazaki, N. Kajimi, and Y. Fukui, “3-d reconstruction using mirror images based on a plane symmetry recovering method,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 14, pp. 941–947, 1992.
- [100] A. Agrawal, “Extrinsic camera calibration without a direct view using spherical mirror,” in *Proc. International Conference on Computer Vision (ICCV)*, 2013, pp. 2368–2375.
- [101] A. Agrawal and S. Ramalingam, “Single image calibration of multi-axial imaging systems,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1399–1406.
- [102] M. Powell, S. Sarkar, and D. Goldgof, “A simple strategy for calibrating the geometry of light sources,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 23, pp. 1022–1027, 2001.
- [103] D. Schnieders and K.-Y. K. Wong, “Camera and light calibration from reflections on a sphere,” *Computer Vision and Image Understanding (CVIU)*, vol. 117, pp. 1536–1547, 2013.
- [104] K.-Y. K. Wong, D. Schnieders, and S. Li, “Recovering light directions and camera poses from a single sphere,” in *Proc. European Conference on Computer Vision (ECCV)*, vol. I, 2008, pp. 631–642.
- [105] C. Nitschke, A. Nakazawa, and H. Takemura, “Display-camera calibration using eye reflections and geometry constraints,” *Computer Vision and Image Understanding (CVIU)*, vol. 115, pp. 835–853, 2011.
- [106] D. Schnieders, X. Fu, and K.-Y. K. Wong, “Reconstruction of display and eyes from a single image,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1442–1449.
- [107] M. Agrawal and L. S. Davis, “Camera calibration using spheres: A semi-definite programming approach,” in *Proc. International Conference on Computer Vision (ICCV)*, 2003, pp. 782–789.
- [108] J. Sun, X. Chen, Z. Gong, Z. Liu, and Y. Zhao, “Accurate camera calibration with distortion models using sphere images,” *Optics & Laser Technology*, vol. 65, pp. 83–87, 2015.
- [109] K.-Y. K. Wong, G. Zhang, and Z. Chen, “A stratified approach for camera calibration using spheres,” *IEEE Transactions on Image Processing (TIP)*, vol. 20, no. 2, pp. 305–316, 2011.

- [110] X. Ying and H. Zha, “Geometric interpretations of the relation between the image of the absolute conic and sphere images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 28, no. 12, pp. 2031–2036, 2006.
- [111] H. Zhang, K.-Y. K. Wong, and G. Zhang, “Camera calibration from images of spheres,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 29, no. 3, pp. 499–503, 2007.
- [112] G. Cardano, *The Rules of Algebra (Ars Magna)*. Dover Publications, 2007.
- [113] M. Okutomi and T. Kanade, “A multiple-baseline stereo,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 15, no. 4, pp. 353–363, 1993.
- [114] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, “Fast cost-volume filtering for visual correspondence and beyond,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3017–3024.
- [115] B. K. Horn and B. G. Schunck, “Determining optical flow: A retrospective,” *Artificial Intelligence*, vol. 59, no. 1, pp. 81–87, 1993.
- [116] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, “Deepmatching: Hierarchical deformable dense matching,” 2015.
- [117] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, “Deepflow: Large displacement optical flow with deep matching,” in *Proc. International Conference on Computer Vision (ICCV)*, 2013, pp. 1385–1392.
- [118] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and Vision Computing (IVC)*, vol. 22, no. 10, pp. 761–767, 2004.
- [119] H. Yang, W.-Y. Lin, and J. Lu, “Daisy filter flow: A generalized discrete approach to dense correspondences,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3406–3413.
- [120] H. Bristow, J. Valmadre, and S. Lucey, “Dense semantic correspondence where every pixel is a classifier,” in *Proc. International Conference on Computer Vision (ICCV)*, 2015, pp. 4024–4031.
- [121] J. Hur, H. Lim, C. Park, and S. C. Ahn, “Generalized deformable spatial pyramid: Geometry-preserving dense correspondence estimation,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1392–1400.

- [122] J. Kim, C. Liu, F. Sha, and K. Grauman, “Deformable spatial pyramid matching for fast dense correspondences,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2307–2314.
- [123] C. Liu, J. Yuen, and A. Torralba, “SIFT flow: Dense correspondence across scenes and its applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33, no. 5, pp. 978–994, 2011.
- [124] T. Tanai, S. N. Sinha, and Y. Sato, “Joint recovery of dense correspondence and cosegmentation in two images,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4246–4255.
- [125] E. Tola, V. Lepetit, and P. Fua, “Daisy: An efficient dense descriptor applied to wide-baseline stereo,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 32, no. 5, pp. 815–830, 2010.
- [126] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, “Flownet: Learning optical flow with convolutional networks,” in *Proc. International Conference on Computer Vision (ICCV)*, 2015, pp. 2758–2766.
- [127] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, “MatchNet: Unifying feature and metric learning for patch-based matching,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3279–3286.
- [128] J. Žbontar and Y. LeCun, “Computing the stereo matching cost with a convolutional neural network,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1592–1599.
- [129] J. Zbontar and Y. LeCun, “Stereo matching by training a convolutional neural network to compare image patches,” *Journal of Machine Learning Research (JMLR)*, vol. 17, no. 2, pp. 1–32, 2016.
- [130] C. Choy, J. Gwak, S. Savarese, and M. Chandraker, “Universal correspondence network,” in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 2406–2414.
- [131] S. Kim, D. Min, B. Ham, S. Jeon, S. Lin, and K. Sohn, “Fcss: Fully convolutional self-similarity for dense semantic correspondence,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6560–6569.
- [132] T. Zhou, P. Krähenbühl, M. Aubry, Q. Huang, and A. A. Efros, “Learning dense correspondence via 3d-guided cycle consistency,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 117–126.

- 
- [133] B. Ham, M. Cho, C. Schmid, and J. Ponce, “Proposal flow,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3475–3484.
- [134] S. Manen, M. Guillaumin, and L. Van Gool, “Prime object proposals with randomized Prim’s algorithm,” in *Proc. International Conference on Computer Vision (ICCV)*, 2013, pp. 2536–2543.
- [135] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition,” *International Journal of Computer Vision (IJCV)*, vol. 104, no. 2, pp. 154–171, 2013.
- [136] C. L. Zitnick and P. Dollár, “Edge boxes: Locating object proposals from edges,” in *Proc. European Conference on Computer Vision (ECCV)*, 2014, pp. 391–405.
- [137] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge 2007 (voc2007) results.”
- [138] B. Hariharan, J. Malik, and D. Ramanan, “Discriminative decorrelation for clustering and classification,” in *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2012, pp. 459–472.
- [139] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1106–1114.
- [140] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, “Discriminative learning of deep convolutional feature point descriptors,” in *Proc. International Conference on Computer Vision (ICCV)*, 2015, pp. 118–126.
- [141] S. Zagoruyko and N. Komodakis, “Learning to compare image patches via convolutional neural networks,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4353–4361.
- [142] J. L. Long, N. Zhang, and T. Darrell, “Do convnets learn correspondence?” in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 1601–1609.
- [143] A. Kanazawa, D. W. Jacobs, and M. Chandraker, “WarpNet: Weakly supervised matching for single-view reconstruction,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3253–3261.
- [144] M. Tau and T. Hassner, “Dense correspondences across scenes and scales,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 38, no. 5, pp. 875–888, 2016.



- [145] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, “Multiscale combinatorial grouping,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 328–335.
- [146] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, “What makes for effective detection proposals?” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 38, no. 4, pp. 814–830, 2016.
- [147] M. Cho, S. Kwak, C. Schmid, and J. Ponce, “Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1201–1210.
- [148] R. Girshick, “Fast r-cnn,” in *Proc. International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.
- [149] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” in *Proc. European Conference on Computer Vision (ECCV)*, 2014, pp. 346–361.
- [150] K. Simonyan and Andrew Zisserman, “Very deep convolutional networks for large-scale visual recognition,” in *arXiv preprint arXiv:1409.1556*, 2014.
- [151] L. Fei-Fei, R. Fergus, and P. Perona, “One-shot learning of object categories,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 28, no. 4, pp. 594–611, 2006.
- [152] T. Zhou, Y. Jae Lee, S. X. Yu, and A. A. Efros, “FlowWeb: Joint image set alignment by weaving consistent, pixel-wise correspondences,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1191–1200.
- [153] Y. Yang and D. Ramanan, “Articulated human detection with flexible mixtures of parts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 35, no. 12, pp. 2878–2890, 2013.
- [154] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *International Journal of Computer Vision (IJCV)*, vol. 104, no. 2, pp. 154–171, 2013.
- [155] S. W. Kim, D. Min, B. Ham, and K. Sohn, “Dasc: Dense adaptative self-correlation descriptor for multi-modal and multi-spectral correspondence,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2103–2108.
- [156] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, “Lift: Learned invariant feature transform,” in *Proc. European Conference on Computer Vision (ECCV)*, 2016, pp. 467–483.

- 
- [157] C. Liu, J. Yuen, and A. Torralba, “Nonparametric scene parsing via label transfer,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 33, no. 12, pp. 2368–2382, 2011.
- [158] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun *et al.*, “Detect what you can: Detecting and representing objects using holistic models and body parts,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1979–1986.
- [159] E. G. Learned-Miller, “Data driven image models through continuous joint alignment,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 28, no. 2, pp. 236–250, 2006.
- [160] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, “Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34, no. 11, pp. 2233–2246, 2012.
- [161] I. Kemelmacher-Shlizerman and S. M. Seitz, “Collection flow,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1792–1799.