

Single View Analysis of Non-Lambertian Objects Based on Deep Learning



Guanying Chen
陳冠英

Department of Computer Science
The University of Hong Kong

This dissertation is submitted for
Doctor of Philosophy

December, 2020

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.

Guanying Chen

December, 2020

Acknowledgements

First, I would like to express my greatest gratitude to my supervisor, Dr. Kwan-Yee Kenneth Wong, for his continuous support, thoughtful comments, and encouragement throughout my PhD study. Dr. Wong's supervision helped me start my research in computer vision. I am very grateful for his efforts in both my research and life.

I was lucky to work with my collaborators, and would like to thank them for their invaluable advice and help: Dr. Kai Han for all my three projects (Chapter 1, Chapter 2, and part of Chapter 3); Dr. Boxin Shi and Prof. Yasuyuki Matsushita for my second and third projects (Chapter 2 and Chapter 3); Dr. Michael Weather for my third project (part of Chapter 3). In particular, part of my third project was done during my internship at Matsushita Lab in Osaka University, hosted by Prof. Yasuyuki Matsushita.

I would like to show my sincere thanks to my colleagues and friends: Dr. Xiaolong Zhu, Dr. Xiao Tan, Dr. Kai Han, Dr. Wei Liu, Mr. Chaofeng Chen, Mr. Zhenfang Chen, Ms. Jingjing Zhang, Ms. Bingbin Liu, Mr. Huiquan Zhou, and Ms. Wenqi Yang, for their infectious enthusiasm in helping me in both studying and living. I also feel grateful for my friends Dr. Miaomiao Liu, Dr. Zhanghui Kuang, Dr. Xuhui Jia, Dr. Hao Zhou, Dr. Xingdou Fu, Dr. Guanbin Li, Dr. Zhen Li, Dr. Weifeng Ge, and Dr. Chaowei Fang.

In the summer of 2019, I was fortunate to have an internship at Matsushita Lab in Osaka University. I would like to thank Prof. Yasuyuki Matsushita, Dr. Michael Weather, Mr. Heng Guo, Mr. Xu Cao, Mr. Feiran Li, Mr. Hiroaki Santo, Mr. Kenji Enomoto, and Mr. Zhuoyu Yang for their help in both my research and living in Osaka.

I would like to express my heartfelt gratitude to The University of Hong Kong Foundation for Educational Development and Research ('HKU Foundation') for its generous support for my research studies in HKU.

Last, I want to thank my parents and beloved one Huijun Li for their unconditional love, support, and trust.

Abstract of thesis entitled

“Single View Analysis of Non-Lambertian Objects Based on Deep Learning”

Submitted by

Guanying Chen

for the degree of Doctor of Philosophy

at The University of Hong Kong

in December, 2020

Non-Lambertian objects (*e.g.*, transparent objects and specular objects) are very common in the real-world. However, existing computer vision algorithms developed for scene analysis often assume a Lambertian reflectance model, and treat non-Lambertian objects as outliers. It is important to develop robust methods for analysing non-Lambertian objects as it enables more complete and accurate understanding of the captured scene. This thesis tackles three vision problems of non-Lambertian objects under a single viewpoint, namely transparent object matting, calibrated photometric stereo, and uncalibrated photometric stereo for non-Lambertian objects.

The first part of this thesis addresses the problem of transparent object matting. Existing approaches often require tedious capturing procedures and long processing time, which limit their practical use. We formulate transparent object matting as a refractive flow estimation problem, and propose a deep learning framework, named *TOM-Net*, for learning the refractive flow. At test time, TOM-Net takes a single image as input, and outputs a matte (consisting of an object mask, an attenuation mask and a refractive flow field) in a fast feed-forward pass. As no off-the-shelf dataset is available for transparent object matting, we create a large-scale synthetic dataset for training and capture a real dataset for testing. Besides, we show that our method can be easily extended to handle cases where a trimap or a background image is available.

The second part of this thesis addresses the problem of calibrated photometric

stereo for non-Lambertian surfaces. Existing approaches often adopt simplified reflectance models to make the problem more tractable, but this greatly hinders their applications on real-world objects. We propose a deep fully convolutional network, named PS-FCN, that takes an arbitrary number of images of a static object captured under different light directions with a fixed camera as input, and predicts a normal map of the object in a fast feed-forward pass. Our method does not depend on a pre-defined set of light directions during training and testing.

The third part of this thesis considers the problem of uncalibrated photometric stereo, where light directions are unknown at test time. Specifically, we focus on estimating light directions from the input images, through which we cast the problem of uncalibrated photometric stereo into a calibrated one. We first introduce a novel convolutional network, named LCNet, to estimate light directions from input images. Unlike previous approaches that heavily rely on assumptions of specific reflectances and light source distributions, our method is able to determine light directions of a scene with unknown arbitrary reflectances observed under unknown varying light directions. We then analyse what had been learned by LCNet to resolve the ambiguity in lighting estimation. Inspired by our observations, we further introduce a guided calibration network (GCNet) to estimate more accurate lightings. (446 words)

Contents

Contents	xi
List of Figures	xv
List of Tables	xvii
1 Introduction	1
1.1 Motivation	1
1.2 Contributions	3
1.3 Thesis Outline	4
2 Learning Transparent Object Matting	7
2.1 Introduction	7
2.2 Related Work	10
2.3 Matting Formulation	11
2.4 Learning Transparent Object Matting	14
2.4.1 Encoder-Decoder for Coarse Prediction	15
2.4.2 Loss Function for Coarse Stage	16
2.4.3 Residual Learning for Matte Refinement	18
2.4.4 Improvement with Trimap and Background Image	19
2.5 Dataset for Learning and Evaluation	20
2.5.1 Synthetic Dataset	21
2.5.2 Real Dataset	24

2.6	Experimental Results	24
2.6.1	Ablation Study for Network Architecture	25
2.6.2	Evaluation on Synthetic Data	28
2.6.3	Evaluation on Real Data	32
2.6.4	Transparent Object Editing by Manipulating Environment Matte	35
2.6.5	Failure Cases	36
2.6.6	Improvement with Trimap and Background Image	37
2.7	Discussion	39
2.7.1	Limitations	39
2.7.2	Colored Objects and Specular Highlights	41
2.7.3	Difficulty in Comparison with Previous Works	41
2.7.4	Generalization to Real Data	42
2.7.5	Design of the Network Architecture	43
2.8	Conclusion	43
3	Learning Photometric Stereo	45
3.1	Introduction	45
3.2	Related Work	47
3.3	Image Formulation Model	48
3.4	A Flexible Learning Framework for Photometric Stereo	49
3.4.1	Max-pooling for Multi-feature Fusion	50
3.4.2	Network Architecture	51
3.4.3	Data Normalization for Handling Surfaces with SVBRDFs	53
3.5	Dataset for Learning and Evaluation	56
3.5.1	Synthetic Data for Training	56
3.5.2	Synthetic Data for Analysis	58
3.5.3	Real Data for Testing	59
3.6	Experimental Results	59
3.6.1	Evaluation on Synthetic Data	60
3.6.2	Evaluation on Real Data	67

3.6.3	Extension for Uncalibrated Photometric Stereo	70
3.7	Conclusion	71
4	Learning Lighting Calibration for Photometric Stereo	73
4.1	Introduction	73
4.2	Related Work	75
4.3	Lighting Calibration Network (LCNet)	76
4.3.1	Discretization of Lighting Space	77
4.3.2	Local-global Feature Fusion	77
4.3.3	Network Architecture	79
4.3.4	Training Data	80
4.3.5	Evaluation of LCNet with Synthetic Data	81
4.4	Analyzing What is Learned in LCNet	85
4.4.1	Lambertian Surfaces and the GBR Ambiguity	86
4.4.2	LCNet and the GBR Ambiguity	88
4.4.3	Feature Analysis for LCNet	90
4.5	Guided Calibration Network (GCNet)	91
4.5.1	Guided Feature Extraction	91
4.5.2	Network Architecture	92
4.6	Experimental Results	95
4.6.1	Evaluation on Synthetic Data	96
4.6.2	Evaluation on Real Data	99
4.6.3	Failure Cases	103
4.7	Conclusion	104
5	Conclusions	105
5.1	Summary	105
5.2	Future Work	106
	References	109

List of Figures

1.1	Example result of the traditional matting method for transparent object	2
1.2	Illustration of photometric stereo	3
2.1	Learning transparent object matting	8
2.2	Network architecture of TOM-Net	15
2.3	Examples of synthetic data	21
2.4	Sample images in real dataset	23
2.5	Qualitative comparison of different model variants	26
2.6	Visualization of the effectiveness of the refinement stage on real data .	27
2.7	Qualitative results on synthetic data (part 1)	30
2.8	Qualitative results on synthetic data (part 2)	31
2.9	Qualitative results on real data	33
2.10	Comparison of the photograph and composite	34
2.11	Image editing by manipulating the predicted environment matte	35
2.12	Failure cases on real data	36
2.13	Qualitative comparison on real data	38
2.14	Results on colored object and objects under natural illumination	40
3.1	Learning photometric stereo	46
3.2	Multi-feature fusion with max-pooling and average-pooling	50
3.3	Network architecture of PS-FCN.	52
3.4	Comparison between PS-FCN and PS-FCN ^{+N} on CAT with SVBRDF .	54
3.5	Illustration of the introduced data normalization operation	55

3.6	Examples of the synthetic training data.	57
3.7	Illustration of the synthetic test dataset $\text{SynTest}^{\text{MERL}}$	58
3.8	Lighting distributions of the real testing datasets	59
3.9	Visualization of the learned feature map after fusion	62
3.10	Effect of the input image number	62
3.11	Illustration of how max-pooling fusion layer handles cast shadow	64
3.12	Comparison between PS-FCN and PS-FCN ^{+N} on $\text{DRAGON}^{\text{SVBRDF}}$ dataset	65
3.13	Quantitative results on SPHERE rendered with 100 MERL BRDFs . . .	66
3.14	Qualitative results on HARVEST in the DiLiGenT benchmark	68
3.15	Qualitative results on Light Stage Data Gallery	69
3.16	Qualitative results on Gourd&Apple dataset	70
4.1	Illustration of an example discretization of the lighting space	77
4.2	Network architecture of LCNet	78
4.3	Results of LCNet under different light direction space discretization levels	82
4.4	Results of LCNet on $\text{SynTest}^{\text{MERL}}$ dataset with varying image numbers	84
4.5	Network architectures of UPS-FCN and UPS-FCN _{deep+mask}	86
4.6	Results of PF14 and LCNet on shapes under different GBR transformation	87
4.7	Feature visualization of LCNet on a non-Lambertian SPHERE	89
4.8	Network architecture of GCNet	93
4.9	Three different cascaded structures	97
4.10	Visualization of the estimated lighting distributions	101
4.11	Visual comparisons of normal estimation for POT1 and GOBLET	102
4.12	Visual comparison of normal estimation for the Light Stage Data Gallery's STANDING KNIGHT.	103
4.13	Failure cases	103

List of Tables

2.1	Comparison of different environment matting methods	12
2.2	Statistics of our synthetic datasets	22
2.3	Statistics of our real dataset	23
2.4	Ablation study for TOM-Net	25
2.5	Quantitative results on the synthetic test dataset	28
2.6	Quantitative results on real data	32
2.7	User study results	34
2.8	Quantitative comparison on the synthetic test dataset	37
2.9	Quantitative comparison on real data	37
3.1	Normal estimation results of PS-FCN on SynTest ^{MERL} dataset	61
3.2	Results on BUNNY rendered using three different lighting distributions	63
3.3	Quantitative comparison on the DiLiGenT benchmark	67
3.4	Runtime comparison of different methods	70
3.5	Results for uncalibrated photometric stereo on the DiLiGenT benchmark	71
4.1	Lighting estimation results of LCNet on SynTest ^{MERL} dataset	83
4.2	Results on SPHERE and BUNNY under different lighting distributions .	83
4.3	Normal estimation results on SynTest ^{MERL} dataset	85
4.4	Results of PF14 and LCNet on a SPHERE rendered with different BRDFs	88
4.5	Results of LCNet trained with different inputs	90
4.6	Ablation study for network architecture of GCNet	96
4.7	Normal estimation results on SynTest ^{MERL} dataset	96

4.8	Results on ARMADILLO under three different lighting distributions . . .	98
4.9	Lighting estimation results on BUNNY rendered with SVBRDFs	98
4.10	Lighting estimation results on surface regions cropped from BUNNY . .	99
4.11	Lighting estimation results on DiLiGenT benchmark	99
4.12	Lighting estimation results on Light Stage Data Gallery	100
4.13	Normal estimation results on DiLiGenT benchmark	101

Chapter 1

Introduction

1.1 Motivation

Over the past few decades, many algorithms have been developed for different vision problems, *e.g.*, multi-view stereo [1, 2], optical flow estimation [3, 4], shape-from-shading [5, 6], image matting [7, 8], and photometric stereo [9, 10]. However, these methods often rely on the assumption of a Lambertian surface, and treat the non-Lambertian objects (*e.g.*, transparent and specular objects) as outliers.

In fact, transparent and specular objects are very common in the real-world (*e.g.*, glass, plastic, and metallic surfaces). Simply treating them as outliers cannot thoroughly solve the underlying problems. It is important to develop robust methods for analyzing non-Lambertian objects as it enables a more complete and accurate understanding of the captured scene.

Different from a Lambertian surface whose observed brightness is the same from different viewing angles, the appearance of a non-Lambertian surface depends on how it interacts with the environment lights. The appearance of a transparent object is determined by how it refracts, reflects, absorbs, and scatters the incident light. The appearance of a specular object is determined by how it reflects, absorbs, and scatters the incident light. The complex interaction between the object and environment light increases the difficulties of analyzing transparent and specular objects.

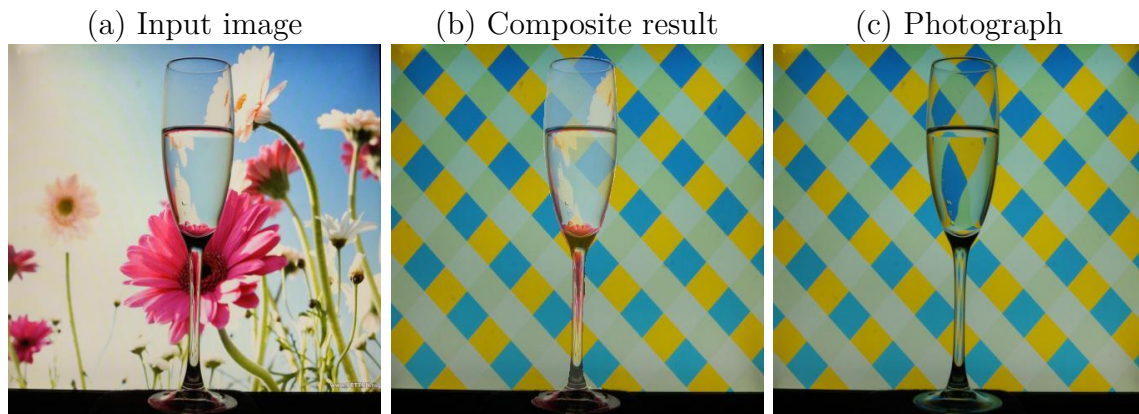


Fig. 1.1 Example result of the traditional image matting method for transparent object. (a) Input image. (b) Composite result obtained by applying the estimated alpha matte on a novel background. (c) Captured photo of the transparent object in front of the novel background.

Inspired by the great successes of deep learning based methods in various vision tasks [11–13], we propose to take advantage of the powerful feature learning capability of convolutional neural networks (CNNs) and the large scale training data to solve the difficult problems involving non-Lambertian objects. Specifically, in this thesis, we focus on three vision problems, namely transparent object matting, calibrated photometric stereo, and uncalibrated photometric stereo for non-Lambertian objects.

Traditional image matting methods are tailored for opaque objects. Given an input image, they estimate an object opacity for each pixel (*i.e.*, alpha matte), then the foreground regions can be extracted and composited onto a novel background. However, object opacity cannot model the refractive effect of a transparent object and thus leads to unrealistic composites (see Fig. 1.1 for an example). Existing methods for transparent object matting often require tedious capturing procedures and long processing time, which limit their practical use. To address the limitation of the existing alpha matting methods, we introduce a simple and effective deep learning framework to tackle the problem of transparent object matting.

Photometric stereo aims at recovering the surface normal of a static object from a set of images captured under different light directions [9, 14] (see Fig. 1.2). Compared with multi-view stereo methods, photometric stereo methods use monocular shading

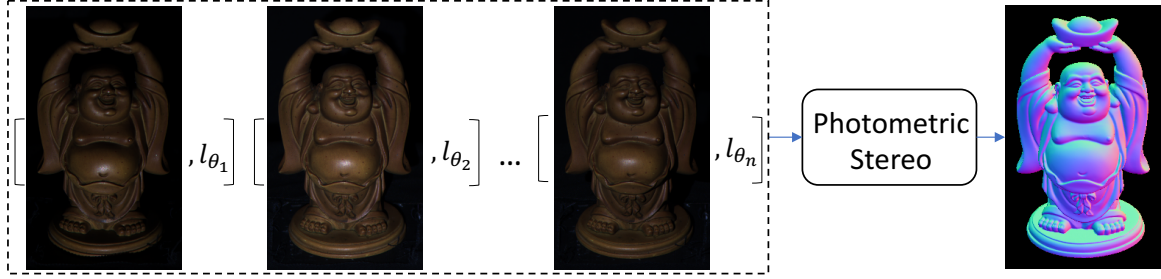


Fig. 1.2 Given multiple images of a static object captured under different light directions, photometric stereo can estimate a surface normal map of the object.

cues and naturally avoid the difficult correspondence problem. The advantage of photometric stereo is that it can handle specular and textureless surfaces, and can recover highly detailed scene geometry. However, there are still some limitations in existing photometric stereo methods. (i) Traditional methods often adopt simplified reflectance models to simplify the problem, and this greatly hinders their applications to real-world objects. (ii) Photometric stereo methods typically require calibrated lightings, and the calibration process is often very tedious. There are a few works for uncalibrated photometric stereo, but their performances are far behind the calibrated ones.

In this thesis, we address both these two limitations of existing photometric stereo methods. First, we develop a flexible deep learning framework for calibrated photometric stereo. By directly learning a mapping from intensity observations to surface normal, our model can bypass the need for explicitly modeling the surface reflectance model of the object. Second, we develop a deep learning method to estimate light directions from the input images, through which we cast the problem of uncalibrated photometric stereo into a calibrated one.

1.2 Contributions

The main contributions of this thesis can be summarized as follows:

- a convolutional neural network for **transparent object matting from a sin-**

gle image. We introduce a simple and efficient model for transparent object matting as simultaneous estimation of an object mask, an attenuation mask, and a refractive flow field. To train and evaluate the proposed method, we create a large-scale synthetic dataset and capture a real dataset as a benchmark for this problem. Preliminary results of this research have been published in [15, 16]

- a flexible convolutional neural network for **calibrated photometric stereo.** Our method directly learns the mapping from reflectance observations to surface normals, and does not depend on a pre-defined set of light directions during training and testing. We introduce two synthetic datasets for learning photometric stereo. Our method outperforms existing methods on multiple real datasets, which demonstrates the effectiveness of the proposed method. Preliminary results of this research have been published in [17, 18].
- a convolutional neural network for **estimating light directions for uncalibrated photometric stereo.** We analyze the features learned by our method, and find that attached shadows, shadings, and specular highlights are key elements for lighting estimation. Based on our findings, we propose an improved method that explicitly utilizes object shape and shading information as guidances for better lighting estimation. Preliminary results of this research have been published in [19, 20].

1.3 Thesis Outline

The remainder of this thesis is organized as follows.

Chapter 2 This chapter addresses the problem of transparent object matting. Existing approaches often require tedious capturing procedures and long processing time, which limit their practical use. In this chapter, we formulate transparent object matting as a refractive flow estimation problem, and propose a deep learning framework,

named *TOM-Net*, for learning the refractive flow. As no off-the-shelf dataset is available for transparent object matting, we introduce a large-scale synthetic dataset for training, and capture a real dataset for evaluation. We then show that our method can be extended to handle cases where a trimap or a background image is available. Promising experimental results have been achieved on both synthetic and real data, which clearly demonstrate the effectiveness of our approach.

Chapter 3 This chapter addresses the problem of calibrated photometric stereo for non-Lambertian surfaces under directional lightings. Existing approaches often adopt simplified reflectance models to make the problem more tractable, but this greatly hinders their applications on real-world objects. We propose a deep fully convolutional network, named PS-FCN, that takes an arbitrary number of images of a static object captured under different light directions with a fixed camera as input, and predicts a normal map of the object in a fast feed-forward pass. As obtaining ground-truth normal maps of real objects is difficult and time-consuming, we introduce two realistic synthetic datasets for training. Extensive experiments show that PS-FCN outperforms existing approaches in calibrated photometric stereo.

Chapter 4 This chapter addresses the problem of lighting estimation for uncalibrated photometric stereo. Previous approaches for this problem often heavily rely on assumptions of specific reflectances and light source distributions. We first introduce a lighting calibration network, named *LCNet*, that takes an arbitrary number of images as input and estimates their corresponding light directions and intensities. Surprised by the incredible effectiveness of *LCNet*, we analyze the features learned by this method and find that they strikingly resemble attached shadows, shadings, and specular highlights. Based on this insight, we propose a guided calibrated network, named *GCNet*, that explicitly leverages object shape and shading information for improved lighting estimation. Our experiments show that combining our network with existing calibrated photometric stereo methods yields significantly improved results over state-of-the-art uncalibrated methods.

Chapter 5 This chapter summarizes the theories and algorithms developed in this dissertation, followed by a brief discussion of potential future work.

Chapter 2

Learning Transparent Object Matting

2.1 Introduction

Image matting refers to the process of extracting the foreground matte of an image by locating the region of the foreground object and estimating the opacity of each pixel inside the foreground region. The foreground object can then be composited onto a new background image using the *matting equation* [7]

$$C = F + (1 - \alpha)B, \quad \alpha \in [0, 1], \quad (2.1)$$

where C denotes the composited color, F the foreground color, B the background color, and α the opacity.

Image matting has been widely used in image editing and film production. However, most of the existing methods are tailored for opaque objects, and cannot handle transparent objects whose appearance depends on how light is refracted from the background.

To model the effect of refraction, Zongker *et al.* [21] introduced *environment mat-*

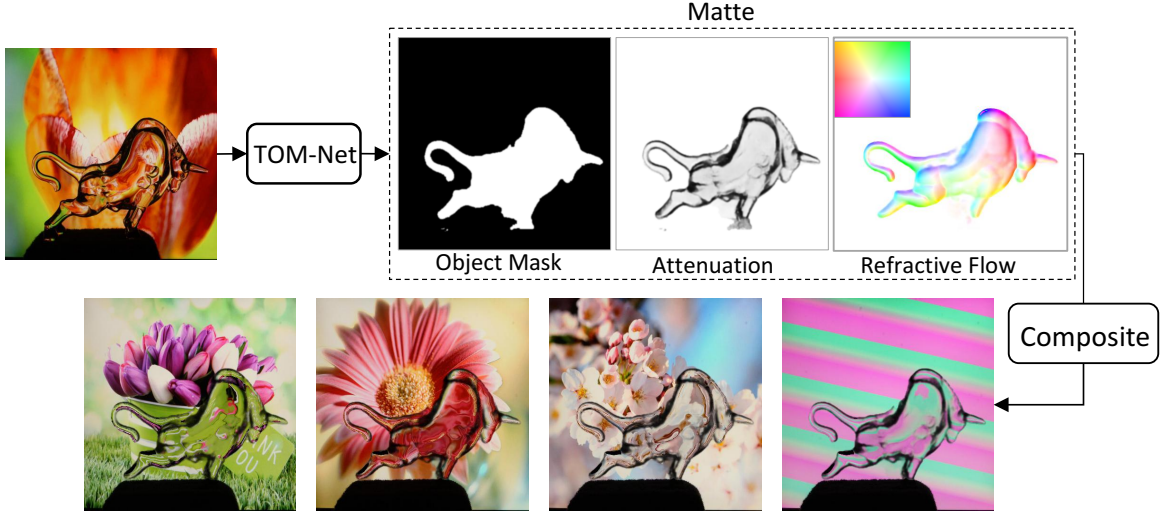


Fig. 2.1 Learning transparent object matting. Given an image of a transparent object as input, our model can estimate the environment matte (consisting of an object mask, an attenuation mask, and a refractive flow field) in a feed-forward pass. The transparent object can then be composited onto new background images with the extracted matte.

ting as

$$C = F + (1 - \alpha)B + \Phi, \quad \alpha \in [0, 1], \quad (2.2)$$

where Φ is the contribution of environment light caused by refraction or reflection at the foreground object. Besides estimating the foreground shape, environment matting also describes how objects interact with the background.

Many efforts [22–27] have been devoted to improving the seminal work of [21]. The resulting methods often require either a huge number of input images to achieve a higher accuracy, or the use of specially designed patterns to reduce the number of required images. They are in general all very computational expensive.

In this work, we focus on environment matting for transparent objects. It is highly ill-posed, if not impossible, to estimate an accurate environment matte for transparent objects from a single image with an arbitrary background. Given the huge solution space, there exist multiple objects and backgrounds which can produce the same refractive effect. In order to make the problem more tractable, we simplify our problem to estimating an environment matte that can produce visually realistic refractive ef-

fect from a single image, instead of estimating a highly accurate refractive flow. We define the environment matte in our model as a triplet consisting of an object mask, an attenuation mask, and a refractive flow field. Realistic refractive effect can then be obtained by compositing the transparent object onto new background images (see Fig. 2.1). We then show that the performance of the proposed method can be improved when a trimap or a background image is available.

Inspired by the great successes of convolutional neural networks (CNNs) in high-level computer vision tasks, we propose a convolutional neural network, called TOM-Net, for simultaneous learning of an object mask, an attenuation mask, and a refractive flow field from a single image with an arbitrary background. The key contributions of this work can be summarized as follows:

- We introduce a simple and efficient model for transparent object matting as simultaneous estimation of an object mask, an attenuation mask, and a refractive flow field.
- We propose a convolutional neural network, TOM-Net, to learn an environment matte of a transparent object from a single image. To the best of our knowledge, TOM-Net is the first CNN that is capable of learning transparent object matting.
- We create a large-scale synthetic dataset and a real dataset as a benchmark for learning transparent object matting. Our TOM-Net has produced promising results on both the synthetic and real datasets.
- We propose two additional convolutional neural networks, denoted as TOM-Net^{+Trimap} and TOM-Net^{+Bg}, for handling the cases where a trimap or a background image is available, respectively.

Preliminary results of this chapter were published in [15, 16]. Our code, trained models, and datasets can be found at <https://guanyingc.github.io/TOM-Net>.

2.2 Related Work

In this section, we briefly review representative works on environment matting and recent works on CNN based image matting.

Environment matting Zongker *et al.* [21] introduced the concept of environment matting, and assumed each foreground pixel being originated from a single rectangular region of the background. They obtained the environment matte by identifying the corresponding background region for each foreground pixel using three monitors and multiple images. Chuang *et al.* [22] extended [21] in two different ways. First, they replaced the single rectangular supporting area for a foreground pixel with multiple 2D oriented Gaussian strips. This makes it possible for their method to model the effects of color dispersion, multiple mapping, and glossy reflection. Second, they simplified the environment matting equation by assuming the object being colorless and perfectly transparent. This allows them to achieve real time capture environment matting (RTCEM). The environment matte was then extracted with one image taken in front of a pre-designed pattern. However, RTCEM requires background images to segment the transparent objects, and depends on a time-consuming off-line processing.

Wexler *et al.* [23] introduced a probabilistic model based method which assumes each background point has a probability to make contribution towards the color of a certain foreground point. Their approach does not require pre-designed patterns during data acquisition, but it still needs multiple images and can only model thin transparent objects. Peers and Dutré [24] used a large number of wavelet basis backgrounds to obtain the environment matte, and their method can also model the effect of diffuse reflection. Based on the fact that a signal can be decomposed uniquely in the frequency domain, Zhu and Yang [25] proposed a frequency-based approach to extract an accurate environment matte. They used Fourier analysis to solve the decomposition problem. Both [24] and [25] require a large number of images to extract the matte (*e.g.*, [24] needs 2,400 images and [25] needs 4,096 images for an image of size 1024×1024), making them not very practical. Recently, compressive sensing theory

has been applied to environment matting to reduce the number of images required. Duan *et al.* [28] applied this theory in the spatial domain and Qian *et al.* [29] applied it in the frequency domain. However, the number of images needed is still in the order of hundreds. In contrast, our work can estimate an environment matte from a single image in a fast feed-forward computation without the need for pre-designed patterns or additional background images.

Yeung *et al.* [30] proposed an interactive way to estimate an environment matte given an image containing a transparent object. Their method requires users to manually mark the foreground and background in the image, and models the refractive effect using a thin-plate-spline transformation. Their method does not produce an accurate environment matte, but instead a visually pleasing refractive effect. Our method shares the same spirit, but does not involve any human interaction.

Table 2.1 shows a comparison of different environment matting methods. Compared with other methods, our method requires only a single image and can extract a matte in 0.5 second without the need for any predefined backgrounds.

CNN based image matting Although the potential of CNN on transparent object matting has not yet been explored, some existing work have adopted CNNs for solving the traditional image matting problem. Shen *et al.* [32] introduced a CNN for image matting of color portrait images. Cho *et al.* [33] proposed a network to predict a better alpha matte by taking the matting results of the traditional method and normalized color images as input. Some deep learning methods [34–36] have been introduced to estimate an alpha matte given an image and its trimap. However, none of these methods can be applied directly to the task of transparent object matting as object opacity alone is not sufficient to model the refractive effect.

2.3 Matting Formulation

As a transparent object may have multiple optical properties (*e.g.*, color attenuation, translucency, and reflection), estimating an accurate environment matte for a generic

Table 2.1 Comparison of different environment matting methods. k indicates the image size and mapping type stands for how a foreground point is composited by the point(s) in the background image.

Methods	Asymptotic # images	# images ($k = 1024$)	Typical runtime ($k = 1024$)	Mapping type	Materials	Remarks
Ours	$O(1)$	1	0.5 secs when $k = 512$ (run on a GPU)	single-pixel	colorless, specularly refractive	aims for visually realistic effect
RTCEM [22]	$O(1)$	1	2 mins	single-pixel	colorless, specularly refractive	requires a coded background and off-line processing
Yeung et. al [30]	$O(1)$	1	30 secs	single-pixel	colored refractive	requires human interaction, aims for visually realistic effect
Zongker et al. [21]	$O(\log k)$	20	20 mins when $k = 512$	single-region	colored refractive, translucent, highly specular	assumes rectangular support region
Chuang et al. [22]	$O(k)$	1800	not available	multi-region	Zongker et al. [21] + (color dispersion, multiple mapping, glossy reflection)	requires solving a complex optimization problem
Wavelet [31]	$O(k)$	2400	12 hours	multi-region	same as Chuang et al. [22]	runtime includes data acquisition
Frequency [25]	$O(k)$	4096	5 – 10 mins	multi-pixel	Zongker et al. [21] + (color dispersion, glossy reflection)	slow data acquisition
Duan et al. [28]	$O(s \log(k^2/s))$	340	2.8 mins	multi-region	same as Chuang et al. [22]	s denotes the sparsity of a signal
Qian et al. [29]	$O(s \log(2k/s))$	400	3.3 mins	multi-pixel	same as Frequency [25]	s denotes the sparsity of a signal

transparent object from a single image is very challenging. Following the work of [22], we cast environment matting to a refractive flow estimation problem by assuming that each foreground pixel only originates from one point in the background due to refraction. Compared to the seminal work of [21], which models each foreground pixel as a linear combination of a patch in the background, our formulation is more tractable and can be easily encoded using a CNN.

In [21], the per-pixel environment matting is obtained through leveraging color information from multiple background images. Given a set of pre-designed background patterns, matting is formulated as

$$C = F + (1 - \alpha)B + \sum_{i=1}^k R_i \mathcal{M}(\mathbf{T}_i, \mathbf{A}_i), \quad (2.3)$$

where F , B and α denote the ambient illumination, background color and opacity, respectively. The last term in Eq. (2.3) accounts for the environment light accumulated from k pre-designed background images ($k = 3$ in [21]). R_i is a factor describing the contribution of light emanating from the i -th background image \mathbf{T}_i . $\mathcal{M}(\mathbf{T}_i, \mathbf{A}_i)$ denotes the average color of a rectangular region \mathbf{A}_i on the background image \mathbf{T}_i .

To obtain an environment matte, the transparent object is placed in front of the monitor(s), and multiple pictures of the object are captured with the monitor(s) displaying different background patterns¹. Generally, a surface point receives light from multiple directions, especially for a diffuse surface. When it comes to a perfectly transparent object, however, a surface point will only receive light from one direction as determined by the law of refraction. Consider a single background image as the only light source (*i.e.*, no ambient illumination), the problem can be modeled as

$$C = (1 - \alpha)B + R\mathcal{M}(\mathbf{T}, P), \quad (2.4)$$

where $\mathcal{M}(\mathbf{T}, P)$ is a bilinear sampling operation at location P on the background image \mathbf{T} . Further, by assuming a colorless transparent object, R becomes a light

¹For an image of size 512×512 , 18 pictures and around 20 minutes processing time are needed.

attenuation index ρ (a scalar value). The formulation in Eq. (2.4) can be simplified to

$$C = (1 - \alpha)B + \rho\mathcal{M}(\mathbf{T}, P), \quad (2.5)$$

where $\rho \in [0, 1]$ denotes the attenuation index.

Here, we use refractive flow to model the refractive effect of a transparent object. The refractive flow of a foreground pixel is defined as the offset between the foreground pixel and its refraction correspondence on the background image.

We further introduce a binary foreground mask to define the object region in the image. The matting equation can now be rewritten as

$$C = (1 - m)B + m\rho\mathcal{M}(\mathbf{T}, P), \quad (2.6)$$

where $m \in \{0, 1\}$ denotes background ($m = 0$) or foreground ($m = 1$). The matte can then be estimated by solving m , ρ and P for each pixel in the input image containing the transparent object².

2.4 Learning Transparent Object Matting

In this section, we present a two-stage deep learning framework, called TOM-Net, for learning transparent object matting (see Fig. 2.2). The first stage, denoted as CoarseNet, is a multi-scale encoder-decoder network that takes a single image as input, and predicts an object mask, an attenuation mask, and a refractive flow field simultaneously. CoarseNet is capable of predicting a robust object mask. However, the estimated attenuation mask and refractive flow field lack local structural details. To overcome this problem, we introduce the second stage of TOM-Net, denoted as RefineNet, to achieve a sharper attenuation mask and a more detailed refractive flow field. RefineNet is a residual network [13] that takes both the input image and the output of CoarseNet as input. After training, our TOM-Net can predict an environment

²For an image with n pixel, we have 7 unknowns (3 for B , 2 for P , 1 for m , and 1 for ρ) for each pixel, resulting in a total of $7n$ unknowns.

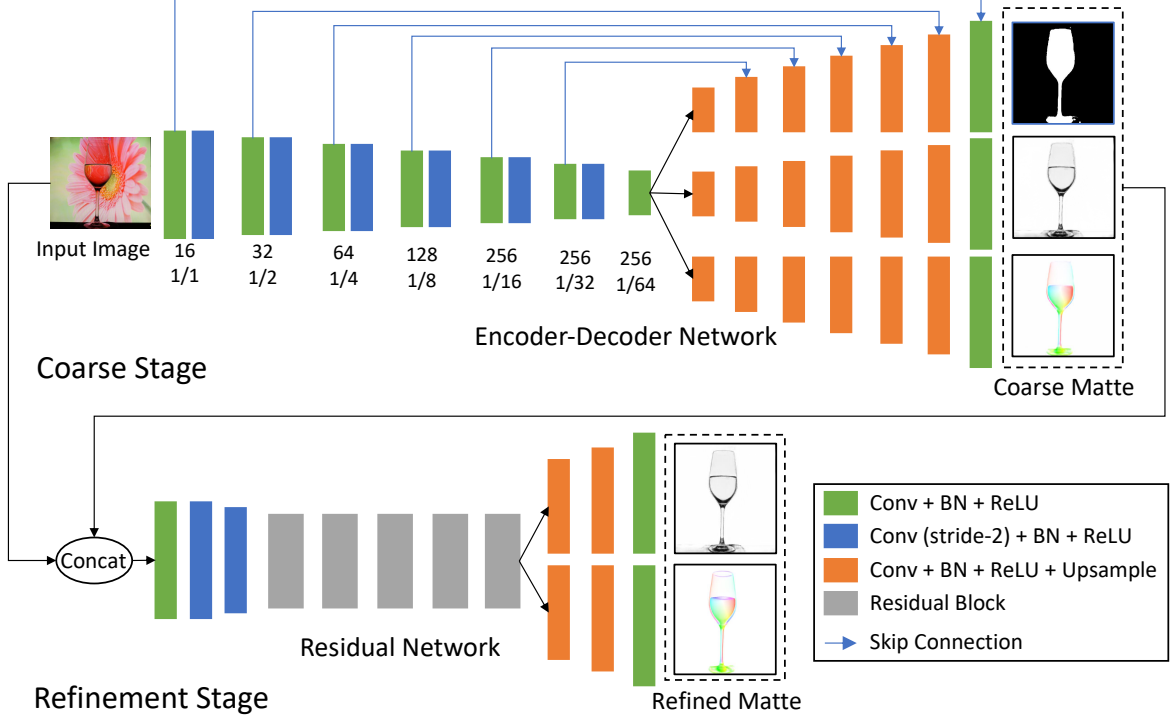


Fig. 2.2 Network architecture of TOM-Net. The upper subnetwork is the CoarseNet and the bottom subnetwork is the RefineNet. (Cross-link and multi-scale outputs are not shown for simplicity.)

matte from a single image in a fast feed-forward pass.

2.4.1 Encoder-Decoder for Coarse Prediction

The first stage of our TOM-Net (*i.e.*, CoarseNet) is based on mirror-link CNN introduced in [37]. Mirror-link CNN was proposed to learn non-Lambertian object intrinsic decomposition. Its output consists of an albedo map, a shading map, and a specular map. It shares a similar output structure with our transparent object matting task (*i.e.*, three output branches sharing the same spatial dimensionality). Therefore, it is reasonable for us to adapt mirror-link CNN for our CoarseNet.

The mirror-link CNN adapted for our CoarseNet consists of one shared encoder and three distinct decoders. The encoder contains six down-sampling convolutional blocks, leading to a down-sampling factor of 64 in the bottleneck layer. Features in the encoder layers are connected to the decoder layers having the same spatial

dimensions through skip connections [38]. Cross-links [37] are introduced to make different decoders share the same input in each layer, so that decoders can better utilize the correlation between different predictions.

Learning with multi-scale loss has been proven to be helpful in dense prediction tasks (*e.g.*, [39, 40]). Since we formulate the problem of transparent object matting as refractive flow estimation, which is a dense prediction task, we augment our mirror-link CNN with multi-scale loss similar to [40]. We use four different scales in our model, where the first scale starts from the decoder features with a down-sampling factor of 8 and the largest scale has the same spatial dimensions as the input.

In contrast to the recent two-stage framework for image matting [34], our TOM-Net has a shared encoder and three parallel decoders to accommodate different outputs. Besides, we augment our CoarseNet with multi-scale loss and cross-link. Moreover, TOM-Net is trained from scratch while the encoder in [34] is initialized with the pre-trained VGG16.

2.4.2 Loss Function for Coarse Stage

CoarseNet takes a single image as input and predicts the environment matte as a triplet consisting of an object mask, an attenuation mask, and a refractive flow field. The learning of CoarseNet is supervised by the ground-truth matte using an **object mask segmentation loss** \mathcal{L}_{ms} , an **attenuation regression loss** \mathcal{L}_{ar} , and a **refractive flow regression loss** \mathcal{L}_{fr} . Besides, the predicted matte is expected to render an image as close to the input image as possible when applied to the ground-truth background based on Eq. (2.6). Hence, in addition to the supervision of the matte, we also take **image reconstruction loss** \mathcal{L}_{ir} into account (bilinear sampling is implemented following [41]). Note that the ground-truth background is only used to calculate the reconstruction error during training but not needed during testing. CoarseNet can therefore be trained by minimizing

$$\mathcal{L}^c = \alpha_{ms}^c \mathcal{L}_{ms} + \alpha_{ar}^c \mathcal{L}_{ar} + \alpha_{fr}^c \mathcal{L}_{fr} + \alpha_{ir}^c \mathcal{L}_{ir}, \quad (2.7)$$

where $\alpha_{ms}^c, \alpha_{ar}^c, \alpha_{fr}^c, \alpha_{ir}^c$ are weights for the corresponding loss terms.

Object mask segmentation loss Object mask segmentation is simply a spatial binary classification problem. The output of the object mask decoder has a dimension of $2 \times H \times W$, where H and W denote the height and width of the input. We normalize the output with *softmax* and compute the loss using the binary cross-entropy function

$$\mathcal{L}_{ms} = -\frac{1}{HW} \sum_{ij} (\tilde{M}_{ij} \log(P_{ij}) + (1 - \tilde{M}_{ij}) \log(1 - P_{ij})), \quad (2.8)$$

where $\tilde{M}_{ij} \in \{0, 1\}$ and $P_{ij} \in [0, 1]$ represent ground truth and normalized foreground probability of the pixel at (i, j) , respectively.

Attenuation regression loss The predicted attenuation mask has a dimension of $1 \times H \times W$. The value of this mask is in the range of $[0, 1]$, where 0 indicates no light can pass and 1 indicates the light will not be attenuated. We adopt a mean square error (MSE) loss

$$\mathcal{L}_{ar} = \frac{1}{HW} \sum_{ij} (A_{ij} - \tilde{A}_{ij})^2, \quad (2.9)$$

where A_{ij} is the predicted attenuation index and \tilde{A}_{ij} the ground truth at (i, j) .

Refractive flow regression loss The predicted refractive flow field has a dimension of $2 \times H \times W$, where we have one channel for the horizontal displacement and another for the vertical displacement. We normalize the refractive flow with *tanh* activation and multiply it by the width of the input, such that the output is constrained in the range of $[-W, W]$. We adopt an average end-point error (EPE) loss

$$\mathcal{L}_{fr} = \frac{1}{HW} \sum_{ij} \sqrt{(F_{ij}^x - \tilde{F}_{ij}^x)^2 + (F_{ij}^y - \tilde{F}_{ij}^y)^2}, \quad (2.10)$$

where (F^x, F^y) and $(\tilde{F}^x, \tilde{F}^y)$ denote the predicted flow and the ground truth, respectively.

Image reconstruction loss We use MSE loss to measure the dissimilarity between the reconstructed image and the input image. Denoting the reconstructed image by I and the ground-truth image (*i.e.*, the input image) by \tilde{I} , the reconstruction loss is given by

$$\mathcal{L}_{ir} = \frac{1}{HW} \sum_{ij} \|I_{ij} - \tilde{I}_{ij}\|_2^2. \quad (2.11)$$

Implementation details In all experiments, we empirically set $\alpha_{ms}^c = 0.1$, $\alpha_{ar}^c = 1$, $\alpha_{fr}^c = 0.01$, and $\alpha_{ir}^c = 1$. The loss weights for different scales are $\frac{1}{2^{(4-s)}}$, where $s \in \{1, 2, 3, 4\}$ denotes the scale. CoarseNet contains 8 million parameters and it takes about 2.5 days to train with Adam optimizer [42] on a single NVIDIA Titan X Pascal GPU. We first train the CoarseNet from scratch until convergence and then train the RefineNet.

2.4.3 Residual Learning for Matte Refinement

As the attenuation mask and the refractive flow field predicted by the CoarseNet lack structural details, a refinement stage is needed to produce a detailed matte. Observing that residual learning is particularly suitable for tasks whose input and output are largely similar [43, 44], we propose a residual network, denoted as RefineNet, to refine the matte predicted by the CoarseNet. Similar strategy has also been successfully applied to progressively refine the estimated optical flow in [45].

We concatenate the input image and the output of the CoarseNet to form the input of the RefineNet. As the object mask predicted by the CoarseNet is already plausible, the RefineNet only outputs an attenuation mask and a refractive flow field. The parameters of the CoarseNet are fixed when training the refinement stage.

Loss for the refinement stage The overall loss for the refinement stage is

$$\mathcal{L}^r = \alpha_{ar}^r \mathcal{L}_{ar} + \alpha_{fr}^r \mathcal{L}_{fr}, \quad (2.12)$$

where \mathcal{L}_{ar} is the refinement attenuation regression loss, \mathcal{L}_{fr} the refinement flow regression loss, and α_{ar}^r , α_{fr}^r their weights. The definitions of these two losses are identical to those defined in the first stage. We found that adding the image reconstruction loss in the refinement stage did reduce the image reconstruction error during training, but was not helpful in preserving sharp edges of the refractive flow field (*e.g.*, mouth of a glass). This could be explained by the fact that a lower image reconstruction loss does not guarantee a better refractive flow field. As the matte estimated by the CoarseNet has already achieved a small reconstruction error, simultaneously optimizing the flow regression loss and image reconstruction loss in the refinement stage may compromise the flow estimation. Since our goal in the refinement stage is to estimate a more detailed matte, we remove the image reconstruction loss to make our network focus on reducing the flow regression loss.

Implementation details We set $\alpha_{ar}^r = 1$, $\alpha_{fr}^r = 1$ for the refinement. RefineNet contains 1 million parameters and it takes about 2 days to train with Adam optimizer on a single NVIDIA Titan X Pascal GPU. RefineNet is randomly initialized during training.

2.4.4 Improvement with Trimap and Background Image

As the problem of transparent object matting from a single image is highly ill-posed, we investigate how to reinforce our framework by utilizing additional information. In particular, we consider the cases where a trimap or a background image is available. Our framework can be easily extended to make use of these additional information by taking the concatenation of the input image and the background image (or trimap) as input, while keeping the overall network architecture unchanged.

TOM-Net^{+Trimap} Trimap can provide a rough location of the transparent object to help the model better locate the transparent object. The trimap used in this work is a single channel image with 3 different values, where values 0, 1, and 2 indicate back-

ground, unknown, and foreground regions, respectively. During training, we randomly generate trimaps based on the ground-truth object mask. We first perform random erosion and cropping on the object mask to form the known (rough) foreground region. The unknown region is then generated by subtracting the foreground region from a tight bounding box of the object mask, leaving the rest of the regions as the background region. The variant model, denoted as TOM-Net^{+Trimap}, takes both the input image and trimap as input, giving rise to an input channel number of 4 in the first convolutional layer.

TOM-Net^{+Bg} Given the background image, the model can easily identify the accurate location of the transparent object based on the difference of the input and background images. Moreover, having access to the background image allows the model to better estimate the refractive flow field. The variant model, denoted as TOM-Net^{+Bg}, takes both the input and background images as input, giving rise to an input channel number of 6 in the first convolutional layer.

TOM-Net^{+Trimap} and TOM-Net^{+Bg} are trained with the same procedure as TOM-Net. Our experimental results show that with the additional information, our framework can achieve better results on both synthetic and real dataset.

2.5 Dataset for Learning and Evaluation

Currently there is no off-the-shelf dataset for transparent object matting. One potential direction is to create a real dataset with ground-truth mattes (*i.e.*, object masks, attenuation masks, and refractive flow fields) for training. However, it is almost impossible for human to manually label the refractive flow field of the transparent object. One may consider estimating the mattes using existing transparent object matting methods and using them as the ground truth for training. However, it is very difficult and tedious as traditional methods require a large number of images and/or a long processing time for each object. Besides, there is no publicly available code for transparent object matting. To bypass this problem, we created a large-scale synthetic

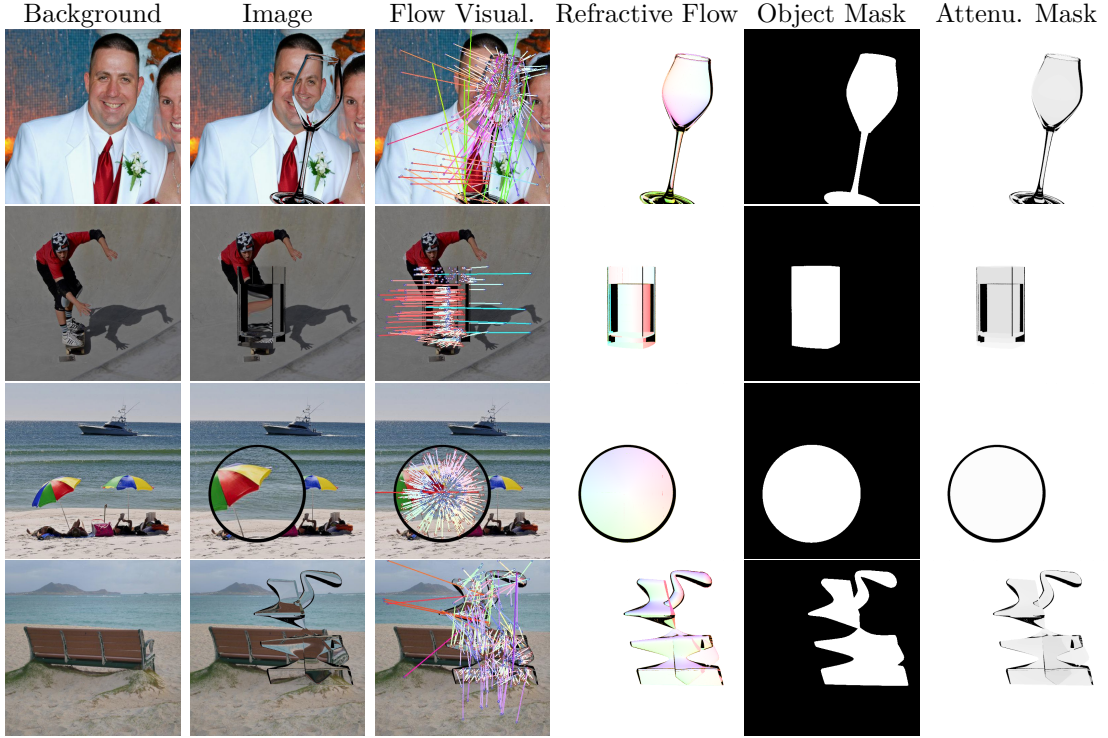


Fig. 2.3 Examples of synthetic data. Top to bottom: examples of *Glass*, *Glass with Water*, *Lens* and *Complex*, respectively. First three columns: background image, rendered image, refractive flow visualization (sparse). Last three columns: ground-truth refractive flow field, object mask, attenuation mask.

dataset using *POV-Ray* [46] to render images of synthetic transparent objects. Besides, we captured a real dataset for evaluation. We will show that our TOM-Net trained on the synthetic dataset can generalize well to real world objects, demonstrating its good transferability.

2.5.1 Synthetic Dataset

We used a large number of background images and 3D models to render our training samples. We randomly changed the pose of the models, as well as the viewpoint and focal length of the camera in the rendering process to avoid overfitting to a fixed setting.

Table 2.2 Statistics of our synthetic datasets.

Type	<i>Glass</i>	<i>Glass with Water</i>	<i>Lens</i>	<i>Complex</i>	Total
Synthetic Train	52K	26K	20K	80K	178K
Synthetic Test	250	250	200	200	900

Background images We employed two types of background images, namely scene images and synthetic patterns. For scene images, we randomly sampled images from the Microsoft COCO [47] dataset³. The background images for the synthetic training set are sampled from COCO Train2014 and Test2015, while that for the synthetic test dataset are from COCO Val2014, giving rise to 100K scene images in total. For synthetic patterns, we rendered 40K patterns of size 512×512 using *POV-Ray* built-in textures.

Transparent objects We divided common transparent objects into four categories, namely *Glass*, *Glass with water*, *Lens*, and *Complex* shape (see Fig. 2.3 for examples). We constructed parametric 3D models for the first three categories, and generated a large number of models using random parameters. For complex shapes, we constructed parametric 3D models for basic shapes like sweeping-spheres and squashed surface of revolution (SOR) parts, and composed a larger number of models using these basic shapes. We generated 178K 3D models in total, with each model assigned a random refractive index $\lambda \in [1.3, 1.5]$. The distribution of these models in four categories is shown in Table 2.2.

Ground-truth matte generation We obtained the ground-truth object mask of a model by rendering it in front of a black background image and setting its color to white. Similarly, we obtained the ground-truth attenuation mask of a model by simply rendering it in front of a white background image. Finally, we obtained the ground-truth refractive flow field (see Fig. 2.3) of a model by rendering it in front of a sequence of Gray-coded patterns. Technical details for the data rendering can be

³Other large-scale datasets like ImageNet [48] can also be used.

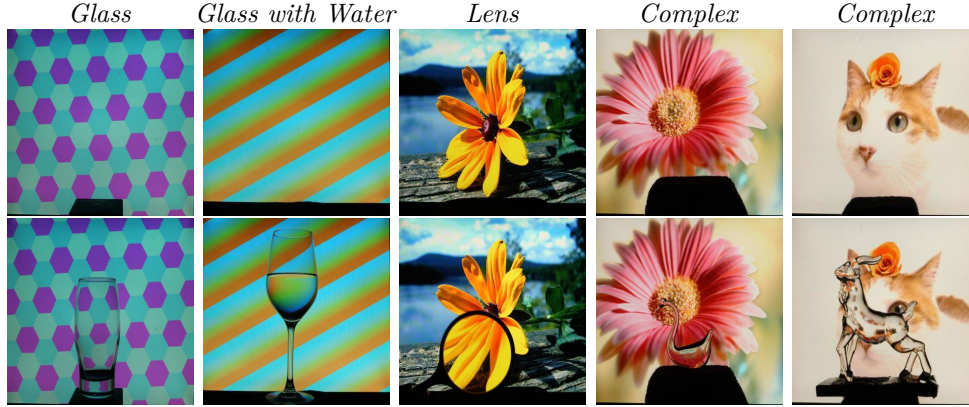


Fig. 2.4 Sample images in real dataset. The first row shows the background images and the second row shows the images of transparent objects.

Table 2.3 Statistics of our real dataset. The first and second rows show the number of objects and the number of backgrounds used during data acquisition, respectively. The last row shows the number of captured samples. Note that the category of *Glass with Water* are created by filling five of the glasses with different amount of water, and some backgrounds are shared between different shape categories.

	<i>Glass</i>	<i>Glass with Water</i>	<i>Lens</i>	<i>Complex</i>
# Objects	7	(5 glasses used)	1	6
# Backgrounds	60	38	4	18
# Samples	470	103	61	242

found at https://github.com/guanyingc/TOM-Net_Rendering.

Data augmentation To improve the diversity of the training data and narrow the gap between real and synthetic data, extensive data augmentation was carried out on-the-fly. For an image of size 512×512 with color intensity normalized to $[0, 1]$, we randomly performed color (brightness, contrast and saturation) augmentation (in a range of $[-0.2, 0.2]$), image scaling (in a range of $[0.875, 1.05]$), noise perturbation (in a range of $[-0.05, 0.05]$), and horizontal/vertical flipping. Besides, we also blurred the object boundary to make the synthetic data visually more natural. A patch with a size of 448×448 was then randomly cropped from an augmented image and used as input to train CoarseNet. To speed up the training and save memory, a smaller patch with a size of 384×384 was used to train RefineNet after the training of CoarseNet.

2.5.2 Real Dataset

To validate the transferability of TOM-Net, we introduce a real dataset, which was captured using 14 objects⁴ and 60 background images, resulting in a dataset of 876 images. Note that the background images for real data have not been used in the synthetic training or test dataset. The data distribution is summarized in Table 2.3. During the data capturing process, the objects were placed under different poses, with the distances between the camera, object, and background uncontrolled. Fig. 2.4 shows some sample images from the real dataset. Note that we do not have the ground-truth matte for the real dataset. We instead captured images of the backgrounds without the transparent objects to facilitate evaluation.

Following previous works, the transparent objects were captured in front of a monitor displaying different background images. Due to the sampling problem, there may exist Moiré-effect in the captured image. We carefully adjusted the focal length and shutter speed to remove the Moiré-effect during the data capturing.

2.6 Experimental Results

In this section, we present experimental results and analysis. We performed ablation study for TOM-Net, and evaluated our approach on both synthetic and real data. For synthetic data, we evaluated end-point error (EPE) for refractive flow fields, intersection over union (IoU) for object masks, mean square error (MSE) for attenuation masks and image reconstruction results, respectively. For real data, due to the absence of ground-truth matte, evaluation on the absolute error with respect to the ground truth is not possible. Instead, we reconstructed the input images using the estimated mattes and background images, and then evaluated the PSNR and SSIM metrics [31] between each pair of input image (*i.e.*, photograph) and reconstructed image (*i.e.*, composite). In addition, a user study was conducted to validate the realism of TOM-

⁴The objects consist of 7 glasses, 1 lens and 6 complex objects. Glasses with water are implicitly included.

Table 2.4 Ablation study for TOM-Net. F, A, I, and M are short for flow, attenuation, image reconstruction, and object mask, respectively. (The first value for EPE is measured on the whole image and the second measured within the object region. A-MSE and I-MSE are computed on the whole image.)

ID	Model Variants	F-EPE	A-MSE	I-MSE	M-IoU	
0	Background	6.5 / 41.0	1.58	0.87	0.15	
1	CoarseNet - (\mathcal{L}_{fr}^c)	3.9 / 26.5	0.24	0.23	0.98	
2	CoarseNet - (cross-link)	2.5 / 17.2	0.30	0.21	0.97	MSE ($\cdot 10^{-2}$)
3	CoarseNet - (multi-scale)	2.4 / 16.6	0.69	0.25	0.94	↓ better
4	CoarseNet - (\mathcal{L}_{ir}^c)	2.3 / 15.7	0.25	0.22	0.98	↑ better
5	CoarseNet	2.2 / 15.4	0.28	0.18	0.97	
6	CoarseNet + RefineNet	2.0 / 13.7	0.25	0.19	0.97	
7	CoarseNet + (RefineNet+ \mathcal{L}_{ir}^r)	2.0 / 13.9	0.24	0.18	0.97	

Net composites.

We showcased an application of image editing of transparent object by manipulating the extracted matte, and analyzed typical failure cases. We also investigated how the performance of our method can be improved when a trimap or a background image is available.

2.6.1 Ablation Study for Network Architecture

We quantitatively analyzed different components of TOM-Net using synthetic dataset⁵. We first verified the effectiveness of *refractive flow regression loss* (\mathcal{L}_{fr}^c), *cross-link*, *multi-scale loss* and *image reconstruction loss* (\mathcal{L}_{ir}^c) in the coarse stage by removing each of them from *CoarseNet* during training. We then validated the effectiveness of *RefineNet* in recovering details of the refractive flow field. RefineNet was evaluated by adding it to a trained CoarseNet and was trained while fixing the parameters of CoarseNet. For comparison, we also included a naive baseline, denoted as *Background*, by considering a zero matte case (*i.e.*, whole image as object mask, no attenuation, and no refractive flow) where the reconstructed image is the same as the background image. The quantitative results are summarized in Table 2.4 and the qualitative comparisons are shown in Fig. 2.5. Overall, the baseline *Background* was outperformed by

⁵Complex shape is excluded in experiments here to speed up training.

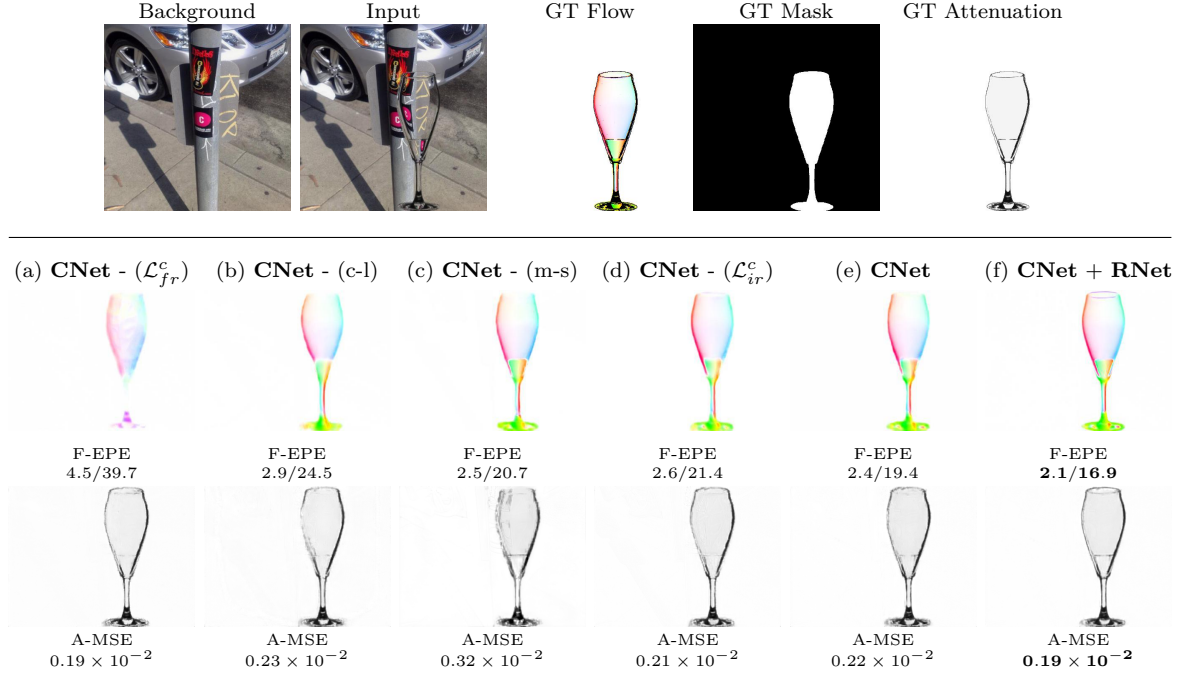


Fig. 2.5 Qualitative comparison of different model variants. The first row shows a sample of *Glass with Water* from the synthetic test dataset. The second and third rows show the estimated refractive flow fields and attenuation masks by different variants, respectively. (CNet and RNet are short for CoarseNet and RefineNet.)

all TOM-Net variants with a large margin for all the evaluation metrics, which clearly shows that TOM-Net can successfully learn the matte.

Effectiveness of refractive flow regression loss Comparing experiments with IDs 1 & 5 in Table 2.4, it can be clearly seen that the CoarseNet trained with the refractive flow regression loss significantly outperformed that without it in refractive flow estimation. This result indicates that image reconstruction loss alone is not enough to supervise the learning of refractive flow. Fig. 2.5 (a & e) qualitatively show that the refractive flow regression loss improved the performance of refractive flow estimation.

Effectiveness of cross-link Comparing experiments with IDs 2 & 5 in Table 2.4, we can see that augmenting the decoders of CoarseNet with cross-link helped improve the performance in all metrics, suggested that utilizing correlation is helpful for the

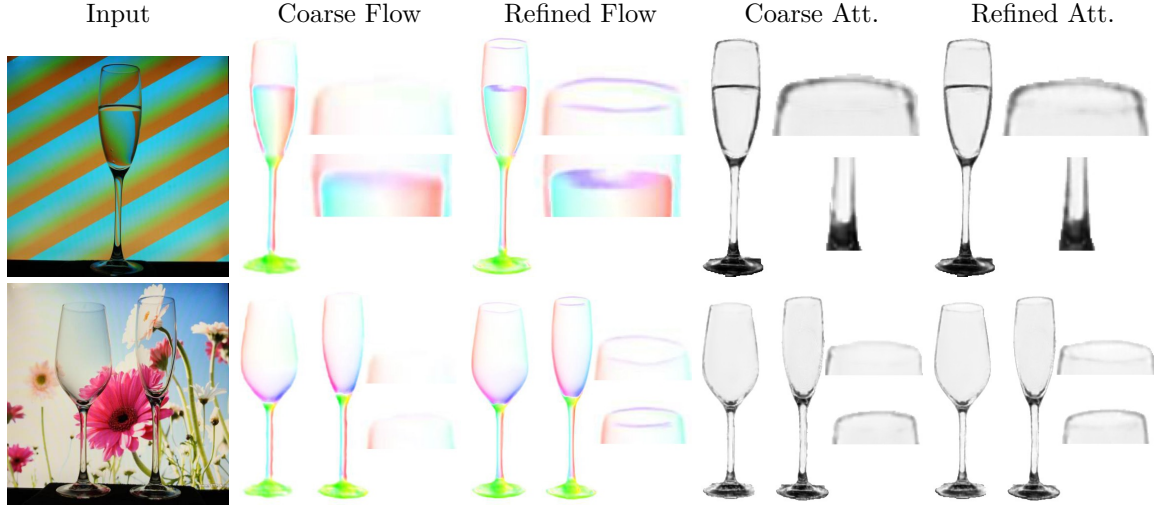


Fig. 2.6 Visualization of the effectiveness of the refinement stage on real data. After refinement, the refractive flow and attenuation mask have more clear structural details (*e.g.*, glass mouth).

matte estimation. Fig. 2.5 (b & e) qualitatively show the results without and with cross-link during training.

Effectiveness of multi-scale loss Comparing experiments with IDs 3 & 5 in Table 2.4, we can see that multi-scale loss boosted performance of CoarsNet in all of the evaluation metrics, particularly the attenuation mask MSE (see Fig. 2.5 (c & e) for qualitative comparison).

Effectiveness of image reconstruction loss Comparing experiments with IDs 4 & 5 in Table 2.4, we can see that adding image reconstruction loss in the coarse stage slightly improved the performance of refractive flow estimation and was very effective for reducing the image reconstruction error (see Fig. 2.5 (d & e) for qualitative comparison).

Effectiveness of RefineNet Comparing experiments with IDs 5 & 6 in Table 2.4, we can clearly see that RefineNet can significantly improve the refractive flow estimation. Fig. 2.5 (e & f) and Fig. 2.6 show that RefineNet can infer sharp details on both the synthetic and real data based on the outputs of CoarsNet, demonstrating

the effectiveness of the RefineNet. We also found that image reconstruction loss is not helpful for refractive flow estimation in the refinement stage (experiments with IDs 6 & 7 in Table 2.4). This is reasonable since the matte produced by CoarseNet already gives a small image reconstruction error, and further reducing the image reconstruction error does not guarantee a better refractive flow field.

2.6.2 Evaluation on Synthetic Data

Table 2.5 Quantitative results on the synthetic test dataset. (The first value for EPE is measured on the whole image and the second measured within the object region. A-MSE and I-MSE are computed on the whole image.)

	<i>Glass</i>				<i>Glass with Water</i>				<i>Lens</i>				<i>Complex</i>			
	F-EPE	A-MSE	I-MSE	M-IoU	F-EPE	A-MSE	I-MSE	M-IoU	F-EPE	A-MSE	I-MSE	M-IoU	F-EPE	A-MSE	I-MSE	M-IoU
Background	3.6 / 30.3	1.33	0.48	0.12	6.4 / 53.2	1.54	0.68	0.12	10.3 / 39.2	1.94	1.57	0.24	6.8 / 56.8	2.50	0.85	0.11
CoarseNet	2.1 / 15.8	0.22	0.14	0.97	3.1 / 23.5	0.31	0.23	0.97	2.0 / 6.7	0.17	0.28	0.99	4.5 / 34.4	0.38	0.33	0.92
TOM-Net	1.9 / 14.7	0.21	0.14	0.97	2.9 / 21.8	0.30	0.22	0.97	1.9 / 6.6	0.15	0.29	0.99	4.1 / 31.5	0.37	0.32	0.92

	Average				MSE ($\cdot 10^{-2}$)
	F-EPE	A-MSE	I-MSE	M-IoU	
Background	6.8 / 44.9	1.83	0.90	0.15	↓ better
CoarseNet	2.9 / 20.1	0.27	0.24	0.96	↑ better
TOM-Net	2.7 / 18.6	0.26	0.24	0.96	

Quantitative results for synthetic test dataset are presented in Table 2.5. We compared TOM-Net against *Background* and CoarseNet. Here, to accelerate training convergence, we first trained CoarseNet from scratch using our synthetic dataset excluding the complex shape subset. The trained CoarseNet was then fine-tuned using the entire training set including complex shapes, followed by training of RefineNet on the entire training set with random initialization. Similar to previous experiments, TOM-Net outperformed *Background* by a large margin, and slightly outperformed CoarseNet in both EPE and MSE, which implies more local details can be learned by RefinedNet.

The average IoU for object mask estimation is 0.96, indicates that TOM-Net can robustly segment the transparent object given only a single image as input. Although TOM-Net is not expected to learn highly accurate refractive flow, the average EPE errors $(2.7/18.6)^6$ are very small compared with the size of the input image (448×448) .

⁶The first value is measured on the whole image and the second measured within the object region.

In this sense, our predicted flow is capable of producing visually plausible refractive effect. The errors of complex shape category are larger than that of others, because complex shapes contain more sharp regions that will induce more errors. Figure 2.7 and Figure 2.8 show the qualitative results on five synthetic objects. The objects in the first four examples come from the test set where each example shows a specific object category. Although the background images and objects in the test set never appear in the training set, TOM-Net can still predict robust matte. The last row (*i.e.*, Fig. 2.8 (e)) shows a sample of *complex dog*, which was rendered using a 3D dog model. The pleasing result on the *complex dog* demonstrates that our model can generalize well from simple shapes to complex shapes.

Figure 2.7 shows that the reconstructed images using the estimated mattes (column 2) are very close to the input images (column 2), which empirically verifies that our simplified matting equation Eq. (2.6) is sufficiently accurate for this problem.

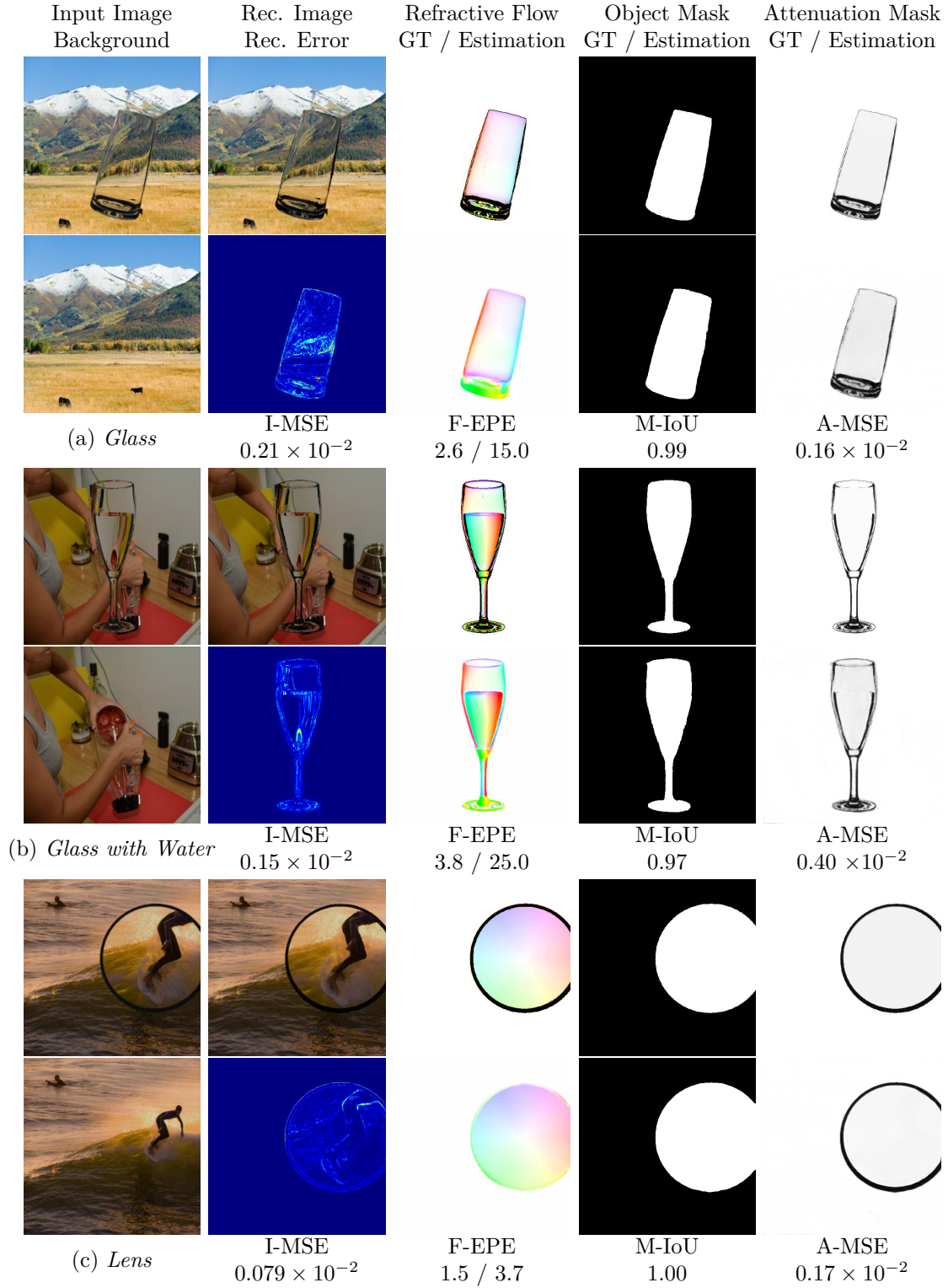


Fig. 2.7 Qualitative results on synthetic data (part 1). For each example, the first column shows the input image and background. The second column shows the reconstructed image and reconstruction error map. The last three columns show the ground truth matte and estimation. Quantitative results are shown below each example. Dark region in GT flow indicates no valid flow.

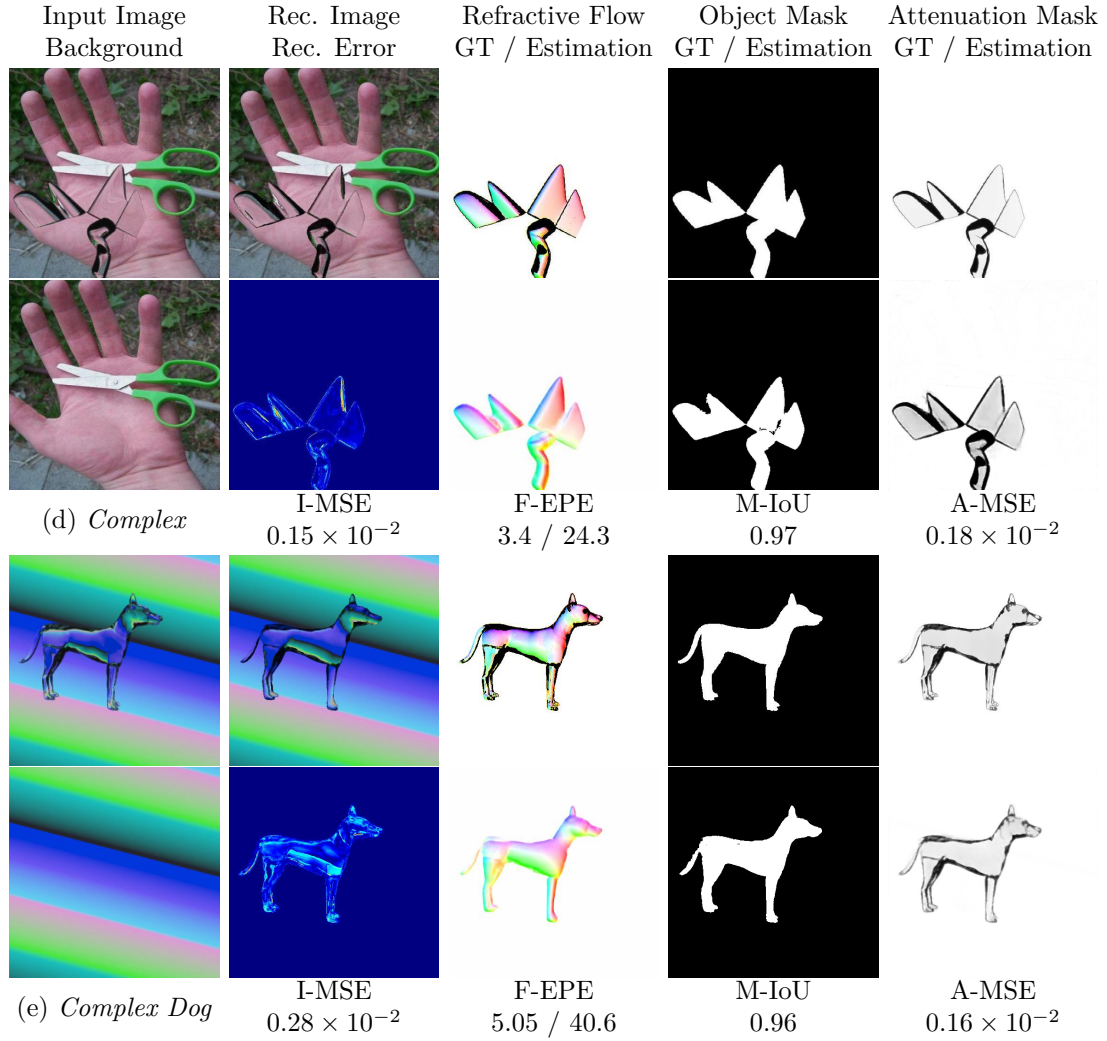


Fig. 2.8 Qualitative results on synthetic data (part 2). For each example, the first column shows the input image and background. The second column shows the reconstructed image and reconstruction error map. The last three columns show the ground truth matte and estimation. Quantitative results are shown below each example. Dark region in GT flow indicates no valid flow.

2.6.3 Evaluation on Real Data

Table 2.6 Quantitative results on real data. (Value the higher the better.)

	<i>Glass</i>		<i>Glass with Water</i>		<i>Lens</i>		<i>Complex</i>		Avg	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Background	22.05	0.894	20.75	0.886	18.60	0.860	16.85	0.816	19.56	0.864
CoarseNet	25.09	0.921	23.53	0.911	21.13	0.895	17.89	0.835	21.91	0.891
TOM-Net	25.06	0.920	23.53	0.911	20.89	0.893	17.88	0.835	21.84	0.890

We evaluated TOM-Net on our captured real dataset, which consists of 876 images of real objects. The results are shown in Table 2.6. The average PSNR and SSIM are above 21.0 and 0.89 respectively. The values are a bit lower for complex shapes, due to the opaque base of complex objects as well as the sharp regions of the objects that might induce large errors. After training, TOM-Net generalized well to common real transparent objects (see Fig. 2.9). It is worth to note that during training, each sample contains only one object, while TOM-Net can predict reliable matte for images containing multiple objects (see Fig. 2.9 (c)), which indicates the transferability and robustness of TOM-Net.

User study Remember that our goal is to estimate an environment matte that can produce visually realistic refractive effect from the input image, instead of estimating the highly accurate refractive flow. A user study was carried out to validate the realism of TOM-Net composites. 69 subjects participated in our user study. At the beginning, we showed each participant photographs of the transparent objects that will be seen during the user study. The objects consisted of 3 different glasses, 1 glass with water, 1 lens, and 1 complex shape. 40 samples, including 20 photographs⁷ and the corresponding 20 TOM-Net composites, were then randomly presented to each subject. When showing each sample, we also showed the corresponding background image to the subject for reference. We provided 3 options for each sample: (P) *photograph*, (C) *composite*, (N) *not distinguishable*. Table 2.7 shows the statistics of the user study. The 69 participants produced 1,380 votes for the 20 real photographs, and 1,380

⁷glass \times 12, glass & water \times 4, lens \times 2, and complex shape \times 2.

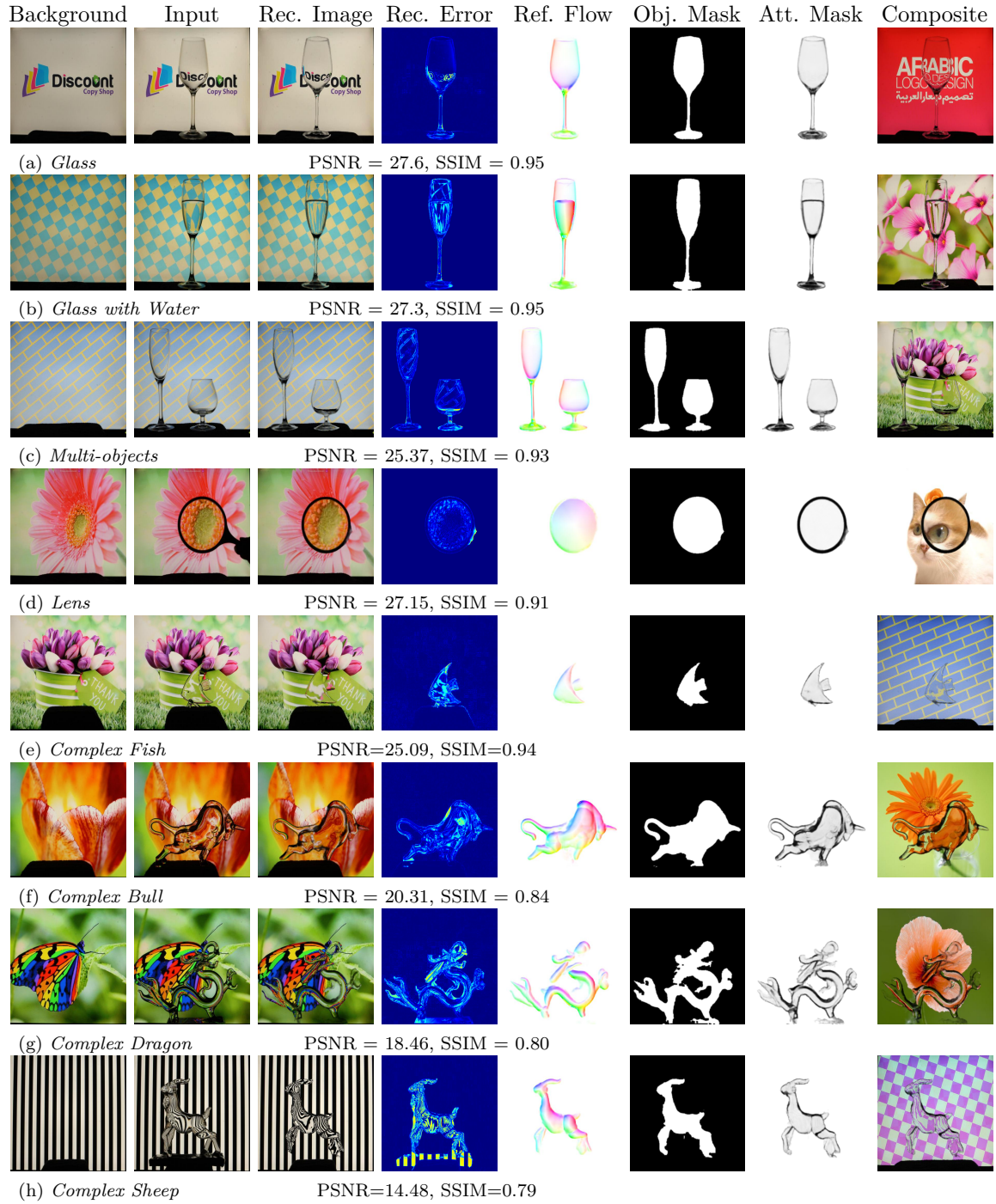


Fig. 2.9 Qualitative results on real data. The PSNR and SSIM between input photographs and reconstructed images are shown below each example. The last column shows the composites on novel backgrounds given the estimated matte.

Table 2.7 User study results. P, C, and N are short for votes for photograph, composite, and not distinguishable.

	<i>Glass</i>			<i>Glass with Water</i>			<i>Lens</i>			<i>Complex</i>			<i>All</i>		
	P	C	N	P	C	N	P	C	N	P	C	N	P	C	N
Photographs	522	275	31	163	97	16	74	48	16	91	35	12	850	455	75
Composites	531	266	31	145	113	18	73	52	13	78	51	9	827	482	71

votes for the 20 composites, respectively. The P:C:N ratios are 850 : 455 : 75 and 827 : 482 : 71 for photographs and composites respectively. The per-category ratios also follow a similar trend, indicating close chance of photographs and composites to be considered real, which further demonstrates TOM-Net can produce realistic matte.

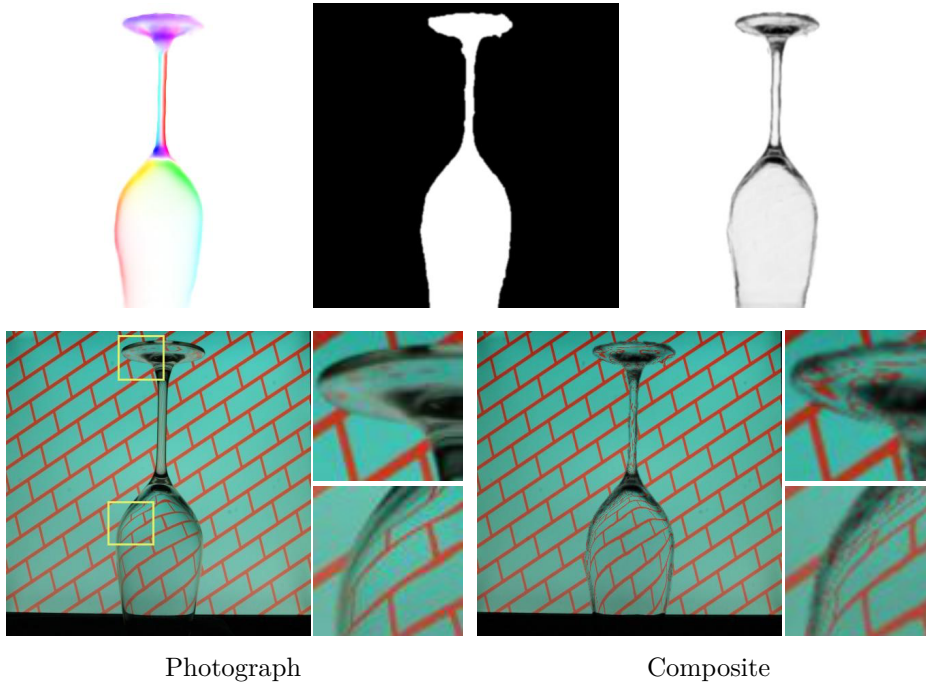


Fig. 2.10 Comparison of the photograph and composite. The first row shows the predicted matte, which is estimated by taking the photograph as input to our method. The second row compares the photograph and composite. When looking at the photograph and composite simultaneously, users can easily spot some imperfections of the composites (mostly in the boundary region).

Although we stress that TOM-Net can produce visually realistic composites, the results are still less than perfect. When looking at the real image and our composite side-by-side, users can spot some imperfections of the composite (mostly in the bound-

ary region, see Fig. 2.10). Therefore, we did not include such a user study by showing the real image and our composite side-by-side. Otherwise, the result will be biased. In the future, we will strengthen our approach to produce more realistic composites, so that the real image and our composite are indistinguishable even when showing them side-by-side.

2.6.4 Transparent Object Editing by Manipulating Environment Matte

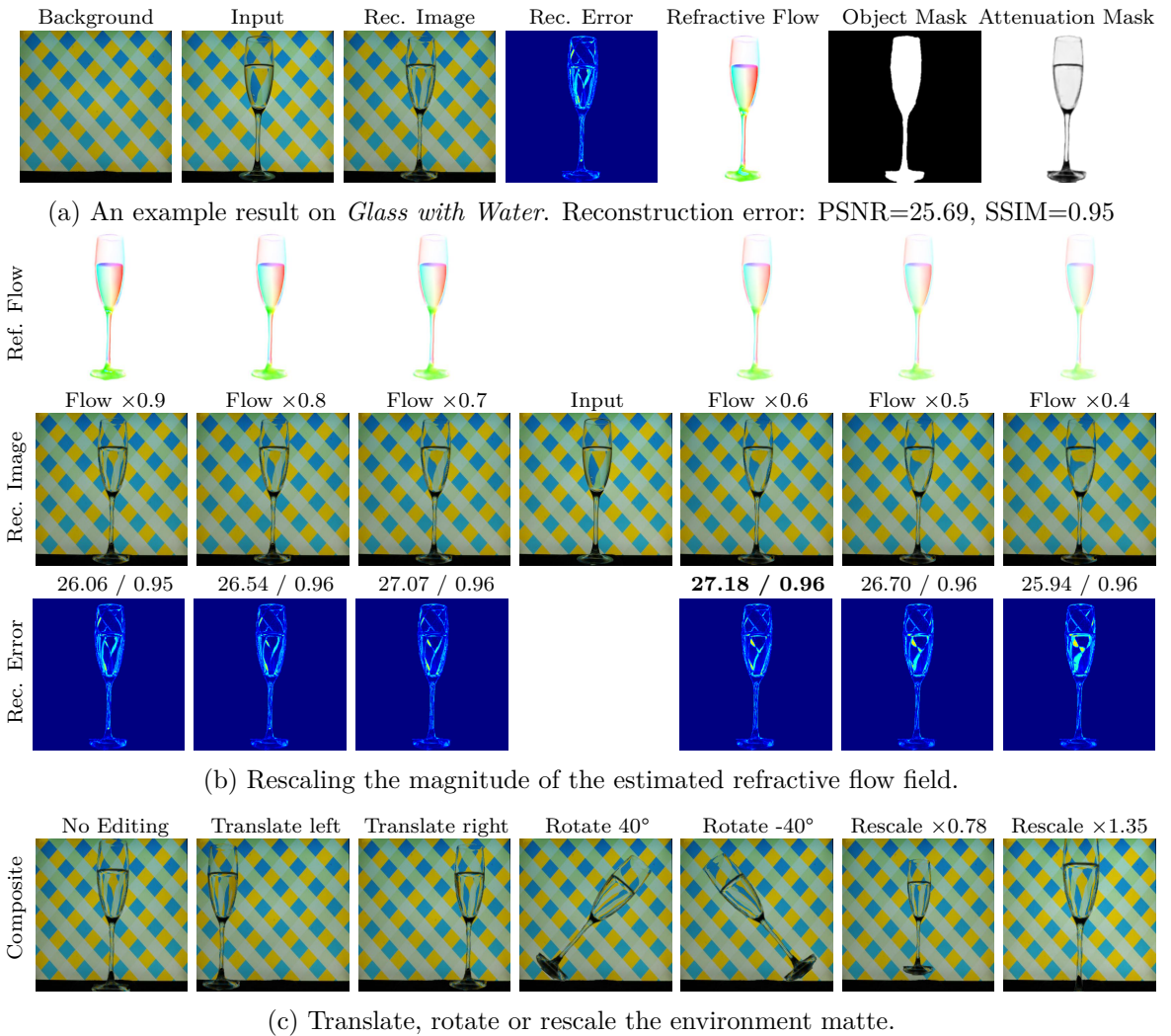


Fig. 2.11 Various novel composites of a *Glass with Water* shape obtained by manipulating the predicted environment matte.

Given a single image as input, our TOM-Net can estimate the environment matte as a triplet (consisting of an object mask, an attenuation mask, and a refractive flow field) in a fast feed-forward pass (see Fig. 2.11 (a) for an example). Note that the goal of the proposed TOM-Net is to extract an environment matte that can produce realistic refractive effect from a single image, instead of estimating a highly accurate environment matte. The reconstructed image in Fig. 2.11 (a) looks realistic but does not have the same refractive effect as the original input, as the refractive effect of the estimated matte seems stronger. By decreasing the magnitude of the estimated refractive flow field⁸, we can produce a similar refractive effect as the input image (see Fig. 2.11 (b)). When the scaling factor becomes 0.6, the reconstructed image achieved the lowest reconstruction error, with an improvement of 1.49 and 0.01 in PSNR and SSIM, respectively. Apart from rescaling the magnitude of the refractive flow field to adjust the refractive effect of the object, more interesting composites can be obtained by translating, rotating and rescaling the environment matte (see Fig. 2.11 (c)).

2.6.5 Failure Cases

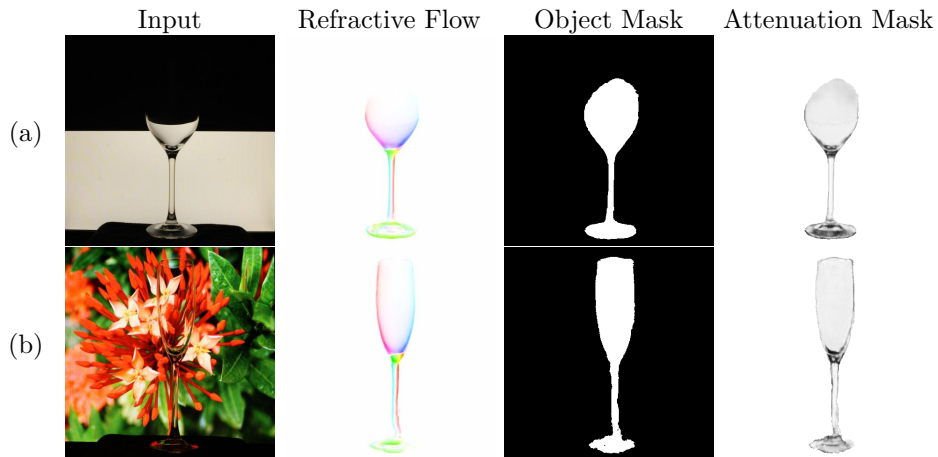


Fig. 2.12 Two failure cases on real data. In (a), our model fails to estimate the upper-part of the matte as there is no visual clue to find the object. In (b), the bottom part of the estimated matte is incomplete as the background image is heavily cluttered and the bottom part of the object is very dark.

⁸We simply multiply the refractive flow field by a scaling factor (< 1).

Our model can robustly estimate environment matte for different transparent objects in front of different backgrounds, however, when there is no visual clue for the objects or the image is too cluttered to separate the object from the background, our model may fail. Figure 2.12 shows two failure cases of our model on real data. In Fig. 2.12 (a), our model fails to extract the upper-part of the environment matte for the transparent glass due to the lack of visual clue. In Fig. 2.12 (b), although our model is still able to estimate a reasonable matte, the bottom part of the estimated matte is incomplete due to the very cluttered background.

2.6.6 Improvement with Trimap and Background Image

Table 2.8 Quantitative comparison between TOM-Net, TOM-Net^{+Trimap} and TOM-Net^{+Bg} on the synthetic test dataset.

	<i>Glass</i>				<i>Glass with Water</i>				<i>Lens</i>				<i>Complex</i>			
	F-EPE	A-MSE	I-MSE	M-IoU	F-EPE	A-MSE	I-MSE	M-IoU	F-EPE	A-MSE	I-MSE	M-IoU	F-EPE	A-MSE	I-MSE	M-IoU
Background	3.6 / 30.3	1.33	0.48	0.12	6.4 / 53.2	1.54	0.68	0.12	10.3 / 39.2	1.94	1.57	0.24	6.8 / 56.8	2.50	0.85	0.11
TOM-Net	1.9 / 14.7	0.21	0.14	0.97	2.9 / 21.8	0.30	0.22	0.97	1.9 / 6.6	0.15	0.29	0.99	4.1 / 31.5	0.37	0.32	0.92
TOM-Net ^{+Trimap}	1.8 / 14.4	0.21	0.14	0.98	2.6 / 20.7	0.29	0.20	0.98	1.7 / 6.1	0.15	0.27	1.00	3.7 / 29.4	0.37	0.29	0.95
TOM-Net ^{+Bg}	1.6 / 13.1	0.21	0.12	0.99	2.4 / 19.3	0.29	0.19	0.98	1.4 / 4.9	0.18	0.19	1.00	3.5 / 27.7	0.36	0.27	0.97

Average					MSE ($\cdot 10^{-2}$)
	F-EPE	A-MSE	I-MSE	M-IoU	
Background	6.8 / 44.9	1.83	0.90	0.15	
TOM-Net	2.7 / 18.6	0.26	0.24	0.96	↓ better
TOM-Net ^{+Trimap}	2.5 / 17.7	0.26	0.23	0.98	↑ better
TOM-Net ^{+Bg}	2.2 / 16.2	0.26	0.19	0.98	

Table 2.9 Quantitative comparison between TOM-Net, TOM-Net^{+Trimap} and TOM-Net^{+Bg} on real data.

	<i>Glass</i>		<i>Glass with Water</i>		<i>Lens</i>		<i>Complex</i>		<i>Average</i>	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Background	22.05	0.894	20.75	0.886	18.60	0.860	16.85	0.816	19.56	0.864
TOM-Net	25.06	0.920	23.53	0.911	20.89	0.893	17.88	0.835	21.84	0.890
TOM-Net ^{+Trimap}	25.48	0.924	23.77	0.914	23.98	0.913	20.88	0.868	23.53	0.905
TOM-Net ^{+Bg}	26.10	0.931	24.58	0.922	25.52	0.924	22.23	0.884	24.61	0.915

At test time, the input trimaps for TOM-Net^{+Trimap} were generated in the same way adopted in the training (as described in Section 2.4.4), except that the foreground regions were obtained by performing erosion operation on the ground-truth object mask with a fixed (rather than a random) kernel size of 10 pixels for evaluation. Table 2.8

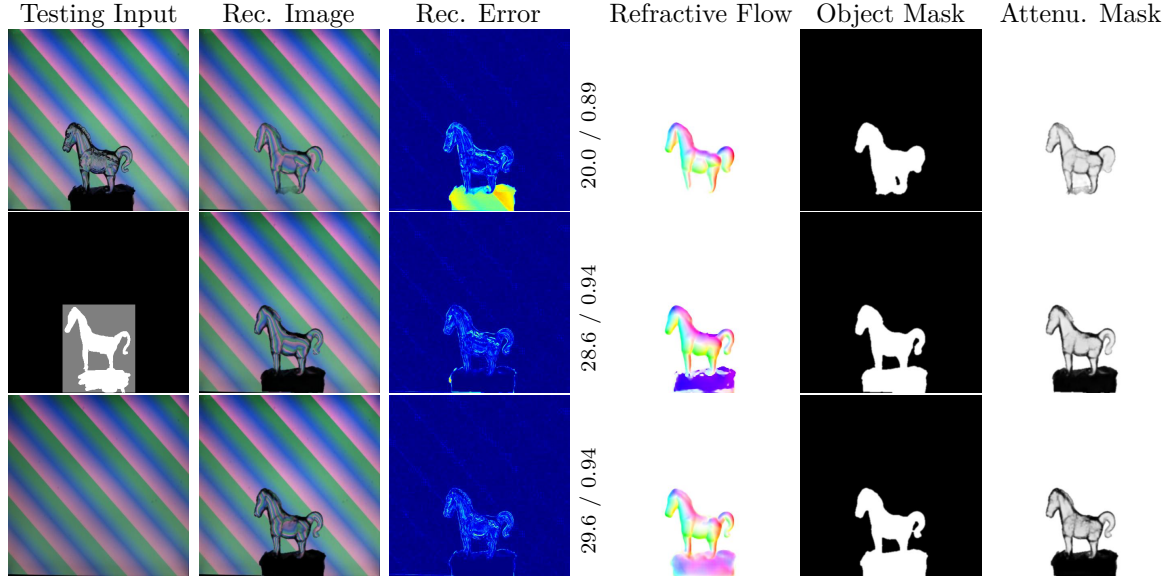
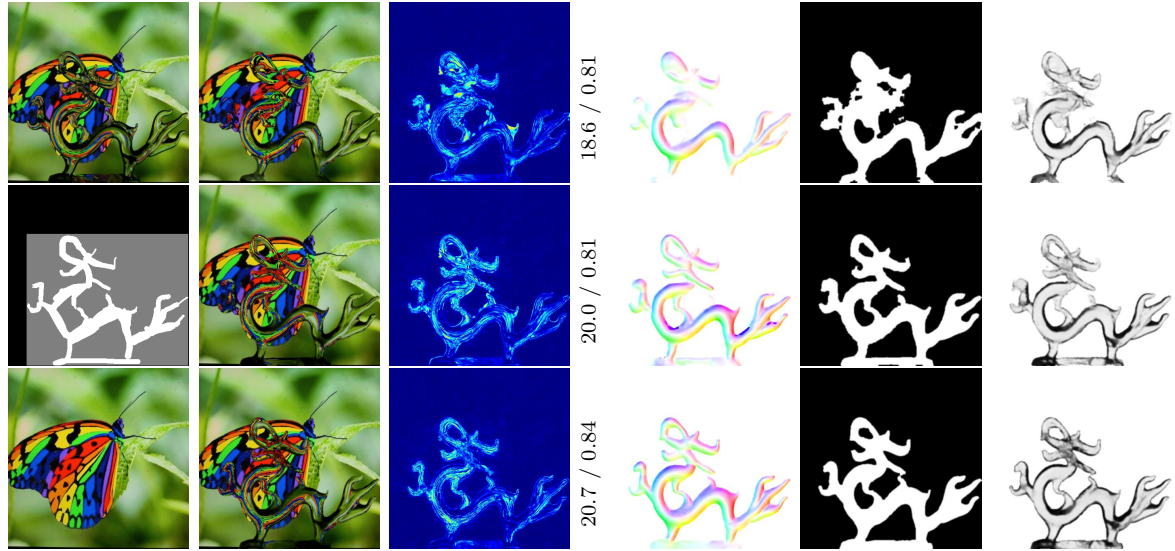
(a) *Complex Horse*(b) *Complex Dragon*

Fig. 2.13 Qualitative comparison between TOM-Net, TOM-Net^{+Trimap} and TOM-Net^{+Bg} on real data. For each testing object, the input to the model is shown on the first column, and the results of TOM-Net (up), TOM-Net^{+Trimap} (middle) and TOM-Net^{+Bg} (bottom) are shown on the rest of the columns. Note that for TOM-Net^{+Trimap} and TOM-Net^{+Bg}, we do not show the input image for simplicity. The PSNR and SSIM between input photographs and reconstructed images are shown right after the error maps.

shows the quantitative comparisons between TOM-Net, TOM-Net^{+Trimap} and TOM-Net^{+Bg} on the synthetic test dataset. As expected, with the access to the additional information, both TOM-Net^{+Trimap} and TOM-Net^{+Bg} performed better than TOM-Net. Due to the fact that a background image contains more useful information than a trimap, TOM-Net^{+Bg} achieved the best results.

Table 2.9 presents the quantitative comparison on real data. Compared with TOM-Net, TOM-Net^{+Trimap} and TOM-Net^{+Bg} achieved an improvement of 1.69 and 2.77 in average PSNR and an improvement of 0.015 and 0.024 in average SSIM, respectively. Fig. 2.13 shows the qualitative comparison on real data, where the foreground region of the trimap was marked by the user. It can be seen that with the additional information, TOM-Net^{+Trimap} and TOM-Net^{+Bg} can identify the transparent object from the cluttered background more accurately than TOM-Net and model the opaque base of the transparent object (Fig. 2.13 (a)). As a result, the environment matte predicted by TOM-Net^{+Trimap} and TOM-Net^{+Bg} can produce more realistic composites and achieve lower reconstruction errors, clearly demonstrating the effectiveness of our framework in handling cases where a trimap or a background image is available.

2.7 Discussion

2.7.1 Limitations

Although our method can produce plausible results for transparent object matting, there do exist limitations that require further study. First, our model assumes objects to be colorless so that the attenuation property of an object can be depicted as a scalar value ρ in our formulation. However, this is not applicable to colored transparent objects, as shown in see Fig. 2.14 (a). Although our method can estimate a reasonably good object mask and refractive flow field for the *Glass with Water*, the estimated attenuation mask cannot model the colored effect of the object.

Second, our model assumes a single planar background (following most of the previous works) as the only light source and simplifies the interaction between object and

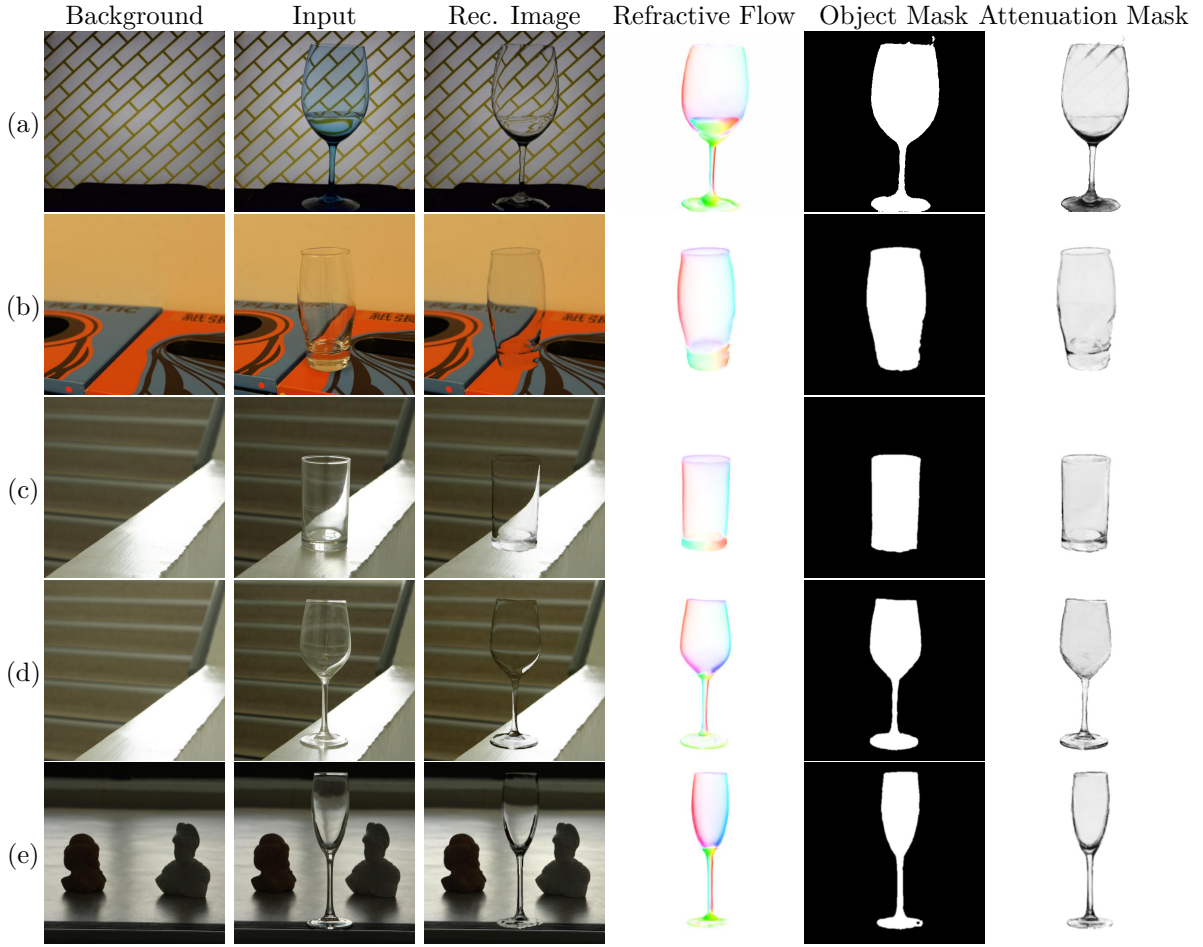


Fig. 2.14 Qualitative results of TOM-Net on colored transparent object (first row) and objects under natural illuminations (last four rows).

background image to a point-to-point (single) mapping. However, more complicated effects exist in the real world, such as specular highlights, translucency, multi-mapping (*i.e.*, refraction and reflection happen simultaneously at a surface point), and color dispersion (*i.e.*, different color components may have different supporting background regions). Fig. 2.14 (b)-(e) show four example results of TOM-Net on transparent objects under different types of natural illuminations. Regardless of the fact that TOM-Net can estimate a plausible object mask and refractive flow field, the composites do not look very realistic. This is because our current formulation does not consider the more sophisticated refractive properties of a transparent object under natural illumination like complex interaction with environment light, specular highlight, Fresnel effect, and

acoustic shadow.

2.7.2 Colored Objects and Specular Highlights

Here we sketch the potential solutions to colored transparent objects as well as the cases when specular highlights appear on transparent objects. In Section 2.3, we simplified matting equation as Eq. (2.6). To handle colored objects, the scalar attenuation index ρ should be expanded to a color attenuation 3-vector R , in which each value corresponds to an attenuation index for a specific color channel. The matting equation then becomes

$$C = (1 - m)B + mR \circ \mathcal{M}(\mathbf{T}, P), \quad (2.13)$$

where \circ represents element-wise multiplication.

Consider a white near point light source, we can simplify the specular highlight effect with a specular highlight component S , then the generalized matting equation can be written as

$$C = (1 - m)B + mR \circ \mathcal{M}(\mathbf{T}, P) + S, \quad (2.14)$$

where S is a 3-vector containing three identical values. The problem of transparent object matting now becomes simultaneously estimating an object mask, a color attenuation mask, a refractive flow field, and a specular highlight mask from a single image, while more efforts are needed to implement them for practical use and we leave this as our future work.

2.7.3 Difficulty in Comparison with Previous Works

Currently, it is not trivial to have a fair comparison with existing methods. On one hand, applying our method on the data used in the previous methods is difficult. Most of the previous methods require multiple images of the transparent object captured in front of pre-designed patterns, which are not publicly available and lack enough textures for our method to estimate the refractive effect of the transparent object. The single image based methods RTCEM [22] and [30] have additional requirements.

In particular, RTCEM [22] requires the object to be captured in front of a coded-pattern (also not publicly available), and the background image is needed to segment the foreground object. [30] requires human interaction to segment the foreground object and model the object’s refractive effect with thin-plate-spline transformation. The data used in [30] does not follow our assumption that the light comes from a single background image, thus it cannot be directly processed by our method. On the other hand, there are no public implementations for the previous methods, and even if there were, those methods cannot be applied to our dataset which is created for single image transparent object matting.

Different from the previous methods, our method aims to estimate the foreground mask, attenuation mask, and refractive flow field from a single natural image. Since our code and datasets have been made publicly available, it will ease the comparison for the future work. We believe our work can serve as a baseline and provide meaningful insight for future researches in this area.

2.7.4 Generalization to Real Data

As it is very difficult and time consuming to create a large scale real dataset for training, we use synthetic data for training and evaluate its performance on real data. It is well-known that there is a domain gap between the synthetic and real data, and a model trained on synthetic data may not generalize well to real data. We hypothesize that the reasons why our method works well on real data are as follows. Following previous works, the real transparent objects are captured in front of a monitor. Under our assumption, the captured images and the rendered images are very similar. Moreover, extensive data augmentation is performed to close the gap between the synthetic and real data.

To further improve the generalization ability of our method on real data, we will explore the idea of exploiting real data for self-supervised training or fine-tuning in the future.

2.7.5 Design of the Network Architecture

To better recover the details of the refractive flow field and attenuation mask, we propose a two-stage network architecture for this problem. Our results show that RefineNet can effectively improve the results of CoarseNet. However, our two-stage network requires stepwise training (*i.e.*, we first trained CoarseNet until convergence and then trained RefineNet) which requires longer training time. An interesting future direction is to develop a more efficient single-stage network that achieves comparable performance as the two-stage network.

2.8 Conclusion

We have introduced a simple and efficient model for transparent object matting, and proposed a CNN architecture, called TOM-Net, that takes a single image as input and predicts environment matte as an object mask, an attenuation mask, and a refractive flow field in a fast feed-forward pass. We created a large-scale synthetic dataset and a real dataset as a benchmark for learning transparent object matting. We have also shown that TOM-Net can perform better by incorporating a trimap or a background image in the input. Promising results have been achieved on both synthetic and real data, which clearly demonstrate the feasibility and effectiveness of the proposed approach.

Chapter 3

Learning Photometric Stereo

3.1 Introduction

Given multiple images of a static object captured under different light directions with a fixed camera, the surface normals of the object can be estimated using photometric stereo techniques. Early calibrated photometric stereo methods assumed a simplified reflectance model, such as the ideal Lambertian model [9, 14] or analytical reflectance models [49–51]. However, most of the real-world objects are non-Lambertian, and a specific analytical model is only valid for a small set of materials. A bidirectional reflectance distribution function (BRDF) is a general form for describing the reflectance property of a surface, but it is difficult to directly use a non-parametric form of BRDFs for photometric stereo. Hence, it remains an open and challenging problem to develop a computationally efficient photometric stereo method that can handle materials with diverse BRDFs.

Recently, with the great success of deep learning in various computer vision tasks, deep learning based methods have been introduced to calibrated photometric stereo to handle surfaces with general and unknown isotropic reflectance [52–54]. Instead of explicitly modeling complex surface reflectances, they directly learn the mapping from reflectance observations to surface normals given known light directions. However, the method in [52] depends on a pre-defined set of light directions during training

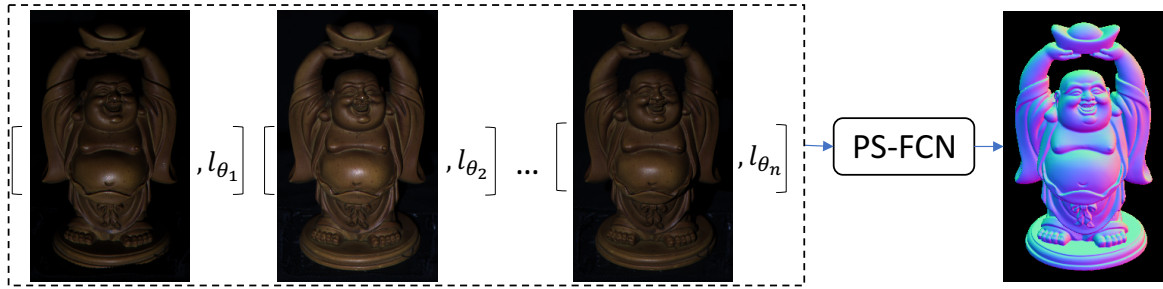


Fig. 3.1 Given an arbitrary number of images and their associated light directions as input, our model estimates a normal map of the object in a fast feed-forward pass.

and testing. The methods in [52, 53] estimate the surface normals in a pixel-wise manner, making them not possible to account for the local context information of a surface point (*e.g.*, surface smoothness prior). Tani and Maehara [54] introduced an optimization framework based on deep neural network, but their method suffers from complex scenes and requires a long processing time.

In this work, we propose a deep fully convolutional network (FCN) [55], called PS-FCN for calibrated photometric stereo. PS-FCN takes an arbitrary number of images with their associated light directions as input, and predicts a surface normal map of the scene in a fast feed-forward pass (see Fig. 3.1). Compared with previous learning based methods, our method does not depend on a pre-defined set of light directions during training and testing, and can handle multiple images in an order-agnostic manner. Moreover, convolutional neural network (CNN) can naturally incorporate information of the observations at neighboring pixels for computing feature maps, allowing our method to take advantage of local context information.

To simulate real-world complex non-Lambertian surfaces for training PS-FCN, we create two synthetic datasets using shapes from the blobby shape dataset [56] and the sculpture shape dataset [57], and BRDFs from the MERL BRDF dataset [58]. After training on synthetic data, we show that PS-FCN can generalize well on real datasets, including the DiLiGenT benchmark [59], the Gourd&Apple dataset [60], and the Light Stage Data Gallery [61]. Extensive experiments on public real datasets show that PS-FCN outperforms existing approaches in calibrated photometric stereo.

Preliminary results of this chapter were published in [17, 18]. Our code, models, and datasets are available at <https://guanyingc.github.io/PS-FCN>.

3.2 Related Work

In this section, we briefly review representative non-Lambertian photometric stereo techniques. More comprehensive surveys of photometric stereo algorithms can be found in [59, 62]. Non-Lambertian photometric stereo methods can be broadly divided into four categories, namely outlier rejection based methods, sophisticated reflectance model based methods, exemplar based methods, and learning based methods.

Outlier rejection based methods assume non-Lambertian observations to be local and sparse such that they can be treated as outliers. Various outlier rejection methods have been proposed so far. They are based on rank minimization [63], RANSAC [64], taking median values [65], expectation maximization [66], and sparse Bayesian regression [67]. These outlier rejection methods generally require lots of input images and have difficulty in handling objects with non-sparse non-Lambertian observations (*e.g.*, materials with broad and soft specular highlights).

Many sophisticated reflectance models have been proposed to approximate the non-Lambertian model, including analytical models like Torrance-Sparrow model [68], Ward model [50], Cook-Torrance model [51], etc. Instead of rejecting specular observations as outliers, sophisticated reflectance model based methods fit an analytical model to all observations. These methods require solving complex optimization problems, and can only handle limited classes of materials. Recently, bivariate BRDF representations [69, 70] were adopted to approximate isotropic BRDF, and a symmetry-based approach [71] was proposed to handle anisotropic reflectance without explicitly estimating a reflectance model.

Exemplar based methods usually require the observation of an additional reference object. Using a reference sphere, Hertzmann and Seitz [72] subtly transformed the non-Lambertian photometric stereo problem to a point matching problem. Exemplar

based methods can deal with objects with spatially-varying BRDFs without knowing the light directions, but the requirement of known shape and material of the reference object(s) limits their applications. As an extension, Hui and Sankaranarayanan [73] introduced a BRDF dictionary to render virtual spheres without using a real reference object, but at the cost of requiring light calibration and longer processing time.

More recently, a few deep learning based methods have been introduced to calibrated photometric stereo [52–54]. Santo *et al.* [52] proposed a fully-connected network to learn the mapping from reflectance observations captured under a pre-defined set of light directions to surface normal in a pixel-wise manner. Ikehata [53] introduced a fixed shape representation, called observation map, that is invariant to the number and permutation of the images. For each surface point of the object, all its observations are merged into an observation map based on the given light directions, and the observation map is then fed to a CNN to regress a normal vector. Li *et al.* [74] and Zheng *et al.* [75] focused on reducing the number of required images while maintaining similar accuracy under the framework proposed by Ikehata [53]. Compared with [52, 53], our method can take advantage of local context information in predicting the surface normals, which results in a more robust behavior. Taniai and Maehara [54] introduced an unsupervised learning framework that predicts both the surface normals and reflectance images of an object. Their model is “trained” at test time for each test object by minimizing the reconstruction loss between the input images and the rendered images, while our model is trained with supervised learning and achieves better performance on complex surfaces.

3.3 Image Formulation Model

Following the conventional practice, we assume an orthographic camera with a linear radiometric response, directional lightings coming from the upper-hemisphere, and the viewing direction is parallel to the z -axis pointing towards the origin of world coordinates. Let us further assume that the image coordinates is aligned with the world

x - y coordinates. Consider a non-Lambertian surface whose appearance is described by a general isotropic BRDF ρ . Given a surface point with a unit surface normal vector $\mathbf{n} \in \mathcal{S}^2$, $\mathcal{S}^2 = \{\mathbf{v} \in \mathbb{R}^3 : \|\mathbf{v}\|_2 = 1\}$ illuminated by the j -th incoming light with direction $\mathbf{l}_j \in \mathcal{S}^2$ and intensity $e_j \in \mathbb{R}_+$, the image formation model from a fixed viewpoint can be expressed as

$$m_j = e_j \rho(\mathbf{n}, \mathbf{l}_j) \max(\mathbf{n}^\top \mathbf{l}_j, 0) + \epsilon_j, \quad (3.1)$$

where m represents the measured intensity, $\max(\cdot, 0)$ operator expresses for attached shadows, and ϵ accounts for the global illumination effects (*e.g.*, cast shadows and inter-reflections) and noise.

For a Lambertian surface, the BRDF ρ becomes an unknown constant. Theoretically, with three observations captured under non-coplanar light directions (without shadows), the albedo scaled surface normal can be uniquely determined [9]. However, pure Lambertian surfaces barely exist, and we therefore have to consider a more complex problem of non-Lambertian photometric stereo.

Based on this model, given the observations of surface points (corresponding to individual pixels) under different (known) incoming lights, our method estimates the surface normals for these surface points. Different from traditional methods which approximate ρ with some sophisticated reflectance models, our method directly learns the mapping from images and lightings to surface normals without explicitly modeling ρ .

3.4 A Flexible Learning Framework for Photometric Stereo

In this section, we first introduce our strategy for adapting CNNs to handle a variable number of inputs, and then present a flexible fully convolutional network, called PS-FCN, for learning photometric stereo.

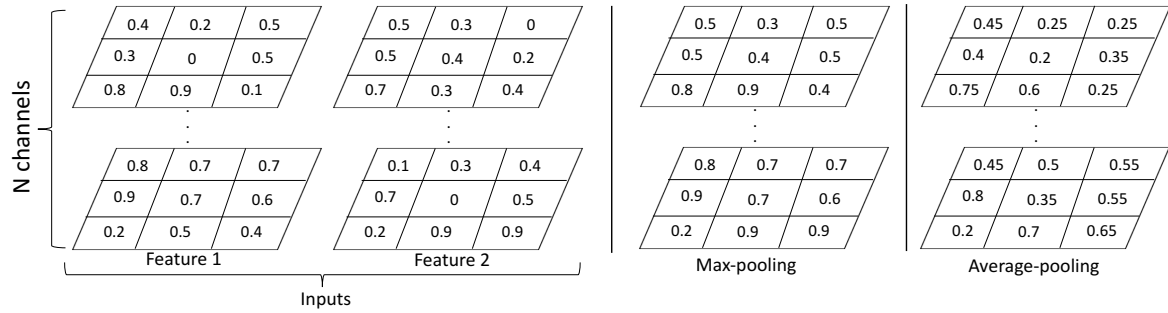


Fig. 3.2 A toy example for max-pooling and average-pooling mechanisms on multi-feature fusion.

3.4.1 Max-pooling for Multi-feature Fusion

CNNs have been successfully applied to dense regression problems like depth estimation [39] and surface normal estimation [76], where the number of input images is fixed and identical during training and testing. Note that adapting CNNs to handle a variable number of inputs during testing is not straightforward, as convolutional layers require the input to have a fixed number of channels during training and testing. Given a variable number of inputs, a shared-weight feature extractor can be used to extract features from each of the inputs (*e.g.*, siamese networks), but an additional fusion layer is required to aggregate such features into a representation with a fixed number of channels. A convolutional layer is applicable for multi-feature fusion only when the number of inputs is fixed. Unfortunately, this is not practical for photometric stereo where the number of inputs often varies.

One possible way to tackle a variable number of inputs is to arrange the inputs sequentially and adopt a recurrent neural network (RNN) to fuse them. For example, [77] introduced a RNN framework to unify single- and multi-image 3D voxel prediction. The memory mechanism of RNN enables it to handle sequential inputs, but at the same time also makes it sensitive to the order of inputs. This order sensitive characteristic is not desirable for photometric stereo as it will restrict the illumination changes to follow a specific pattern, making the model less general.

More recently, order-agnostic operations (*e.g.*, pooling layers) have been exploited in CNNs to aggregate multi-image information. Wiles and Zisserman [57] used max-

pooling to fuse features of silhouettes from different views for novel view synthesis and 3D voxel prediction. Hartmann *et al.* [78] adopted average-pooling to aggregate features of multiple patches for learning multi-patch similarity. In general, max-pooling operation can extract the most salient information from all the features, while average-pooling can smooth out the salient and non-activated features. Fig. 3.2 illustrates how max-pooling and average-pooling operations aggregate two features with a toy example.

For photometric stereo, we argue that max-pooling is a better choice for aggregating features from multiple inputs. Our motivation is that, under a certain light direction, regions with high intensities or specular highlights provide strong clues for surface normal inference (*e.g.*, for a surface point with a sharp specular highlight, its normal is close to the bisector of the viewing and light directions). Max-pooling can naturally aggregate such strong features from images captured under different light directions. Besides, max-pooling can ignore non-activated features during training, making it robust to cast shadow. As will be seen in Section 3.6, our experimental results do validate our arguments. We observe from experiments that each channel of the feature map fused by max-pooling is highly correlated to the response of the surface to a certain light direction. Strong responses in each channel are found in regions with surface normals having similar directions. The feature map can therefore be interpreted as a decomposition of the images under different light directions (see Fig. 3.9).

3.4.2 Network Architecture

PS-FCN is a multi-input-single-output (MISO) network consisting of three components, namely a shared-weight *feature extractor*, a *fusion layer*, and a *normal regression sub-network* (see Fig. 3.3). It can be trained and tested using an arbitrary number of images with their associated light directions as input¹.

¹For calibrated photometric stereo, the input images are normalized by light intensities, and each light direction is represented by a unit 3-vector.

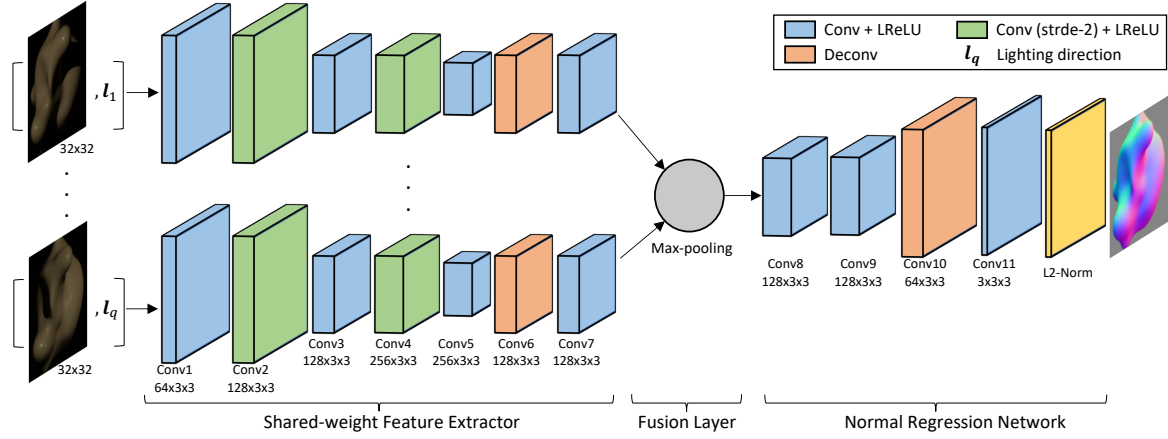


Fig. 3.3 Network architecture of PS-FCN.

For each light direction, we have a 3-channel input image with the dimensions of $3 \times h \times w$, where h and w are the image height and width, respectively. Concatenating images taken under q different lightings $\{l_1, \dots, l_q\}$, we have the data with the dimensions of $q \times 3 \times h \times w$. In addition, we represent the light vectors $\{l_1, \dots, l_q\}$ as 3-channel images having the same spatial resolution as the input images, resulting in another $q \times 3 \times h \times w$ data. Putting them together, we finally have $q \times 6 \times h \times w$ dimensional inputs to our model. We separately feed the image-light pairs to the shared-weight feature extractor to extract a feature map from each of the inputs, and apply a max-pooling operation in the fusion layer to aggregate these feature maps. Finally, the normal regression sub-network takes the fused feature map as input and estimates a normal map of the object.

The shared-weight feature extractor has seven convolutional layers, where the feature map is down-sampled twice and then up-sampled once, resulting in a down-sample factor of two. This design can increase the receptive field and preserve spatial information with a small memory consumption. The normal regression sub-network has four convolutional layers and up-samples the fused feature map to the same spatial dimension as the input images. A L2-normalization layer is appended at the end of the normal regression sub-network to produce the normal map.

As PS-FCN is a fully convolutional network, it can be applied to datasets with

different image sizes. Thanks to the max-pooling operation in the fusion layer, PS-FCN possesses an order-agnostic property. Besides, PS-FCN can be easily extended to handle uncalibrated photometric stereo, where the light directions are not known, by simply removing the light directions during training.

Loss function Training of our PS-FCN is supervised by the estimation error between the predicted and the ground-truth normal maps. We formulate our loss function as the commonly used cosine similarity loss, given by

$$\mathcal{L}_{\text{Normal}} = \frac{1}{hw} \sum_i^{hw} (1 - \mathbf{n}_i^\top \tilde{\mathbf{n}}_i), \quad (3.2)$$

where \mathbf{n}_i and $\tilde{\mathbf{n}}_i$ denote the predicted normal and the ground-truth normal, respectively, at pixel i . If the predicted normal has a similar orientation as the ground truth, the dot-product $\mathbf{n}_i \cdot \tilde{\mathbf{n}}_i$ will be close to 1 and the loss becomes small, and vice versa. Other losses like mean squared error can also be alternatively adopted.

3.4.3 Data Normalization for Handling Surfaces with SVBRDFs

As PS-FCN is a fully-convolutional network that processes the input images in a patch-wise manner and is trained on surfaces with homogeneous BRDF, it may have difficulties in dealing with steep color changes caused by surfaces with spatially-varying BRDFs (SVBRDFs), as shown in Fig. 3.4 (c). A straightforward idea to tackle this problem is to train a model on surfaces with SVBRDFs. However, creating a large-scale training dataset for this purpose is not trivial, since modeling surface appearance with realistic SVBRDFs requires manual editing from artists. Even someone can collect a large-scale dataset of objects with SVBRDFs, the created dataset may not be able to faithfully cover the distribution of real data. In this work, we introduce a simple yet effective data normalization strategy to enable PS-FCN to handle surfaces with SVBRDFs robustly. We will show that with the proposed data normalization strategy, our method can generalize well to surfaces with SVBRDFs after training

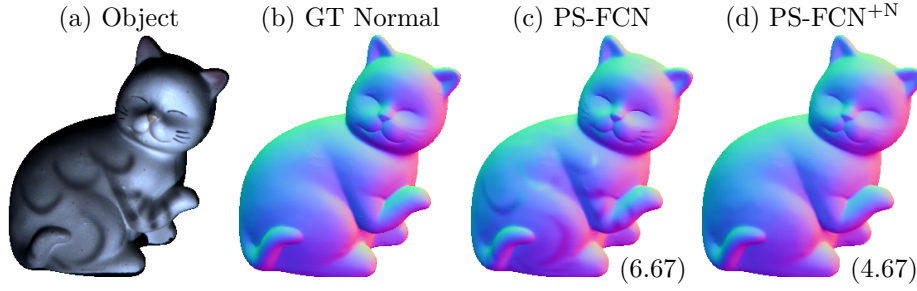


Fig. 3.4 Comparison between PS-FCN and PS-FCN^{+N} on CAT with spatially-varying BRDFs. Numbers in parentheses denote mean angular error (MAE) in degree.

only on surfaces with homogeneous BRDF.

During training, given q observations of a surface point², we concatenate all the observations and normalize them to a unit length vector by

$$(m'_1, \dots, m'_q) = \left(\frac{m_1}{\sqrt{m_1^2 + \dots + m_q^2}}, \dots, \frac{m_q}{\sqrt{m_1^2 + \dots + m_q^2}} \right), \quad (3.3)$$

where m and m' represent the original and normalized observations, respectively (for RGB images, we perform normalization on each channel separately). The intuition behind this operation is as follows. Consider a Lambertian model, the BRDF $\rho(\mathbf{n}, \mathbf{l})$ degenerates to a constant albedo ρ and $m = \rho \max(\mathbf{n}^\top \mathbf{l}_j, 0)$. After the data normalization operation, we have

$$m'_i = \frac{\max(\mathbf{n}^\top \mathbf{l}_i, 0)}{\sqrt{\max(\mathbf{n}^\top \mathbf{l}_1, 0)^2 + \dots + \max(\mathbf{n}^\top \mathbf{l}_q, 0)^2}}. \quad (3.4)$$

Equation (3.4) shows that the effect of albedo in Lambertian surfaces can be removed after performing data normalization, as shown in the first example in Fig. 3.5.

However, the above conclusion is not true for non-Lambertian surfaces, because for regions with specular highlights under some light directions, the observations under other light directions will be suppressed after data normalization (see the example of BALL in Fig. 3.5). Nevertheless, we experimentally found that such a normalization strategy works equally well for non-Lambertian surfaces under the PS-FCN

²Note that the observations are already normalized by the light intensities.

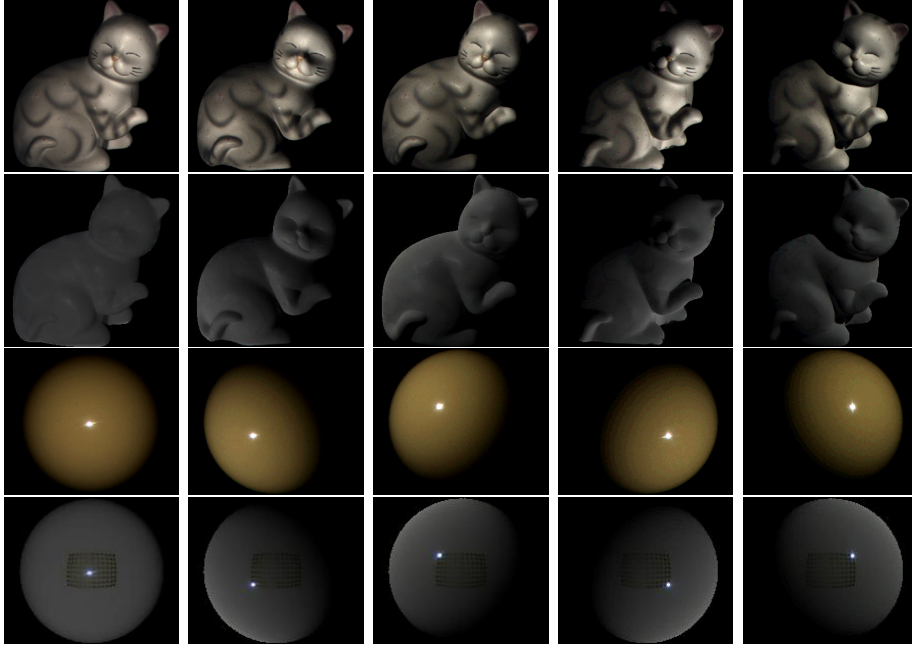


Fig. 3.5 Illustration of the introduced data normalization operation on CAT and BALL in the DiLiGenT benchmark. The first and third rows show the original images, while the second and last rows show the normalized images. Only 5 out of 96 images for each object are shown.

framework. This might be explained by the fact that for a non-Lambertian surface under directional lighting, the low-frequency observations are quite close to Lambertian reflectance [69]. For observations exhibiting specular highlights under some light directions, the max-pooling operation in the fusion layer can naturally ignore the non-activated features (*i.e.*, features extracted from the suppressed observations) and aggregate the most salient features. Note that this normalization strategy has also been adopted in [79, 80] to compute the similarity between two pixel intensity profiles of non-Lambertian surfaces, while we use this normalization strategy as a preprocessing for CNNs to handle surfaces with SVBRDFs.

When the number of input images at test time t is different from that in training q , the magnitude of the normalized observations will be different, which leads to decreased performance (*e.g.*, when all observations have the same values, we have $m'_{\text{train}} = 1/\sqrt{q}$, $m'_{\text{test}} = 1/\sqrt{t}$). We experimentally verified that multiplying the normalized observations with the scalar $\sqrt{t/q}$ at test time solves this problem. We trained

a variant model of PS-FCN, denoted as PS-FCN^{+N}, using the proposed data normalization strategy. Figure 3.4 (d) shows an example result that PS-FCN^{+N} performed better than PS-FCN on surfaces with SVBRDFs.

3.5 Dataset for Learning and Evaluation

The training of PS-FCN requires the ground-truth normal maps of the objects. However, obtaining ground-truth normal maps of real objects is a difficult and time-consuming task. Hence, we create two synthetic datasets for training and one synthetic dataset for testing. The publicly available real photometric stereo datasets are reserved to validate the generalization ability of our model. Experimental results show that our PS-FCN trained on the synthetic datasets generalizes well on the challenging real datasets.

3.5.1 Synthetic Data for Training

We used shapes from two existing 3D datasets, namely the blobby shape dataset [56] and the sculpture shape dataset [57], to generate our training data using the physically based raytracer Mitsuba [81]. Following DPSN [52], we employed the MERL dataset [58], which contains 100 different BRDFs of real-world materials, to define a diverse set of surface materials for rendering these shapes. Note that our datasets explicitly consider cast shadows during rendering. For the sake of data loading efficiency, we stored our training data in 8-bit PNG format.

Blobby dataset We first followed [52] to render our training data using the blobby shape dataset [56], which contains 10 blobby shapes with various normal distributions. For each blobby shape, 1,296 regularly-sampled views (36 azimuth angles \times 36 elevation angles) were used, and for each view, 2 out of 100 BRDFs were randomly selected, leading to 25,920 samples ($10 \times 36 \times 36 \times 2$). For each sample, we rendered 64 images with a spatial resolution of 128×128 under light directions randomly sampled from a range of $180^\circ \times 180^\circ$, which is more general than the range ($74.6^\circ \times 51.4^\circ$) used in the

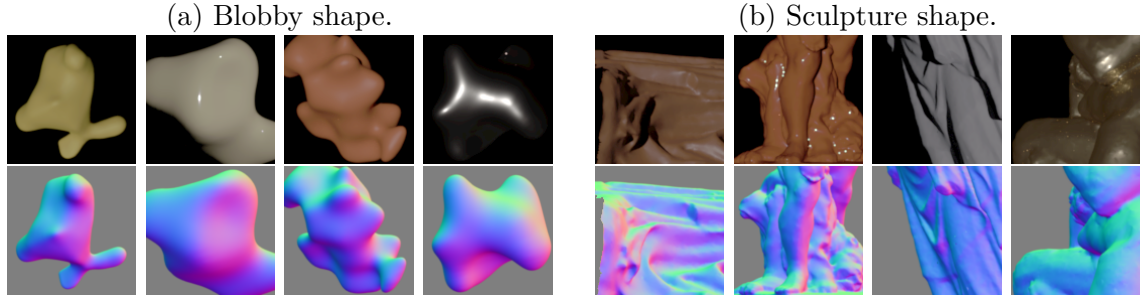


Fig. 3.6 Examples of the synthetic training data.

real data benchmark [59]. We randomly split this dataset into 99 : 1 for training and validation (see Fig. 3.6(a)).

Sculpture dataset The surfaces in the blobby shape dataset are usually largely smooth and lack of details. To provide more complex (realistic) normal distributions for training, we employed 8 complicated 3D models from the sculpture shape dataset introduced in [57]. We generated samples for the sculpture dataset in exactly the same way we did for the blobby shape dataset, except that we discarded views containing holes or showing uniform normals (*e.g.*, flat facets). The rendered images are with a size of 512×512 when a whole sculpture shape is in the field of view. We then regularly cropped patches of size 128×128 from the rendered images and discarded those with a foreground ratio less than 50%.³ This gave us a dataset of 59,292 samples, where each sample contains 64 images rendered under different light directions. Finally, we randomly split this dataset into 99 : 1 for training and validation (see Fig. 3.6(b)).

Data augmentation To narrow the gap between real and synthetic data, data augmentation was carried out on-the-fly during training. Given an image of size 128×128 , we randomly performed image rescaling (with the rescaled width and height within the range of $[32, 128]$, without preserving the original aspect ratio) and noise perturbation (in a range of $[-0.025, 0.025]$). Image patches of size 32×32 were then randomly cropped for training.

³Each training image in the blobby dataset shows part of an object or a whole object depending on the viewpoint, whereas each training image in the sculpture dataset shows only part of an object since the sculpture shapes are much larger than the blobby shapes.

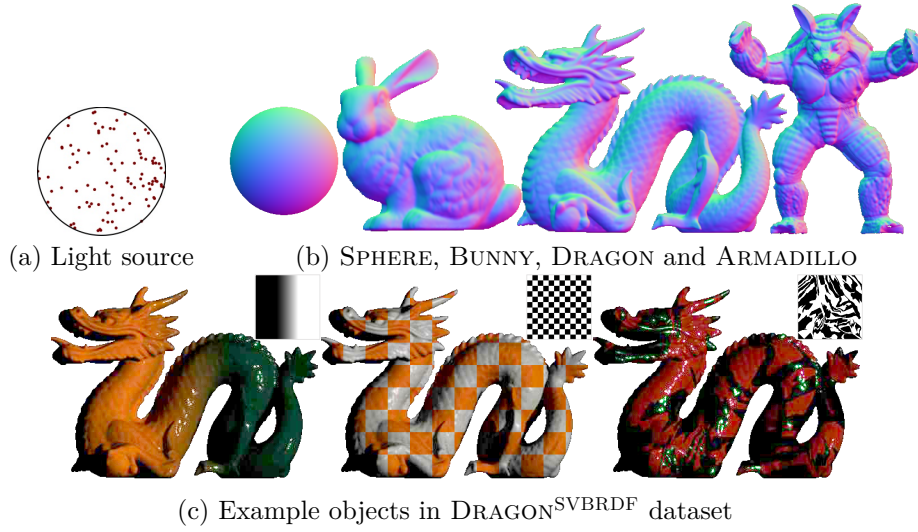


Fig. 3.7 (a) Lighting distribution of $\text{SynTest}^{\text{MERL}}$ dataset. The light direction is visualized by mapping a 3-d vector $[x, y, z]$ to a point $[x, y]$. (b) Ground-truth normals of SPHERE, BUNNY, DRAGON, and ARMADILLO. (c) Visualization of the selected material maps (Ramp, Checker, Irregular) and examples in $\text{DRAGON}^{\text{SVBRDF}}$ dataset.

3.5.2 Synthetic Data for Analysis

To quantitatively evaluate the performance of our method on different materials and shapes, we rendered a synthetic test dataset including Sphere, Bunny, Dragon, and Armadillo shapes. Hereafter, we denote this test dataset as $\text{SynTest}^{\text{MERL}}$ and these shapes as SPHERE, BUNNY, DRAGON, ARMADILLO respectively. Each shape was rendered with 100 isotropic BRDFs from MERL dataset [58] under 100 light directions randomly sampled from the upper-hemisphere, leading to 400 test objects (see Fig. 3.7 (a)-(b)). Cast shadows and inter-reflections were considered during rendering using the physically based raytracer Mitsuba [81].

To analyze how surfaces with SVBRDFs affect the performance of our method, we created another synthetic test dataset with SVBRDFs, denoted as $\text{DRAGON}^{\text{SVBRDF}}$, following [82]. Specifically, we blended two BRDFs from 100 MERL dataset for DRAGON using 3 materials maps, namely the *Ramp*, *Checker*, and *Irregular*, as shown in Fig. 3.7 (c). Note that for each material map, there are $C(100, 2) = 4,950$ combinations of two BRDFs, leading to 14,850 test objects.

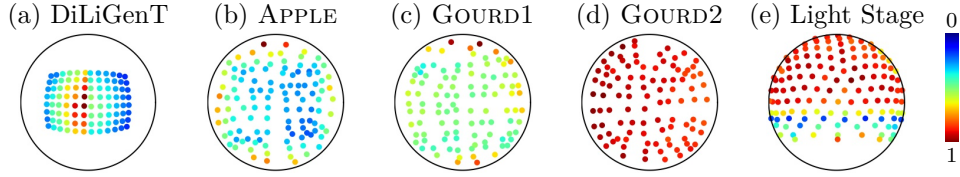


Fig. 3.8 Lighting distributions of the real testing datasets. The color of the point indicates the light intensity (value is divided by the highest intensity to normalize to $[0, 1]$).

3.5.3 Real Data for Testing

We employed three challenging real non-Lambertian photometric stereo datasets for testing, namely the *DiLiGenT benchmark* [59], *Gourd&Apple dataset* [60], and *Light Stage Data Gallery* [61]. Note that none of these datasets were used in the training.

DiLiGenT benchmark [59] is a public dataset containing 10 real objects, and each object was captured under 96 predefined light directions (see Fig. 3.8 (a)). Both ground-truth lighting conditions and normal maps are provided. We quantitatively evaluated the performance of our method on both lighting and normal estimation.

Gourd&Apple dataset [60] consists of three objects, namely APPLE, GOURD1, and GOURD2, with 112, 102 and 98 images, respectively. Figures 3.8 (b)-(d) visualize the lighting distributions of this dataset. Light Stage Data Gallery [61] is composed of six objects, and 253 images are provided for each object. We only used 133 images with the front side of the object under illumination. Figure 3.8 (e) visualizes the lighting distribution of the selected images. Since these two datasets only provide calibrated lightings (without ground-truth normal maps), we quantitatively evaluated our method on lighting estimation but only qualitatively evaluated it on normal estimation.

3.6 Experimental Results

In this section, we present experimental results and analysis. We carried out network analysis for PS-FCN on the synthetic test dataset, and compared our method with

the previous state-of-the-art methods on the DiLiGenT benchmark [59]. Mean angular error (MAE) in degree was used to measure the accuracy of the predicted normal maps. We further provided qualitative results on the Gourd&Apple dataset [60] and the Light Stage Data Gallery [61].

Implementation Details Our framework was implemented in PyTorch [83] with 2.2 million learnable parameters. We trained our model using a batch size of 32 for 30 epochs, and it only took a few hours for training to converge using a single NVIDIA Titan X Pascal GPU (*e.g.*, about 1 hour for 8 image-light pairs per sample on the blobby dataset, and about 9 hours for 32 image-light pairs per sample on both the blobby and sculpture datasets). Adam optimizer [42] was used with default parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.999$), where the learning rate was initially set to 0.001 and divided by 2 every 5 epochs.

3.6.1 Evaluation on Synthetic Data

We quantitatively analyzed PS-FCN on the synthetic dataset. In particular, we first validated the effectiveness of max-pooling in multi-feature fusion by comparing it with average-pooling and convolutional layers. We then investigated the influence of the complexity of training data, and the influence of input image number during training and testing. For all the experiments in network analysis, we performed 100 random trials (save for the experiments using all 100 image-light pairs per sample during testing) and reported the average results. Last, we analyzed the performance of PS-FCN on surfaces with cast shadows, SVBRDFs, and different materials.

Effectiveness of max-pooling We first validated the effectiveness of max-pooling in multi-feature fusion by comparing it with convolutional layers and average-pooling. Experiments with IDs A0 & A1 in Table 3.1 show that fusion by convolutional layer on the concatenated features was sub-optimal. This could be explained by the fact that the weights of the convolutional layer are related to the order of the input features,

Table 3.1 Normal estimation results of different variant models of PS-FCN on SynTest^{MERL} dataset. The results are averaged over samples rendered with 100 BRDFs. B and S stand for the blobby and sculpture training datasets, respectively.

Model Variants					Test Objects			
ID	Data	Fusion	Train #	Test #	SPHERE	BUNNY	DRAGON	ARMOD.
A0	B	Conv	32	32	4.54	6.74	9.57	9.87
A1	B	Max-p	32	32	3.65	5.33	7.86	8.09
A2	B	Avg-p	32	100	3.71	5.36	8.17	7.92
A3	B	Max-p	32	100	3.40	4.80	7.23	7.21
A4	B+S	Max-p	32	100	2.66	3.80	4.83	5.24

while the order of the input image-light pairs is random in our case, thus increasing the difficulty for the convolutional layer to find the relations among multiple features. Experiments with IDs A2 & A3 compared the performance of average-pooling and max-pooling for multi-feature fusion. It can be seen that max-pooling performs consistently better than average-pooling on SynTest^{MERL} dataset. Figure 3.9 visualizes the fused features by max-pooling for four objects with different shapes and reflectances. We can see that each channel of the fused features can be interpreted as the probability of the normal belonging to a certain direction, and max-pooling can naturally aggregate such information from multiple observations.

Effects of training data and input image number By comparing experiments with ID A3 & A4 in Table 3.1, we can see that training with the additional sculpture dataset that has a more complex normal distribution helped to boost the performance of PS-FCN. This result suggests that the performance of PS-FCN could be further improved by introducing more complex and realistic training data.

Figure 3.10 (a) shows that for a fixed number of inputs during testing, PS-FCN performs better when the number of inputs during training is close to that during testing. It is worth noting that when there is only one input image, the problem reduces to the more challenging shape-from-shading problem. Figure 3.10 (a) shows that PS-FCN performs best when the training image number is also 1, with an average MAE of 18.75° for SPHERE. However, this result is moderately inaccurate, indicating

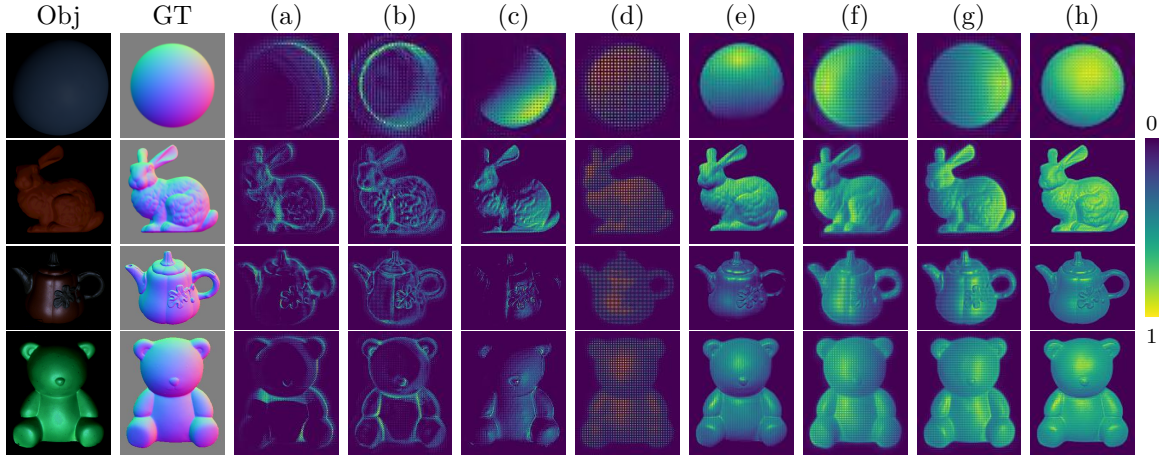


Fig. 3.9 Visualization of the learned feature map after fusion. The first two columns show the images and ground-truth normal maps. Each of the subsequent columns (a-h) shows one particular channel of the fused feature map. 8 out of the 128 channels of the feature map are presented. Note that different regions with similar normal directions are fired in different channels. Each channel can therefore be interpreted as the probability of the normal belonging to a certain direction (or alternatively as the object shading rendered under a certain light direction). Accurate normal maps can then be inferred from these probability distributions.

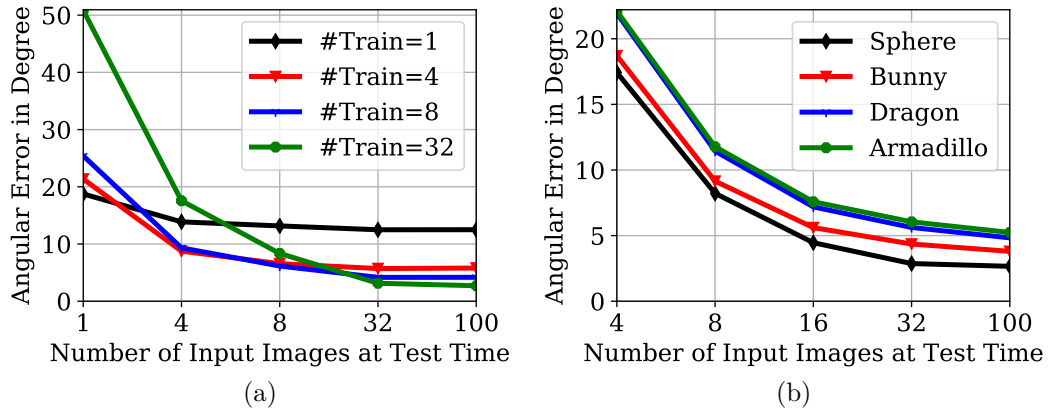


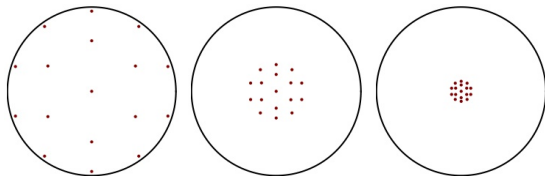
Fig. 3.10 (a) Results of PS-FCN trained and tested with different numbers of input images on SPHERE. (b) Results of PS-FCN trained with a fixed number of 32 input images and tested with different numbers of input images.

that PS-FCN has difficulties in resolving the ambiguity in the problem of shape from shading.

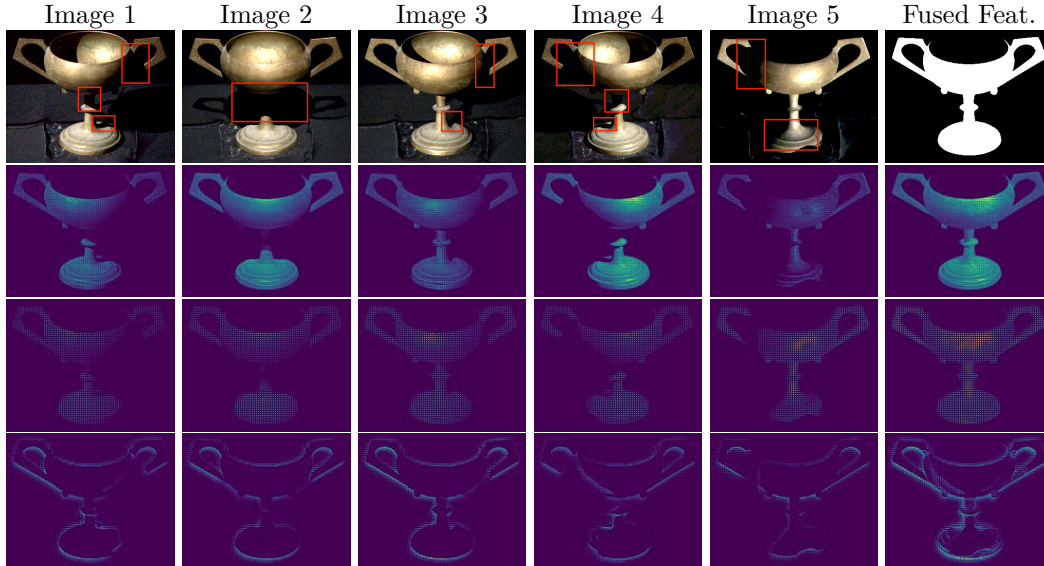
Figure 3.10 (b) shows that for a fixed number of inputs during training, the performance of PS-FCN increases with the number of inputs during testing. This is a desired property for photometric stereo as we can simply capture more images for robust estimation. For the rest of this chapter, we refer PS-FCN as the model trained on both datasets and with an input of 32 image-light pairs per sample.

Effects of lighting distributions We tested PS-FCN on BUNNY rendered with three different lighting distributions, as shown in Table 3.2. These three distributions have the same number of light source (*i.e.*, 17), but with different spanning ranges. We can see that PS-FCN performs better when lightings are more diversely distributed. For the highly clustered distribution (see Table 3.2 (c)), the results of PS-FCN drops notably. Since the lightings are randomly sampled from the upper-hemisphere (*i.e.*, spanning range of $180^\circ \times 180^\circ$) during training, it is therefore not surprising to see PS-FCN with decreased performance under this extreme lighting distribution.

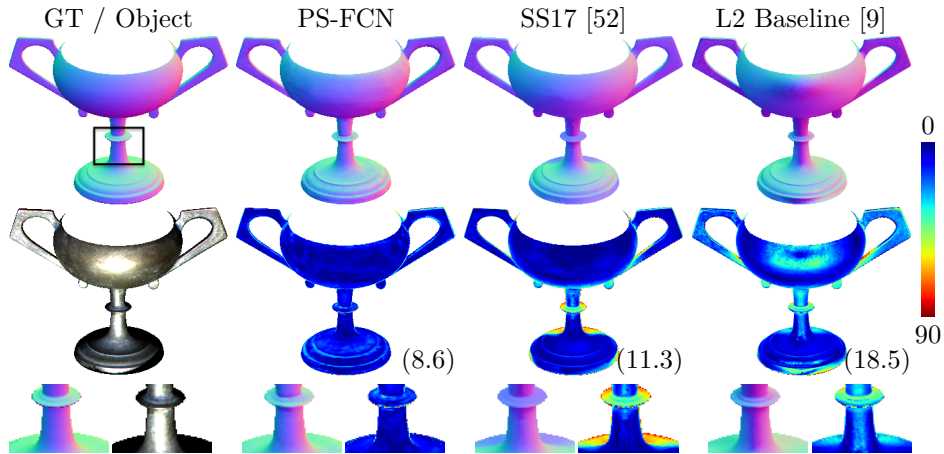
Table 3.2 Results of PS-FCN on BUNNY rendered using three different lighting distributions.

			Type	Range	MAE
	(a)		(a)	$144^\circ \times 144^\circ$	4.21
	(b)		(b)	$37^\circ \times 37^\circ$	10.90
	(c)		(c)	$22^\circ \times 22^\circ$	18.72
			(d)	Normal estimation	

Results on surface with cast shadows The presence of cast shadow is almost inevitable when the geometry of the object is non-convex, and is one of the major difficulties in photometric stereo. Given the observation that a real surface point is unlikely to be shadowed under all light directions, we argue that max-pooling fusion can naturally overcome the effect of cast shadow when determining the surface normals. This is because even a surface point is shadowed under some light directions,



(a) The first five columns show the input images and the extracted features for each image (only 3 out of 128 feature channels are shown). The last column shows the object mask and the fused features by max-pooling. Red boxes in the images indicate regions with cast shadows.



(b) Comparison between PS-FCN, SS17 [52] and L2 Baseline [9] on GOBLET. The first row shows the ground-truth and estimated normals, and the second row shows the object and the error maps.

Fig. 3.11 Illustration of how max-pooling fusion layer handles surface regions with cast shadow using GOBLET from the DiLiGenT benchmark. (Note that the provided object mask and ground-truth normal map do not include the concave interior of GOBLET.)

it can be observed under other light directions, and max-pooling can ignore those non-activated features and aggregate those activated features. Figure 3.11 (a) visualizes how max-pooling aggregates features from multiple observations and handles cast shadow. Compared with L2 baseline [9] and SS17 [52], our method is more robust in regions with cast shadow (see Fig. 3.11 (b)).

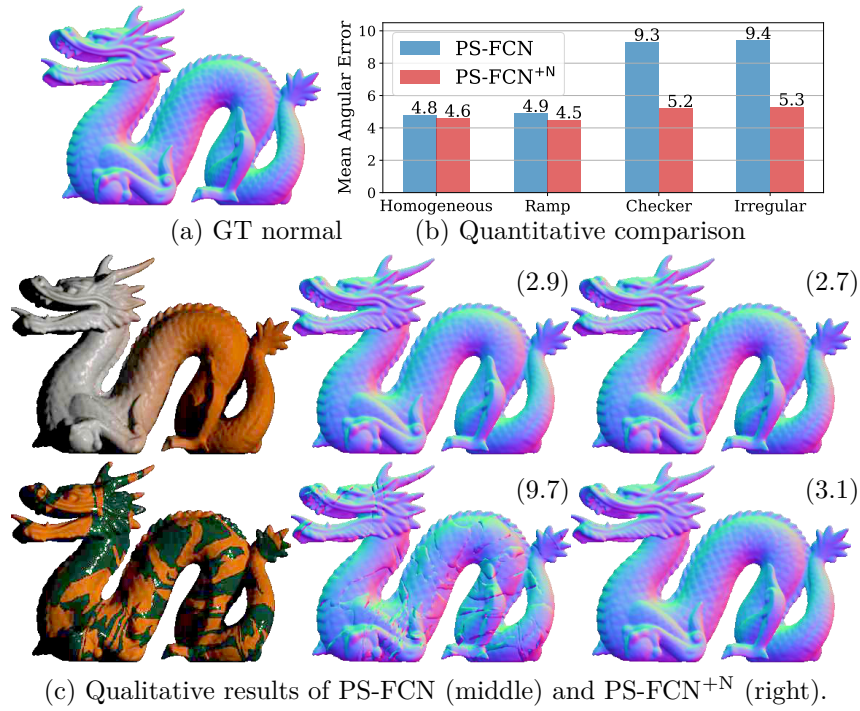
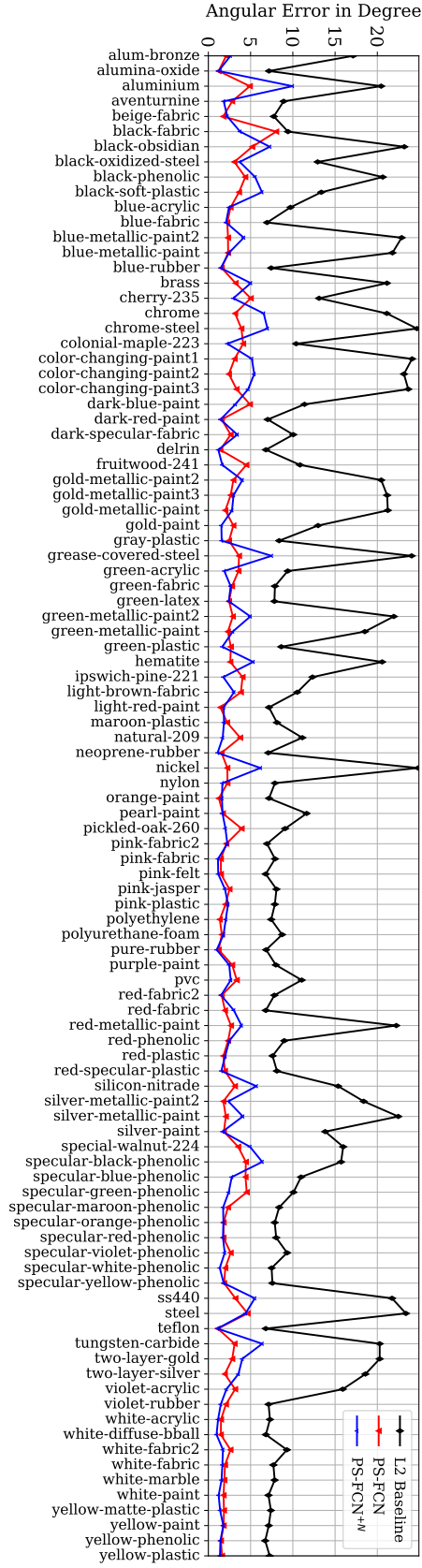


Fig. 3.12 Comparison between PS-FCN and PS-FCN^{+N} on DRAGON^{SVBRDF} dataset.

Results on surfaces with SVBRDFs To analyze how PS-FCN deteriorates in dealing with surfaces with SVBRDFs and verify the effectiveness of the proposed data normalization strategy, we compared PS-FCN and PS-FCN^{+N} on DRAGON^{SVBRDF} dataset and the results are summarized in Fig. 3.12. We can see that both models perform well on surfaces with homogeneous materials or surfaces with smooth BRDF changes (*e.g.*, surfaces blended with Ramp). However, PS-FCN has difficulty in dealing with steep color changes caused by SVBRDFs (*e.g.*, surfaces blended with Checker and Irregular). In contrast, PS-FCN^{+N} is robust in handling surfaces with different types of SVBRDFs, which clearly demonstrates the effectiveness of the proposed data

Fig. 3.13 Quantitative results on SPHERE rendered with 100 MERL BRDFs. The average MAE for L2 Baseline [9], PS-FCN, and PS-FCN^{+N} are 12.59, 2.66 and 2.91, respectively.



normalization strategy.

Results on different materials Figure 3.13 compares PS-FCN, PS-FCN^{+N}, and L2 Baseline [9] on SPHERE that was rendered with 100 different BRDFs. It can be seen that PS-FCN and PS-FCN^{+N} achieved comparable results on different materials, which indicates that training with data normalization strategy will not worsen the results on homogeneous surfaces. Besides, both models significantly outperformed the L2 Baseline [9].

3.6.2 Evaluation on Real Data

Table 3.3 Quantitative comparison of calibrated photometric stereo on the DiLiGenT benchmark. The numbers represent the MAE (the lower the better).

Method	BALL	CAT	POT1	BEAR	POT2	BUDDHA	GOBLET	READING	COW	HARVEST	Average
L2 [9]	4.1	8.4	8.9	8.4	14.7	14.9	18.5	19.8	25.6	30.6	15.4
AZ08 [60]	2.7	6.5	7.2	6.0	11.0	12.5	13.9	14.2	21.5	30.5	12.6
WG10 [63]	2.1	6.7	7.2	6.5	13.1	10.9	15.7	15.4	25.9	30.0	13.4
IA14 [70]	3.3	6.7	6.6	7.1	8.8	10.5	9.7	14.2	13.1	26.0	10.6
ST14 [69]	1.7	6.1	6.5	6.1	8.8	10.6	10.1	13.6	13.9	25.4	10.3
SS17 [52]	2.0	6.5	7.1	6.3	7.9	12.7	11.3	15.5	8.0	16.9	9.4
TM18 [54]	1.5	5.4	6.1	5.8	7.8	10.4	11.5	11.0	6.3	22.6	8.8
HS17 [84]	1.3	4.9	5.2	5.6	6.4	8.5	7.6	12.1	8.2	15.8	7.6
IS18* [53]	2.2	4.6	5.4	8.3	6.0	7.9	7.3	12.6	8.0	14.0	7.6
PS-FCN	2.8	6.2	7.1	7.6	7.3	7.9	8.6	13.3	7.3	15.9	8.4
PS-FCN ^{+N}	2.7	4.8	6.2	7.7	7.2	7.5	7.8	10.9	6.7	12.4	7.4

* indicates that the results of IS18 [53] on BEAR was computed using all of the 96 images. The result reported in IS18 [53] (BEAR: 4.1, Avg.: 7.2) was evaluated by discarding the first 20 images. When discarding the first 20 images, our results are PS-FCN (BEAR: 5.0, Avg.: 8.1) and PS-FCN^{+N} (BEAR: 5.0, Avg.: 7.1).

Comparison on DiLiGenT Benchmark We compared our method against the recently proposed learning based methods [52–54] and other previous state-of-the-art methods on the DiLiGenT benchmark, as shown in Table 3.3. After training with the data normalization strategy, PS-FCN^{+N} performs better than PS-FCN on almost

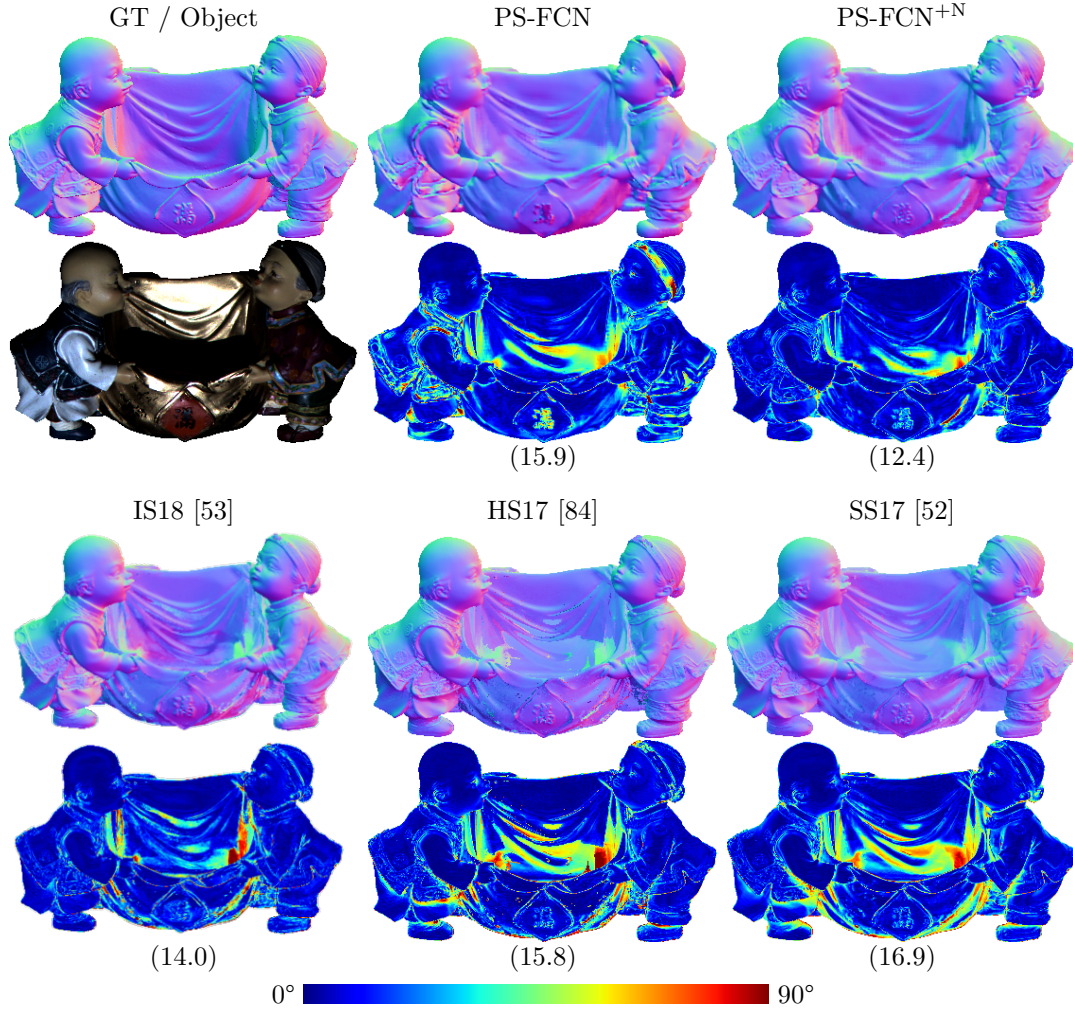


Fig. 3.14 Qualitative results on HARVEST in the DiLiGenT benchmark. Compared with PS-FCN, PS-FCN^{+N} performs better for surfaces with SVBRDFs. In contrast to those per-pixel normal estimation methods [52, 53, 84], PS-FCN^{+N} can take advantage of the surface smooth prior and estimate a smoother normal map with less noise artifacts. Numbers in parentheses denote MAE in degree.

all of the ten objects, except for BEAR. Compared with the other state-of-the-art methods, PS-FCN^{+N} performs particularly well on surface with complex geometry and/or SVBRDFs (*e.g.*, BUDDHA, READING, and HARVEST), and achieves state-of-the-art results with an average MAE of 7.4. Qualitative comparison on HARVEST is shown in Fig. 3.14. Note that PS-FCN did not outperform previous methods on all the 10 objects. We hypothesize that this might be caused by the limited training data. Different from pixel-wise approaches like IS18 [53] and HS17 [84], our method relies on diverse surface patches for training, while the current training data are only generated from 18 objects.

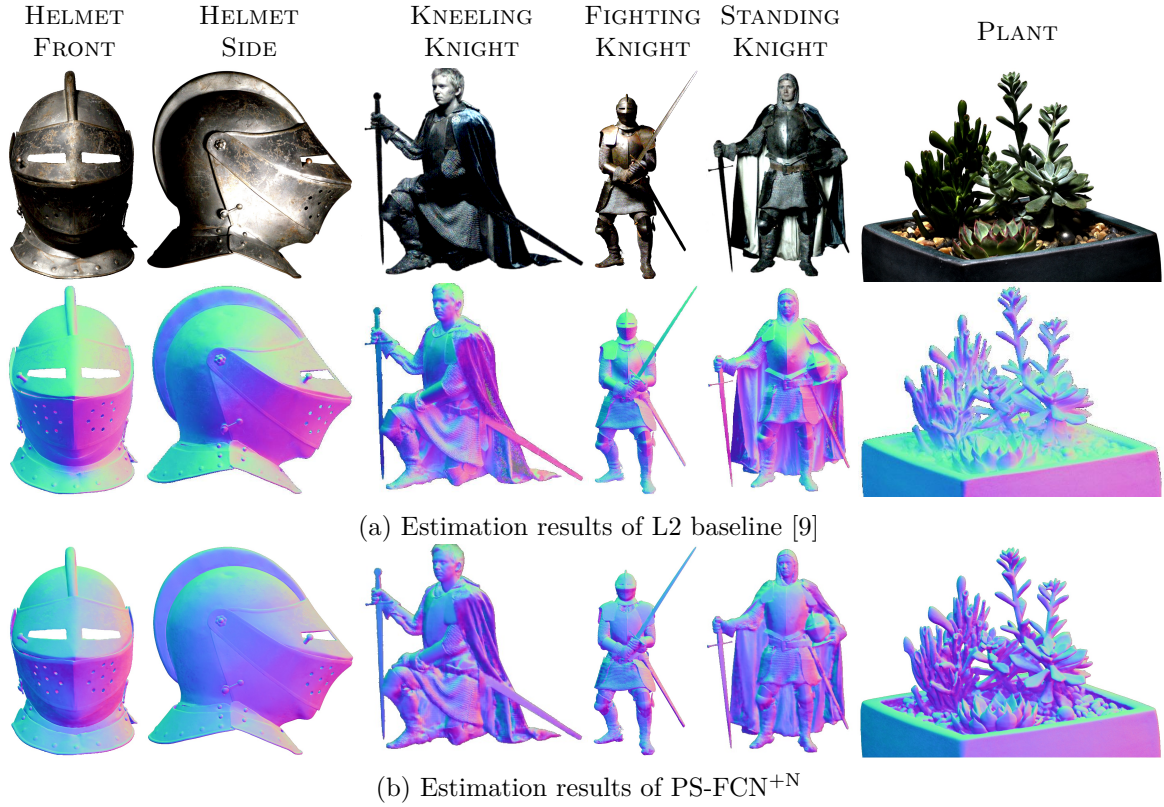


Fig. 3.15 Qualitative results of calibrated photometric stereo on Light Stage Data Gallery.

Evaluation on other real datasets Due to absence of ground-truth normal maps, we qualitatively evaluated our best-performing model PS-FCN^{+N} on the Gourd&Apple dataset [60] and the Light Stage Data Gallery [61] (see Fig. 3.15 and Fig. 3.16). Our

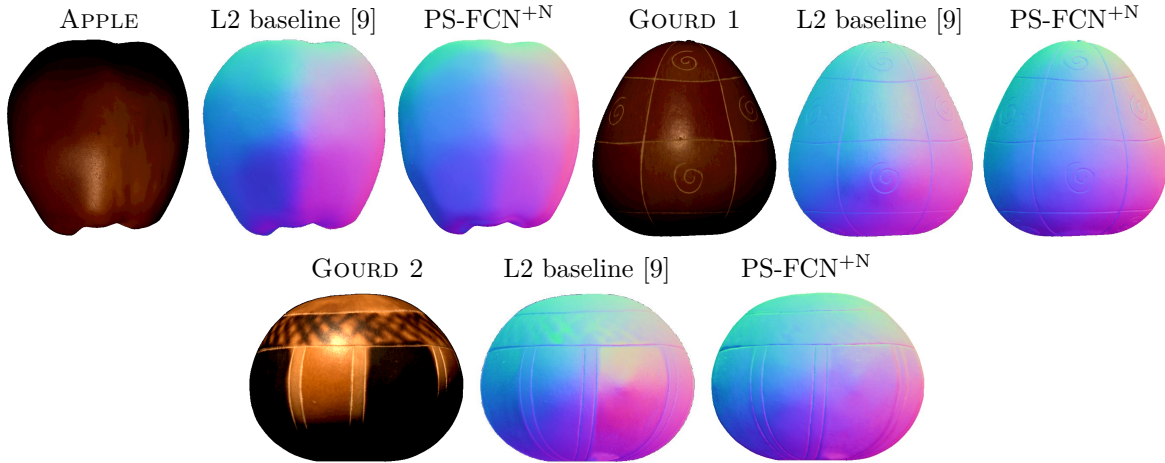


Fig. 3.16 Qualitative results of calibrated photometric stereo on Gourd&Apple dataset.

method can estimate visually pleasing and consistent surface normals for these two challenging datasets, while the L2 baseline [9] produces seemingly inaccurate surface normals for regions with specular highly or strong cast shadow (see PLANT in Fig. 3.15 for example).

Runtime comparison Table 3.4 compares the runtime of four different deep learning methods in estimating a normal map with 612×512 pixels. We can see that our method runs significantly faster than the online optimization based method [54] and more than $2\times$ faster than the per-pixel method [52, 53].

Table 3.4 Runtime comparison of different methods in estimating a normal map with 612×512 pixels.

Method	SS17 [52]	TM18 [54]	IS18 [53]	PS-FCN
Runtime	4 seconds	~ 1 hour	16 seconds	1.5 seconds

3.6.3 Extension for Uncalibrated Photometric Stereo

PS-FCN can be easily extended to handle uncalibrated photometric stereo by simply removing the light directions from the input. To verify the potential of our framework towards uncalibrated photometric stereo, we trained an uncalibrated variant of

Table 3.5 Comparison of results for uncalibrated photometric stereo on the DiLiGenT benchmark. The numbers represent the MAE (the lower the better). The results of a calibrated method PS-FCN are included in the last row as reference. Bold font indicates the best results in the uncalibrated methods.

Method	BALL	CAT	POT1	BEAR	POT2	BUDDHA	GOBLET	READING	COW	HARVEST	Average
AM07 [85]	7.27	31.45	18.37	16.81	49.16	32.81	46.54	53.65	54.72	61.70	37.25
SM10 [86]	8.90	19.84	16.68	11.98	50.68	15.54	48.79	26.93	22.73	73.86	29.59
WT13 [87]	4.39	36.55	9.39	6.42	14.52	13.19	20.57	58.96	19.75	55.51	23.93
PF14 [88]	4.77	9.54	9.51	9.07	15.90	14.92	29.93	24.18	19.53	29.21	16.66
LC18 [89]	9.30	12.60	12.40	10.90	15.70	19.00	18.30	22.30	15.00	28.00	16.30
UPS-FCN	6.62	14.68	13.98	11.23	14.19	15.87	20.72	23.26	11.91	27.79	16.02
PS-FCN	2.8	6.2	7.1	7.6	7.3	7.9	8.6	13.3	7.3	15.9	8.4

our model, denoted as UPS-FCN, taking only images as input (note that we assume the images were normalized by the light intensities). UPS-FCN was trained on both synthetic datasets using 32 image-light pairs as input. We compared our UPS-FCN with the existing uncalibrated methods. The results are reported in Table 3.5, our UPS-FCN outperformed existing methods in terms of the average MAE, which demonstrates the flexibility of our model.

However, the performance of the UPS-FCN lags far behind the calibrated model PS-FCN, which takes both images and light directions as input.

3.7 Conclusion

In this work, we have proposed a flexible deep fully convolutional network, named PS-FCN, that accepts an arbitrary number of images and their associated light directions as input and regresses an accurate normal map. Our PS-FCN does not require a pre-defined set of light directions during training and testing, and allows both the number of lights and their directions used in testing different from that used in training. It can handle multiple images and light directions in an order-agnostic manner. In order to train PS-FCN, two synthetic datasets with various realistic shapes and materials have been created. After training, PS-FCN can generalize well on challenging real datasets. In addition, PS-FCN can be easily extended to handle uncalibrated photometric stereo. Results on diverse real datasets have clearly shown that PS-FCN outperforms previous

calibrated photometric stereo methods, and promising results have been achieved in uncalibrated scenario.

Chapter 4

Learning Lighting Calibration for Photometric Stereo

4.1 Introduction

Photometric stereo aims at recovering the surface normal of a static object from a set of images captured under different light directions [9, 14]. *Calibrated* photometric stereo methods assume known light directions, and promising results have been reported [59] at the cost of tedious light source calibration. The problem of *uncalibrated* photometric stereo, where light directions are unknown, still remains an open challenge, and its stable solution is wanted because of the ease of setting. In this chapter, we study the problem of uncalibrated photometric stereo for surfaces with general and unknown isotropic reflectance.

Most of the existing methods for uncalibrated photometric stereo [85, 86, 88] assume a simplified reflectance model, such as the Lambertian model, and focus on resolving the shape-light ambiguity, such as the Generalized Bas-Relief (GBR) ambiguity [90]. Although methods of [80, 91] can handle surfaces with general bidirectional reflectance distribution functions (BRDFs), they rely on a uniform distribution of light directions for deriving a solution.

Recently, with the great success of deep learning in various computer vision tasks,

deep learning based methods have been introduced to calibrated photometric stereo [17, 52–54]. Instead of explicitly modeling complex surface reflectances, they directly learn the mapping from reflectance observations to surface normals given light directions. Although they have produced promising results in a calibrated setting, they cannot handle the more challenging problem of *uncalibrated* photometric stereo, where light directions and intensities are unknown. One simple strategy to handle uncalibrated photometric stereo with deep learning is to directly learn the mapping from images to surface normals without taking the light directions as input. However, as reported in Section 3.6.3, the performance of such a model lags far behind those which take both images and light directions as input.

Instead of directly predicting surface normals from images, we propose to first estimate light directions and intensities from images. The problem of uncalibrated photometric stereo can then be reduced to a calibrated one, which can be effectively solved by existing calibrated methods [17, 53, 69]. The rationales behind this two-stage approach are as follows. First, lighting information is very important for normal estimation since lighting is the source of various cues, such as shading and reflectance, and estimating the light directions (3-vectors) and intensities (scalars) is in principle much easier than directly estimating the normal map (a 3-vector at each pixel location) together with the lighting conditions. Second, by explicitly learning to estimate light directions and intensities, the model can take advantage of the intermediate supervision by the ground-truth lighting, resulting in a more interpretable behavior.

This work focuses on learning lighting calibration for uncalibrated photometric stereo. The contributions of this work can be summarized as follows:

- We introduce a lighting calibration network, named LCNet, for estimating light directions and intensities from images.
- We discuss the differences between LCNet and traditional uncalibrated methods, and analyze the features learned by LCNet to resolve the GBR ambiguity.
- We find that attached shadows, shadings, and specular highlights are key ele-

ments for lighting estimation, and that LCNet extracts features independently from each input image without exploiting any inter-image information (“inter-image” means information shared by all images).

- Based on our findings, we propose a guided calibration network (GCNet) that explicitly utilizes object shape and shading information as guidances for better lighting estimation.

Preliminary results of this research have been published in [19, 20]. Our code and models can be found at <https://guanyingc.github.io/SDPS-Net>.

4.2 Related Work

In this section, we review uncalibrated photometric stereo methods and the loosely related work on learning based lighting estimation. Review for recent deep calibrated photometric stereo methods can be found in Section 3.2.

Uncalibrated photometric stereo When lighting is unknown, the surface normals of a Lambertian object can only be estimated up to a 3×3 linear ambiguity [10], which can be reduced to a 3-parameter GBR ambiguity [90, 92] using the surface integrability constraint. Previous work used additional clues like albedo priors [85, 86], inter-reflections [93], specular spikes [94], Torrance and Sparrow reflectance model [68], reflectance symmetry [87, 95], multi-view images [96], and local diffuse maxima [88], to resolve the GBR ambiguity. Cho *et al.* [97] considered a semi-calibrated case where the light directions are known but not their intensities. There are few works that can handle non-Lambertian surfaces under unknown lighting. Hertzmann and Seitz [72] proposed an exemplar based method by inserting an additional reference object to the scene. Methods based on cues like similarity in radiance changes [79, 80] and attached shadow [98] were also introduced, but they require the light sources to be uniformly distributed on the entire viewing sphere. Recently, Lu *et al.* [89] introduced a method based on the “constrained half-vector symmetry” to work with non-uniform

lightings. Different from these traditional methods, our method can deal with surfaces with general and unknown isotropic reflectance without the need of explicitly utilizing any additional clues or reference objects, solving a complex optimization problem at test time, or making assumptions on the light source distribution.

Other methods related to uncalibrated photometric stereo include exemplar-based methods [72], regression-based methods [99], semi-calibrated photometric stereo [100], inaccurate lighting refinement [101], and photometric stereo under general lighting [102–104].

Learning based lighting estimation Recently, learning based single-image lighting estimation methods have attracted considerable attention. Gardner *et al.* [105] introduced a CNN for estimating HDR environment lighting from an indoor scene image. Hold-Goeffroy *et al.* [106] learned outdoor lighting using a physically-based sky model. Weber *et al.* [107] estimated indoor environment lighting from an image of an object with known shape. Zhou *et al.* [108] estimated lighting, in the form of Spherical Harmonics, from a human face image by assuming a Lambertian reflectance model. Different from the above methods, our method can estimate accurate directional lightings from multiple images of a static object with general shape and non-Lambertian surface.

4.3 Lighting Calibration Network (LCNet)

In the rest of this chapter, we refer to light direction and intensity as “lighting”. To estimate lighting from the images, an intuitive approach would be directly regressing the light direction vectors and intensity values. However, we propose that formulating the lighting estimation as a classification problem is a superior choice, as will be verified by our experiments. Our arguments are as follows. First, classifying a light direction into a certain range is easier than regressing the exact value(s), and this will reduce the learning difficulty. Second, when training a normal estimation network, taking discretized light directions as input will allow it to better tolerate small errors in the

estimated light directions.

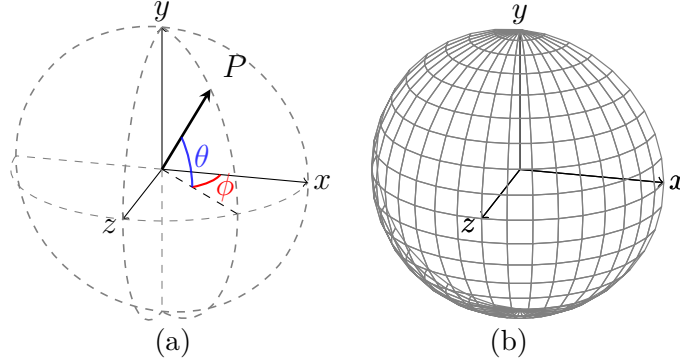


Fig. 4.1 (a) Illustration of the coordinate system (z axis is the viewing direction). $\phi \in [0^\circ, 180^\circ]$ and $\theta \in [-90^\circ, 90^\circ]$ are the azimuth and elevation of the light direction, respectively. (b) Example discretization of the light direction space when $K_d = 18$.

4.3.1 Discretization of Lighting Space

Since we cast our lighting estimation as a classification problem, we need to discretize the continuous lighting space. Note that a light direction in the upper-hemisphere can be described by its azimuth $\phi \in [0^\circ, 180^\circ]$ and elevation $\theta \in [-90^\circ, 90^\circ]$ (see Fig. 4.1 (a)). We can discretize the light direction space by evenly dividing both the azimuth and elevation into K_d bins, resulting in K_d^2 classes (see Fig. 4.1 (b)). Solving a K_d^2 -class classification problem is not computationally efficient, as the softmax probability vector will have a very high dimension even when K_d is not large (*e.g.*, $K_d^2 = 1,296$ when $K_d = 36$). Instead, we estimate the azimuth and elevation of a light direction separately, leading to two K_d -class classification problems. Similarly, we evenly divide the range of possible light intensities into K_e classes (*e.g.*, $K_e = 20$ for a possible light intensity range of $[0.2, 2.0]$).

4.3.2 Local-global Feature Fusion

A straightforward approach to estimate the lighting for each image is simply taking a single image as input, encoding it into a feature map using a CNN, and feeding

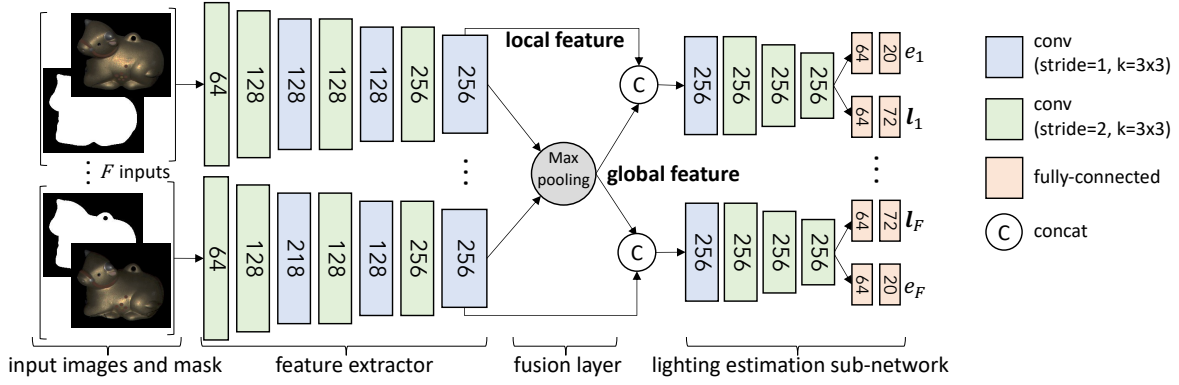


Fig. 4.2 Network architecture of LCNet. Each layer’s value indicates its output channel number.

the feature map to a lighting prediction layer. It is not surprising that the result of such a simple solution is far from satisfactory. Note that the appearance of an object is determined by its surface geometry, reflectance model and the lighting. The feature map extracted from a single observation obviously does not provide sufficient information for resolving the shape-light ambiguity. Thanks to the nature of photometric stereo where multiple observations of an object are considered, we propose a local-global feature fusion strategy to extract more comprehensive information from multiple observations.

Specifically, we separately feed each image into a shared-weight feature extractor to extract a feature map, which we call *local feature* as it only provides information from a single observation. All local features of the input images are then aggregated into a *global feature* through a max-pooling operation, which has been proven to be efficient and robust on aggregating salient features from a varying number of unordered inputs [17, 57]. Such a global feature is expected to convey implicit surface geometry and reflectance information of the object which help resolve the ambiguity in lighting estimation. Each local feature is concatenated with the global feature, and fed to a shared-weight lighting estimation sub-network to predict the lighting for each individual image. By taking both local and global features into account, our model can produce much more reliable results than using the local features alone. We empirically found that additionally including the object mask as input can effectively improve the

performance of lighting estimation, as will be seen in the experiment section.

4.3.3 Network Architecture

LCNet is a multi-input-multi-output (MIMO) network that consists of a shared-weight *feature extractor*, an *aggregation layer* (*i.e.*, max-pooling layer), and a shared-weight *lighting estimation sub-network* (see Fig. 4.2). It takes the observations of the object together with the object mask as input, and outputs the light directions and intensities in the form of softmax probability vectors of dimension K_d (azimuth), K_e (elevation) and K_e (intensity), respectively. We convert the output of LCNet to 3-vector light directions and scalar intensity values by simply taking the middle value of the range with the highest probability. We have experimentally verified that alternative ways like taking the expectation of the probability vector or performing quadratic interpolation in the neighborhood of the peak value do not improve the result.

Loss function Multi-class cross entropy loss is adopted for both light direction and intensity estimation, and the overall loss function is

$$\mathcal{L}_{\text{Light}} = \lambda_{l_a} \mathcal{L}_{l_a} + \lambda_{l_e} \mathcal{L}_{l_e} + \lambda_e \mathcal{L}_e, \quad (4.1)$$

where \mathcal{L}_{l_a} and \mathcal{L}_{l_e} are the loss terms for azimuth and elevation of the light direction, and \mathcal{L}_e is the loss term for light intensity. For example, given F input images,

$$\mathcal{L}_{l_a} = -\frac{1}{F} \sum_{f=1}^F \sum_{i=1}^{K_d} \{y_i^f = 1\} \log(p_i^f), \quad (4.2)$$

where $\{\cdot\}$ is a binary indicator (0 or 1) function, y_i^f is the ground-truth label (0 or 1) and p_i^f is the predicted probability for bin i (K_d bins in our case) for the f^{th} image. Detailed definition of other loss terms are similar to Eq. (4.2). During training, weights λ_{l_a} , λ_{l_e} , and λ_e for the loss terms are set to 1.

4.3.4 Training Data

We adopted the synthetic Blobby and Sculpture datasets introduced in Section 3.5 for training. Blobby and Sculpture datasets provide surfaces with complex normal distributions and diverse materials from MERL dataset [58]. Effects of cast shadow and inter-reflection were considered during rendering using the physically based ray-tracer Mitsuba [81]. There are 85,212 samples in total. Each sample was rendered under 64 distinct light directions sampled from the upper-hemisphere with uniform light intensity, resulting in 5,453,568 images ($85,212 \times 64$). The rendered images have a dimension of 128×128 .

To simulate images under different light intensities, we randomly generated light intensities in the range of $[0.2, 2.0]$ to scale the magnitude of the images (*i.e.*, the ratio of the highest light intensity to the lowest one is 10)¹. Note that this selected range contains a wider range of intensity value than the public photometric stereo datasets like DiLiGenT benchmark [59] and Gourd&Apple dataset [60]. The color intensities of the input images were normalized to the range of $[0, 1]$. During training, we applied noise perturbation in the range of $[-0.025, 0.025]$ for data augmentation, and the input image size for LCNet was 128×128 . At test time, the input for LCNet is rescaled to 128×128 as it contains fully-connected layers and requires the input to have a fixed spatial dimension. Trained only on the synthetic dataset, we will show that our model can generalize well on real datasets.

Implementation details Our method was implemented in PyTorch [83] and Adam optimizer [42] was used with default parameters. LCNet contains 4.4 million parameters. We trained LCNet using a batch size of 32 for 20 epochs, and the learning rate was initially set to 0.0005 and halved every 5 epochs. It took about 22 hours to train LCNet on a single Titan X Pascal GPU with a fixed input image number of 32.

¹Note that the ratio (other than the exact value) matters, since light intensity can only be estimated up to a scale factor.

4.3.5 Evaluation of LCNet with Synthetic Data

We evaluate LCNet on the synthetic test dataset $\text{SynTest}^{\text{MERL}}$ introduced in Section 3.5.2. To measure the accuracy of the predicted light directions, the widely used mean angular error (MAE) in degree is adopted. Since the light intensities among the testing images can only be estimated up to a scale factor s , we introduce the scale-invariant relative error (RE)

$$RE_{scale} = \frac{1}{q} \sum_i^q \left(\frac{|se_i - \tilde{e}_i|}{\tilde{e}_i} \right), \quad (4.3)$$

where q is the number of images, e_i and \tilde{e}_i are the estimated and ground-truth light intensities, respectively, for image i . The scale factor s is computed by solving $\underset{s}{\operatorname{argmin}} \sum_i^n (se_i - \tilde{e}_i)^2$ with least squares. As the calibrated intensity in the real dataset is in the form of a 3-vector, we repeat the estimated intensity to form a 3-vector and calculate the average result.

For all experiments on synthetic dataset involving input with unknown light intensities, we randomly generated light intensities in the range of $[0.2, 2.0]$. Each experiment was repeated five times and the average results were reported.

Discretization of lighting space For a given number of bins K_d , the maximum deviation angle for azimuth and elevation of a light direction is $\delta = 180^\circ / (K_d \times 2)$ after discretization (*e.g.*, $\delta = 2.5^\circ$ when $K_d = 36$). Note that discretizing azimuth and elevation angles independently indicates that lighting space is more densely discretized around the poles and less around the equator. This suggests that the link between the quantization of the lighting space and surface normal estimation error correlates with the lighting distribution. To investigate how the light direction discretization affects the surface normal estimation accuracy, we tested PS-FCN on SPHERE and BUNNY rendered under three different lighting distributions, namely, *Near Uniform*, *Around Equator*, and *Around Poles* (see Fig. 4.3 (a)).

We divided the azimuth and elevation angles of light directions into different num-

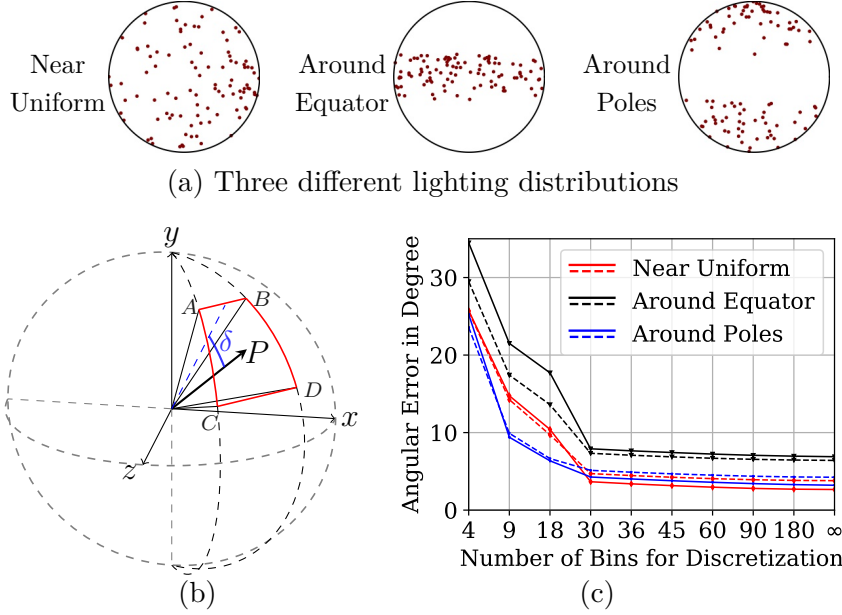


Fig. 4.3 (a) Three different lighting distributions. (b) Light directions A , B , C , and D have the maximum deviation angles with the actual light direction P after discretization. (c) Normal estimation error of PS-FCN on SPHERE (solid lines) and BUNNY (dashed lines) under different light direction space discretization levels (∞ indicates no discretization).

bers of bins ranging from 4 to 180. For a specific bin number, we perturbed the azimuth and elevation of each ground-truth light direction by the maximum deviation angle, leading to four light directions that have the maximum possible angular deviations after discretization (see Fig. 4.3 (b)). We then used these light directions as input for PS-FCN to infer surface normals. The normal estimation error reported in Fig. 4.3 (c) is the upper-bound error for PS-FCN caused by discretization. We can see that the error increase caused by discretization is marginal for all three lighting distributions when $K_d \geq 30$. We chose a relatively sparse discretization of lighting space in this work as it allows PS-FCN to learn to better tolerate small errors in the estimated lighting at test time.

Effectiveness of LCNet We first investigated the effect of object mask input and local-global feature fusion. Table 4.1 shows that taking the object mask as input and adopting the proposed local-global feature fusion strategy can effectively improve

Table 4.1 Lighting estimation results (MAE in degree for light direction and relative error for intensity) on SynTest^{MERL} dataset. The results are averaged over samples rendered with 100 BRDFs. (Value the lower the better)

Model	SPHERE		BUNNY		DRAGON		ARMADILLO	
	Direction	Intensity	Direction	Intensity	Direction	Intensity	Direction	Intensity
LCNet	3.47	0.082	5.38	0.089	7.85	0.096	7.50	0.103
LCNet _{w/o mask}	5.46	0.104	8.85	0.144	11.81	0.176	13.02	0.166
LCNet _{local}	6.87	0.198	9.98	0.255	10.58	0.264	9.50	0.266

Table 4.2 Results of LCNet and LCNet_{reg} on SPHERE and BUNNY rendered under different lighting distributions.

Model	Near Uniform				Around Equator				Around Poles			
	SPHERE		BUNNY		SPHERE		BUNNY		SPHERE		BUNNY	
	Direction	Intensity	Direction	Intensity	Direction	Intensity	Direction	Intensity	Direction	Intensity	Direction	Intensity
LCNet	3.47	0.082	5.38	0.089	3.32	0.079	5.33	0.077	4.82	0.088	6.34	0.095
LCNet _{reg}	4.10	0.104	5.46	0.094	3.72	0.091	5.85	0.092	5.57	0.104	7.47	0.102

the lighting estimation results. This might be explained by that object mask helps the network distinguish the shadow region from the non-object region, while the proposed local-global feature fusion strategy can effectively make use of information from multiple observations.

We then compared LCNet with a regression based baseline, denoted as LCNet_{reg}, to validate the effectiveness of the classification based model. LCNet_{reg} shares the same architecture with LCNet, except that LCNet_{reg} estimates a 3-vector for light direction and a scalar value for light intensity, rather than the softmax probability vectors. Given q images, the loss function for the lighting regression is

$$\mathcal{L}_{\text{Reg}} = \lambda_l \frac{1}{q} \sum_i^q (1 - \mathbf{l}_i^\top \tilde{\mathbf{l}}_i) + \lambda_e \frac{1}{q} \sum_i^q (e_i - \tilde{e}_i)^2, \quad (4.4)$$

where λ_l and λ_e are the weighting factors for the loss terms, \mathbf{l}_i (e_i) and $\tilde{\mathbf{l}}_i$ (\tilde{e}_i) denote the predicted light direction (intensity) and the ground truth, respectively, for image i . During training, λ_l and λ_e are set to 1 (we found that using other weighting factors produce similar results).

Specifically, we tested LCNet and LCNet_{reg} on three different lighting distributions illustrated in Fig. 4.3 (a). The results are shown in Table 4.2. The proposed classi-

fication based LCNet consistently outperforms $\text{LCNet}_{\text{reg}}$ on both light direction and intensity estimation. This echoes our hypothesis that classifying a light direction to a certain range is easier than regressing an exact value. Thus, solving the classification problem reduces the learning difficulty and improves the performance. It can also be seen that both methods perform better on *Around Equator* and worse on *Around Poles*. This suggests that lightings around the poles are more difficult to estimate due to their extremely directions, independent of the lighting space discretization.

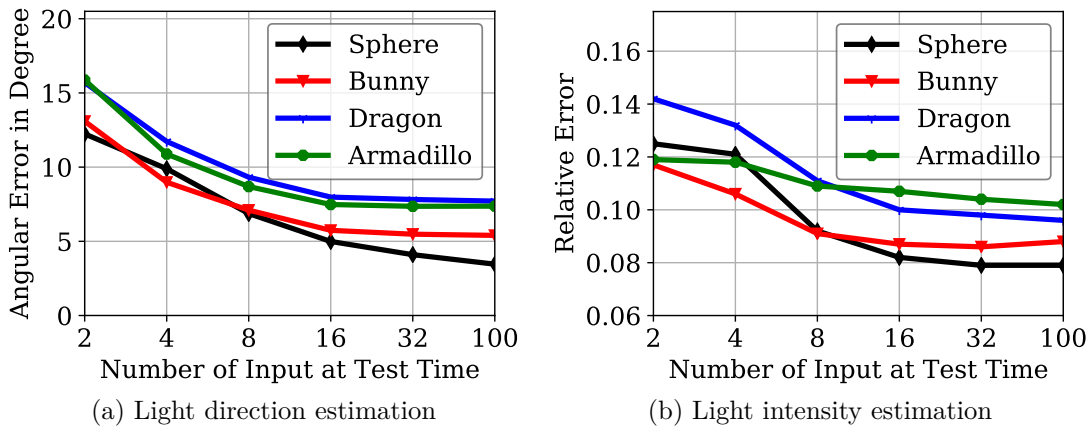


Fig. 4.4 Lighting estimation results of LCNet on SynTest^{MERL} dataset with varying input image numbers.

Figure 4.4 shows that the performance of LCNet increases with the number of input images. This is expected, since more useful information can be used to infer lightings with more input images.

Integration with PS-FCN for normal estimation For uncalibrated photometric stereo, we train a variant of PS-FCN, denoted as PS-FCN^\dagger , using the lighting estimated by LCNet. Note that the weights of LCNet was fixed during the training of PS-FCN^\dagger , as we found that end-to-end fine-tuning did not improve the performance. Experiments with IDs C1 & C2 in Table 4.3 show that after training with the discretized lighting estimated by LCNet, PS-FCN^\dagger performs better than PS-FCN given possibly noisy lightings at test time. Besides, experiments with IDs C2 & C3 show that PS-FCN^\dagger coupled with the classification based LCNet consistently outperforms that with the

Table 4.3 Normal estimation results on SynTest^{MERL} dataset. PS-FCN[†] was trained given lightings estimated by LCNet or LCNet_{reg}.

ID	Model	# Param	SPHERE	BUNNY	DRAGON	ARMAD.
C0	PS-FCN	2.2 M	2.66	3.80	4.83	5.24
C1	LCNet + PS-FCN	6.6 M	3.19	4.67	6.92	7.70
C2	LCNet + PS-FCN [†]	6.6 M	2.71	4.09	6.41	7.09
C3	LCNet _{reg} + PS-FCN [†]	6.6 M	3.22	4.99	6.63	7.54
C4	UPS-FCN _{deep+mask}	6.1 M	3.65	6.41	9.68	11.26
C5	UPS-FCN	2.2 M	7.44	12.34	14.44	15.93

regression based LCNet_{reg}.

Comparison with single-stage deep uncalibrated models To validate the effectiveness of the proposed two-stage approach, we compared our method with two different single-stage baseline models. We first train a variant of PS-FCN, denoted as UPS-FCN, without taking the light direction as input during training and testing. We then increased the model capacity of UPS-FCN by training a deeper network, denoted as UPS-FCN_{deep+mask}, that takes both the images and object mask as input (see Fig. 4.5).

Experiments with IDs C4 & C5 in Table 4.3 show that utilizing a deeper network and taking the object mask as input can improve the performance of single-stage model. However, experiments with IDs C2 & C4 show that the proposed method significantly outperforms the single-stage model, when the input as well as the number of parameters are comparable. This result indicates that simply increasing the depth of the network cannot produce optimal results.

4.4 Analyzing What is Learned in LCNet

Section 4.3.5 shows that the proposed LCNet can predict accurate lightings for different testing objects. Also, given the lightings estimated by LCNet, the calibrated method PS-FCN significantly outperforms the single-stage uncalibrated method UPS-FCN. However, what specifically inside the LCNet contributes to its success remains

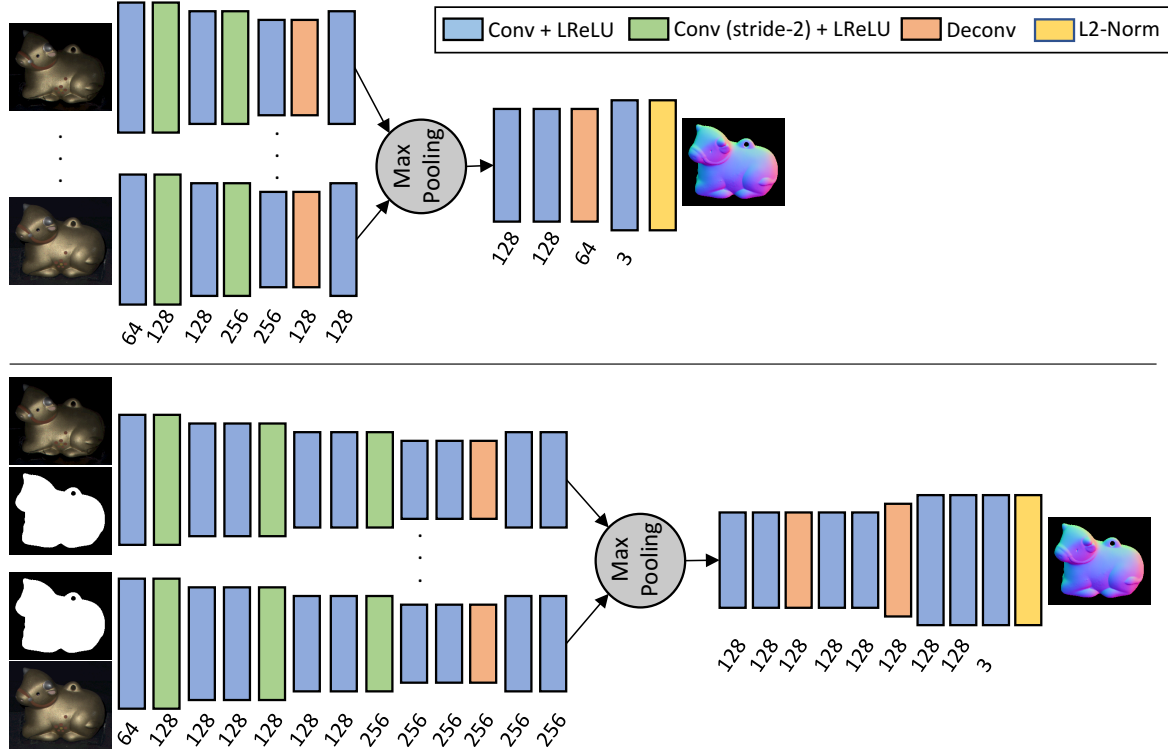


Fig. 4.5 Network architectures of UPS-FCN (top) and UPS-FCN_{deep+mask} (bottom).

a mystery.

In this section, we discuss the inherent ambiguity in uncalibrated photometric stereo of Lambertian surfaces, the fact that it can be resolved for non-Lambertian surfaces, and the features learned by LCNet to resolve such ambiguity.

4.4.1 Lambertian Surfaces and the GBR Ambiguity

When ignoring shadows (*i.e.*, attached and cast shadows) and inter-reflections, the image formation of a Lambertian surface with P pixels captured under F lightings can be written as

$$\mathbf{M} = \mathbf{N}^\top \mathbf{L}, \quad (4.5)$$

where $\mathbf{M} \in \mathbb{R}^{P \times F}$ is the measurement matrix. $\mathbf{N} \in \mathbb{R}^{3 \times P}$ is the surface normal matrix whose columns are albedo scaled normals $\mathbf{N}_{:,p} = \rho_p \mathbf{n}_p$, where ρ_p and \mathbf{n}_p are the albedo and unit-length surface normal of pixel p . $\mathbf{L} \in \mathbb{R}^{3 \times F}$ is the lighting matrix whose

columns are intensity scaled light directions $\mathbf{L}_{:,f} = e_f \mathbf{l}_f$, where e_f and \mathbf{l}_f are the light intensity and unit-length light direction of image f .

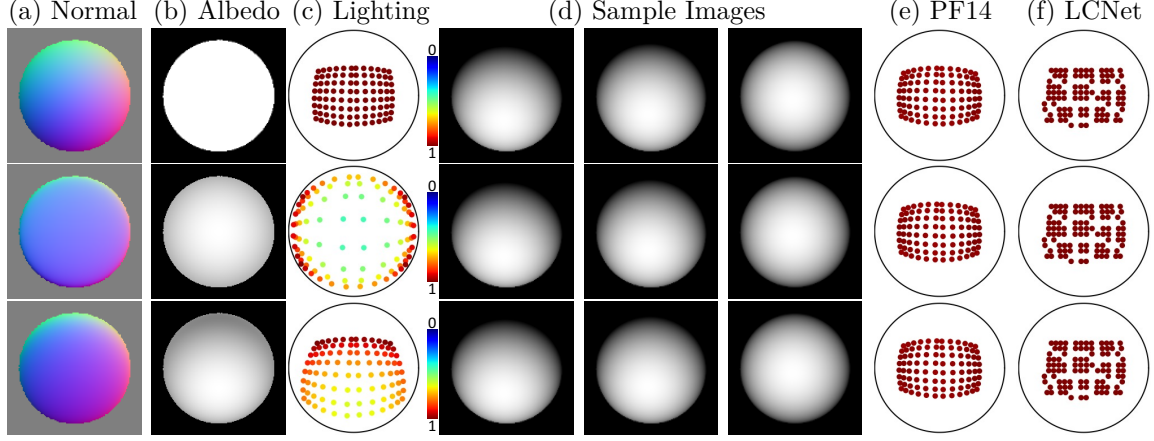


Fig. 4.6 Row 1 is the true shape of a SPHERE, while rows 2 and 3 are shapes under two different GBR transformations. In column (c), the points' positions and colors indicate light direction and relative intensity, respectively. Columns (e) and (f) show the lightings estimated by PF14 [88] and LCNet.





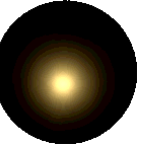
By matrix factorization and applying the surface integrability constraint, \mathbf{N} and \mathbf{L} can be recovered up to an unknown 3-parameter GBR transformation \mathbf{G} [90] such that $\mathbf{M} = (\mathbf{G}^{-\top} \mathbf{N})^\top (\mathbf{G} \mathbf{L})$. This GBR ambiguity indicates that there are infinitely many combinations of albedo ρ , normal \mathbf{n} , light direction \mathbf{l} , and light intensity e that produce the same appearance \mathbf{M} (see Fig. 4.6 (a)-(d)):

$$\hat{\rho} = \rho |\mathbf{G}^{-\top} \mathbf{n}|, \quad \hat{\mathbf{n}} = \frac{\mathbf{G}^{-\top} \mathbf{n}}{|\mathbf{G}^{-\top} \mathbf{n}|}, \quad \hat{\mathbf{l}} = \frac{\mathbf{G} \mathbf{l}}{|\mathbf{G} \mathbf{l}|}, \quad \hat{e} = e |\mathbf{G} \mathbf{l}|. \quad (4.6)$$

Although the surface's appearance remains the same after GBR transformation (*i.e.*, $\hat{\rho} \hat{\mathbf{n}}^\top \hat{\mathbf{l}} \hat{e} = \rho \mathbf{n}^\top \mathbf{l} e$, see Fig. 4.6 (d)), a surface point's albedo will be scaled by $|\mathbf{G}^{-\top} \mathbf{n}|$. As a result, the albedo of an object will change gradually and become spatially-varying. Because this kind of spatially-varying albedo distribution resulting from GBR transformations rarely occurs on real world objects, some previous methods make explicit assumptions on the albedo distribution (*e.g.*, constant albedo [88, 90] or low entropy [85]) to resolve the ambiguity.

PF14 [88], a state-of-the-art non-learning uncalibrated method [59], detects Lam-

Table 4.4 Light direction estimation results of PF14 [88] and LCNet on a SPHERE rendered with different BRDF types. Non-Lambertian BRDFs are taken from the MERL dataset [58].

						
Model	Lambertian	Fabric	Plastic	Phenolic	Metallic	Avg.
PF14	7.19	14.26	28.04	47.96	31.12	25.7
LCNet	5.38	4.07	3.08	3.05	4.09	3.93

bertian diffuse reflectance maxima (*i.e.*, image points satisfying $\mathbf{n}^\top \mathbf{l} = 1$) to estimate \mathbf{G} 's 3 parameters. We will later use it as a non-learning benchmark in our comparative experiments.

4.4.2 LCNet and the GBR Ambiguity

Figure 4.6 (e)-(f) compare the results of LCNet and PF14 on surfaces that differ by GBR transformations. Since the input images are the same in all cases, LCNet estimates the same lightings in all cases, namely the most likely lightings for the input images. The same also applies to PF14. Although LCNet's result does not exactly equal the lightings for uniform albedo, we note that it learned from the training data that GBR-transformed surfaces are unlikely.

Although uncalibrated photometric stereo has an intrinsic GBR ambiguity for Lambertian surfaces, it was shown that GBR transformations do not preserve specularities [68, 90, 94]. Hence, specularities are helpful for ambiguity-free lighting estimation. However, traditional methods often treat non-Lambertian observations as outliers, and thus fail to make full use of specularities for disambiguation [88]. In contrast, learning-based methods can learn the relation between specular highlights and light directions through end-to-end learning. As shown in Table 4.4, LCNet achieves good results for non-Lambertian surfaces while PF14 completely fails when non-Lambertian observations dominate.

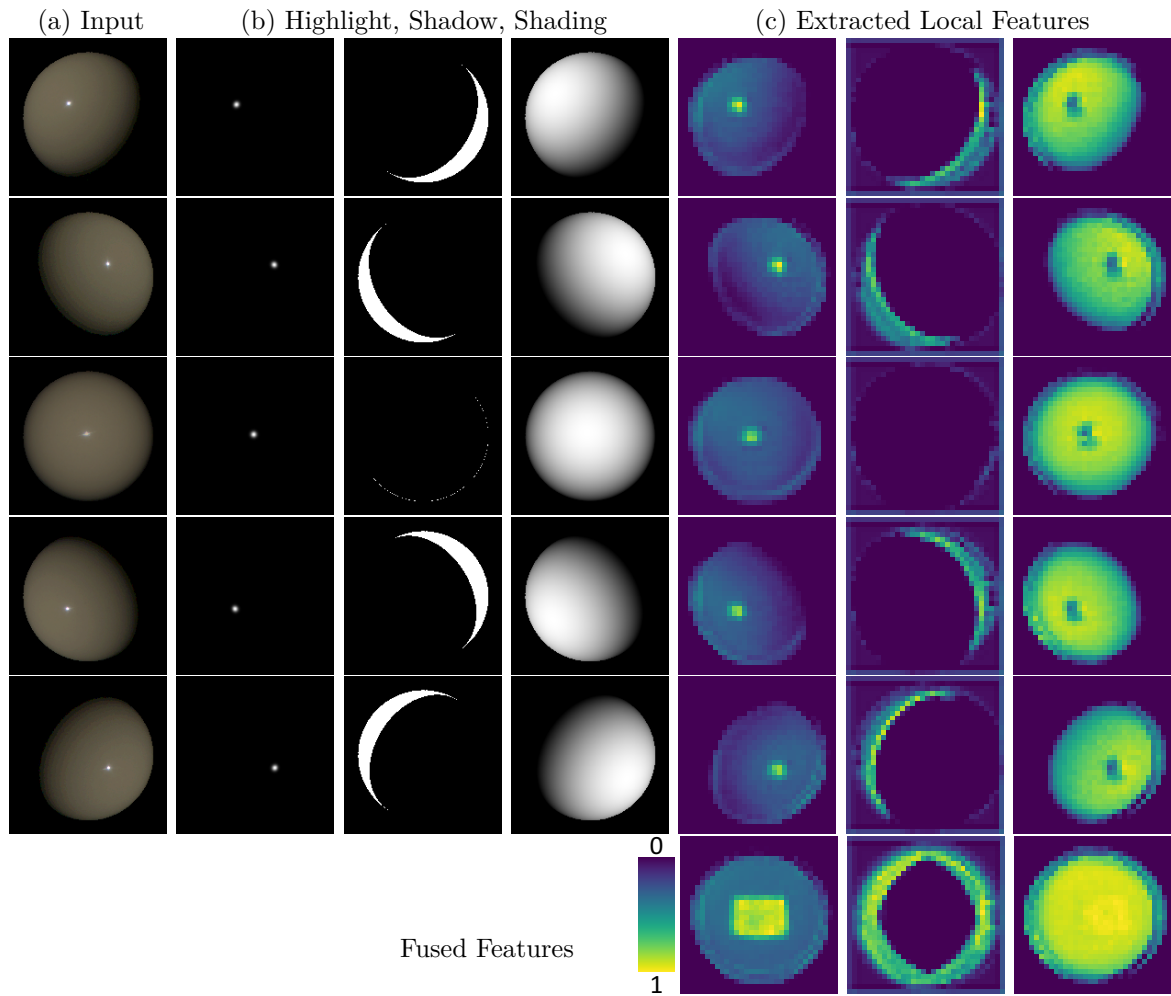


Fig. 4.7 Feature visualization of LCNet on a non-Lambertian SPHERE. *Column 1*: 5 of the 96 input images; *Columns 2–4*: Specular highlight centers, attached shadows, and shading rendered from ground truth; *Columns 5–7*: 3 of LCNet’s 256 features maps. The last row shows the global features produced by fusing local features with max-pooling. All features are normalized to $[0, 1]$ and color coded.

Table 4.5 Light direction estimation results of LCNet trained with different inputs. Values indicate mean angular error in degree.

model input	SPHERE	BUNNY	DRAGON	ARMADILLO
images	3.03	4.88	6.30	6.37
(a) attached shadows	3.50	5.07	9.78	5.22
(b) specular component	2.53	6.18	7.33	4.08
(c) shading	2.29	3.95	4.64	3.76
(a) + (b) + (c)	1.87	2.06	2.34	2.12

4.4.3 Feature Analysis for LCNet

To analyze the features learned by LCNet, we first visualize the learned local and global features. Figure 4.7 shows 3 representative features selected from 256 feature maps extracted by LCNet from images of a non-Lambertian SPHERE dataset. Comparing Fig. 4.7’s Column 2 with Column 5, Column 3 with Column 6, and Column 4 with Column 7, we can see that some feature maps are highly correlated with attached shadows (regions where the angle $\angle(\mathbf{n}, \mathbf{l}) \geq 90^\circ$), shadings ($\mathbf{n}^\top \mathbf{l}$), and specular highlights (regions where \mathbf{n} is close to the half angle $\mathbf{h} = \frac{\mathbf{l} + \mathbf{v}}{|\mathbf{l} + \mathbf{v}|}$ of \mathbf{l} and viewing direction \mathbf{v}). As discussed earlier in the related work (Section 4.2), these provide strong clues for resolving the ambiguity.

To further verify our observations, we did the following. We computed (a) attached shadows, (b) the “specular components” (with a bit of concept abuse, we denote $\mathbf{h}^\top \mathbf{n}$ as specular component), and (c) shadings for the synthetic Blobby and Sculpture datasets from their ground-truth light directions and normals. We then trained 4 variants of the LCNet, taking (a), (b), (c), and (a) + (b) + (c), respectively, as input instead of regular images. We compared these 4 variant networks with LCNet (*i.e.*, the network trained with Blobby and Sculpture images) on a synthetic test dataset introduced in Section 4.6.1. Similar to LCNet, the variant networks also took the object mask as input. Table 4.5 shows that the variant models achieve results comparable to or even better than the model trained on regular images.

We can see that shadings contribute more than attached shadows and specular

components for lighting estimation. This is because shading information actually includes attached shadows (*i.e.*, pixels with a zero value in the shading for synthetic data), and can be considered as an image of an object with uniform albedo (albedo equals to 1). The uniform albedo constraint is a well-known clue for resolving the GBR ambiguity [88, 90]. In practice, attached shadows, shadings, and the specular components are not directly available as input, but this confirms our assumption that they provide strong clues for accurate lighting estimation.

As discussed before, LCNet learns to resolve ambiguity by assuming that a surface with a gradually changing albedo corresponding to GBR transformations rarely exists. However, we have not observed features apparently related to albedo distribution. We hypothesize that the albedo distribution prior is implicitly employed to restrict the mapping space, since LCNet learns the mapping from extracted features to lightings.

4.5 Guided Calibration Network (GCNet)

We have analyzed the features learned by LCNet and discussed how it resolves the ambiguity. In this section, we present the motivations for our guided calibration network (GCNet) and detail its structure.

4.5.1 Guided Feature Extraction

As we have seen, features like attached shadows, shadings, and specularities are important for lighting estimation, and a lighting estimation network may benefit greatly from being able to estimate them accurately. We further know that these features are completely determined by the light direction for each image as well as the inter-image shape information derived from the surface normal map. However, LCNet extracts features independently from each input image and thus cannot exploit any inter-image information during feature extraction. This observation also indicates that simply increasing the layer number of LCNet’s shared-weight feature extractor cannot produce significant improvement.

Surface normal as inter-image guidance Intuitively, if we can provide such inter-image shape information as input to the network to guide the feature extraction process, it should be able to perform better. This, however, constitutes a chicken-and-egg problem where we require normals and lightings for accurate feature extraction but at the same time we require these features for estimating accurate lightings. We therefore suggest a cyclic network structure in which we first estimate initial lightings, and then use them to estimate normals as inter-image information to guide the extraction of local (*i.e.*, per-image) features to ultimately estimate final lightings. An alternative idea might be directly estimating surface normals from images. However, we have shown in Table 4.5 that surface normals estimated directly from images are not accurate.

Shading as intra-image guidance Another advantage of first estimating initial lighting and surface normals is that we can easily compute coarse attached shadows, shadings, or specular components as intra-image guidance for the feature extraction process (intra-image means the information is different for each image). As shading information already includes attached shadows, and not all materials exhibit specular highlights, we only compute the shading for each image as the dot-product of the estimated lighting with the surface normals, and use it as intra-image guidance. We experimentally verified that additionally including the specular component as network input does not improve results. The computed shading, on the other hand, does improve results and can assist the network to extract better features.

4.5.2 Network Architecture

As shown in Fig. 4.8, the proposed GCNet consists of two lighting estimation sub-networks (L-Net) and a normal estimation sub-network (N-Net). The first L-Net, “L-Net₁”, estimates initial lightings given the input images and object masks. The N-Net then estimates surface normals from the lightings estimated by L-Net₁ and the input images. Finally, the second L-Net, “L-Net₂”, estimates more accurate lightings

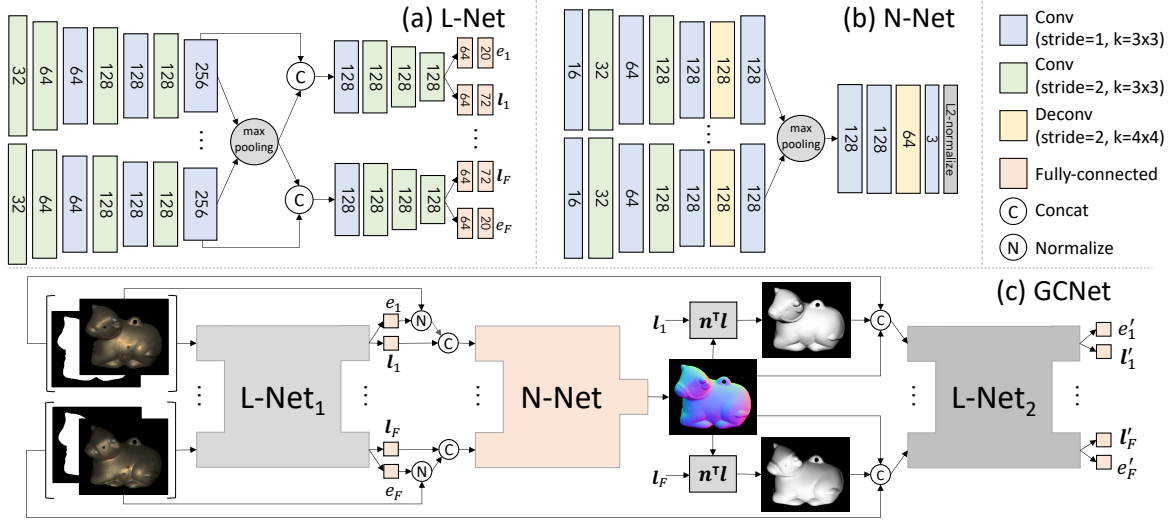


Fig. 4.8 (a) Structure of the lighting estimation sub-network L-Net. (b) Structure of the normal estimation sub-network N-Net. (c) The entire GCNet. Values in layers indicate the output channel number.

based on the input images, object masks, the estimated normals, and the computed shadings.

L-Net The L-Net is designed based on LCNet but has less channels in the convolutional layers to reduce the model size (see Fig. 4.8 (a)). Compared to LCNet’s 4.4 million parameters, each L-Net has only 1.78 million parameters.

Following LCNet, we discretize the lighting space and treat lighting estimation as a classification problem. Specifically, L-Net’s output light direction and intensity are in the form of softmax probability vectors (a 32-vector for elevation, a 32-vector for azimuth, and a 20-vector for intensity). Given F images, the loss function for L-Net is

$$\mathcal{L}_{\text{light}} = \frac{1}{F} \sum_f (\mathcal{L}_{l_a}^f + \mathcal{L}_{l_e}^f + \mathcal{L}_e^f), \quad (4.7)$$

where $\mathcal{L}_{l_a}^f, \mathcal{L}_{l_e}^f$, and \mathcal{L}_e^f are the cross-entropy loss for light azimuth, elevation, and intensity classifications for the f^{th} input image, respectively. The output probability vectors can be converted to a 3-vector light direction \mathbf{l}_f and a scalar light intensity e_f

by taking the probability vector’s expectation, which is differentiable for later end-to-end fine-tuning.

L-Net₁ and L-Net₂ differ in that L-Net₁ has 4 input channels (3 for the image, 1 for the object mask) while L-Net₂ has 8 (3 additional channels for normals, 1 for shading).

N-Net The N-Net is designed based on PS-FCN [17] but with less channels, resulting in 1.1 million parameters compared to PS-FCN’s 2.2 million parameters (see Fig. 4.8 (b) for details). Following PS-FCN, the N-Net’s loss function is

$$\mathcal{L}_{\text{normal}} = \frac{1}{P} \sum_p (1 - \mathbf{n}_p^\top \tilde{\mathbf{n}}_p), \quad (4.8)$$

where P is the number of pixels per image, and \mathbf{n}_p and $\tilde{\mathbf{n}}_p$ are the predicted and the ground-truth normal at pixel p , respectively.

End-to-end fine-tuning We train L-Net₁, N-Net, and L-Net₂ one after another until convergence and then fine-tune the entire network end-to-end using the following loss

$$\mathcal{L}_{\text{fine-tune}} = \mathcal{L}_{\text{light}_1} + \mathcal{L}_{\text{normal}} + \mathcal{L}_{\text{shading}} + \mathcal{L}_{\text{light}_2}, \quad (4.9)$$

$$\mathcal{L}_{\text{shading}} = \frac{1}{FP} \sum_f \sum_p (\mathbf{n}_p^\top \mathbf{l}_f - \tilde{\mathbf{n}}_p^\top \tilde{\mathbf{l}}_f)^2, \quad (4.10)$$

where $\mathcal{L}_{\text{light}_1}$ and $\mathcal{L}_{\text{light}_2}$ denote the lighting estimation loss for L-Net₁ and L-Net₂. The shading loss term $\mathcal{L}_{\text{shading}}$ is included to encourage better shading estimation, and \mathbf{l}_f and $\tilde{\mathbf{l}}_f$ denote the light direction predicted by L-Net₁ and ground-truth light direction for the f^{th} image, respectively.

Training details Following LCNet, we trained the networks on the synthetic Blobby and Sculpture Dataset which contains 85,212 surfaces, each rendered under 64 random light directions.

First, we train L-Net₁ from scratch for 20 epochs, halving the learning rate every 5 epochs. Second, we train N-Net using ground-truth lightings and input images following PS-FCN’s training procedure (see Section 3.6), and then retrain N-Net given the lightings estimated by L-Net₁ for 5 epochs, halving the learning rate every 2 epochs. Third, we train L-Net₂ given the input images, object masks, estimated normals, and computed shadings for 20 epochs, halving the learning rate every 5 epochs. The initial learning rate is 0.0005 for L-Net₁ and L-Net₂, and 0.0002 for retraining N-Net. End-to-end training is done for 20 epochs with an initial learning rate of 0.0001, halving it every 5 epochs.

We implemented our framework in PyTorch [83] and used the Adam optimizer [42] with default parameters. The full network has a total of 4.66 million parameters which is comparable to LCNet (4.4 million). The batch size and the input image number for each object are fixed to 32 during training. The input images for all sub-networks are resized to 128×128 at both training and test time.

4.6 Experimental Results

We first compare LCNet and GCNet on the synthetic test dataset SynTest^{MERL} introduced in Section 3.5.2, and then compared our methods with existing methods on the publicly available real datasets. Similarly, the widely used mean angular error (MAE) in degree is adopted to measure the accuracy of the predicted light directions and surface normals. The scale-invariant relative error (RE) is used to measure the accuracy of the predicted light intensities (see Section 4.3.5 for definition).

For all experiments on synthetic dataset involving input with unknown light intensities, we randomly generated light intensities in the range of $[0.2, 2.0]$. Each experiment was repeated five times and the average results were reported.

Table 4.6 Ablation study for network architecture of GCNet. Lighting estimation results of GCNet on SynTest^{MERL} dataset. The results are averaged over 100 MERL BRDFs (bold fonts indicates the best).

ID	Model	SPHERE		BUNNY		DRAGON		ARMADILLO	
		Direction	Intensity	Direction	Intensity	Direction	Intensity	Direction	Intensity
0	LCNet	3.03	0.064	4.88	0.066	6.30	0.072	6.37	0.074
1	L-Net ₁ + N-Net + L-Net ₂ + Finetune	2.21	0.042	2.44	0.046	3.88	0.055	3.52	0.060
2	L-Net ₁ + N-Net + L-Net ₂	2.52	0.052	2.90	0.054	4.20	0.061	3.92	0.060
3	L-Net ₁ + N-Net + L-Net ₂ ^(w/o normal)	2.45	0.050	3.35	0.051	5.82	0.070	5.25	0.059
4	L-Net ₁	3.20	0.053	4.47	0.060	5.80	0.081	5.71	0.079
5	L-Net ₁ + N-Net + L-Net ₂	2.52	0.052	2.90	0.054	4.20	0.061	3.92	0.060
6	L-Net ₁ + N-Net + L-Net ₂ ^(w/o shading; w/ light)	2.83	0.047	3.50	0.051	5.36	0.075	4.04	0.068
7	L-Net ₁ + N-Net + L-Net ₂ ^(w/o shading)	2.79	0.046	3.21	0.056	4.63	0.072	4.29	0.062
8	L-Net ₁ + L-Net ₂	2.92	0.051	4.37	0.058	5.99	0.079	5.31	0.077

Table 4.7 Normal estimation results on SynTest^{MERL} dataset. The estimated normals are predicted by PS-FCN [17] given the lightings estimated LCNet and GCNet.

Model	SPHERE	BUNNY	DRAGON	ARMADILLO
LCNet + PS-FCN	2.98	4.06	5.59	6.73
GCNet + PS-FCN	2.93	3.68	4.85	5.01

4.6.1 Evaluation on Synthetic Data

Ablation study To validate the design of the proposed GCNet, we performed an ablation study and summarized the results in Table 4.6. The comparison between experiments with IDs 2-4 verifies that taking both the estimated normals and shading as input is beneficial for lighting estimation. The comparison between experiments with IDs 1 & 2 demonstrates that end-to-end fine-tuning further improves the performance. We can also see that L-Net₁ achieves results comparable to LCNet despite using only half of the network parameters, which indicates that simply increasing the channel number of the convolutional layers cannot guarantee better feature extraction. In the rest of this chapter, we denote the results of “L-Net₁ + N-Net + L-Net₂ + Finetune” as “GCNet”.

Table 4.7 shows that, as expected, the calibrated photometric stereo method PS-FCN (trained with ground-truth lightings) can estimate more accurate normals given better estimated lighting.

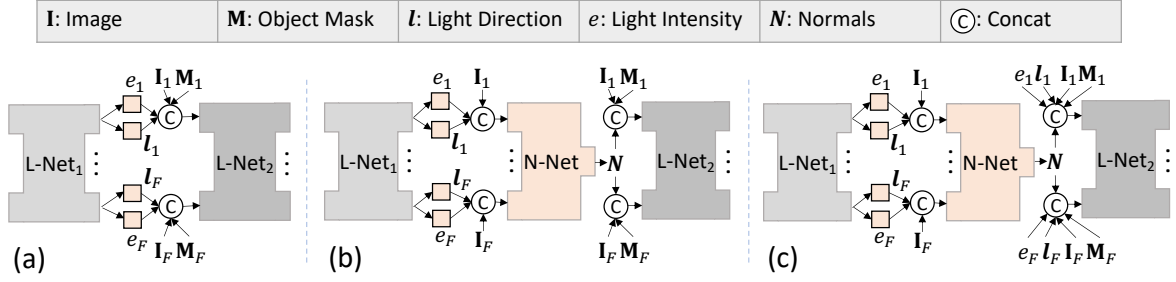



Fig. 4.9 Three different cascaded structures. (a) L-Net₁ + L-Net₂. (b) L-Net₁ + N-Net + L-Net₂^(w/o shading). (c) L-Net₁ + N-Net + L-Net₂^(w/o shading; w/ light). We skip the input of L-Net₁ and the output of L-Net₂ for all models to simplify the illustration.

Comparison of different cascaded structures Cascaded structure is a popular strategy to improve performance. We compared the proposed structure for GCNet with three different structures (see Fig. 4.9) to verify our method’s effectiveness.

Figure 4.9 (a) is a common structure to refine a network’s estimation using another similar network. As discussed in Section 4.5.1, L-Net’s bottleneck is the lack of inter-image information (*e.g.*, normals) during feature extraction, making this structure sub-optimal (compare the experiments with IDs 5 & 8 in Table 4.6). Figure 4.9 (b) is a sequential structure where L-Net₂ additionally takes estimated normals as input. However, the experiments with the IDs 5 & 7 show that taking the estimated shading (intra-image information) as input is beneficial. In Fig. 4.9 (c), L-Net₂ takes estimated normals and lightings as input. Our experiment shows that taking lightings as input can lead to faster convergence, but the final performance is worse than the proposed method, as show in the experiments with the IDs 5 & 6. We suspect that L-Net₂ becomes more dependent on input lightings if directly taking them as input during training. When L-Net₁’s estimated lightings are not accurate, the refined estimation may not be good.


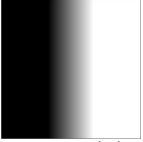


Results on different lighting distributions To analyze the effect of biased lighting distributions on the proposed method, we evaluated GCNet on the ARMADILLO illuminated under three different lighting distributions: a near uniform, a narrow, and an upward-biased distribution. Table 4.8 shows that both GCNet and LCNet have

Table 4.8 Results on ARMADILLO under three different lighting distributions.

	Near Uniform			Narrow			Upward-biased		
	Direction	Intensity	Normal	Direction	Intensity	Normal	Direction	Intensity	Normal
LCNet + PS-FCN	6.09	0.072	6.49	5.92	0.059	8.44	7.10	0.065	8.80
GCNet + PS-FCN	3.39	0.059	4.90	4.29	0.048	6.82	5.96	0.054	7.53

decreased performance under biased lighting distributions (*e.g.*, the upward-biased distribution), but GCNet consistently outperforms LCNet.




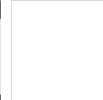
Table 4.9 Lighting estimation results on BUNNY rendered with SVBRDFs. (a) BUNNY with uniform BRDF. (b) and (c) show the “Ramp” and “Irregular” material maps and two sample images of BUNNY with the corresponding SVBRDFs.

							
(a) Uniform		(b) Ramp		(c) Irregular			
Model	Uniform		Ramp		Irregular		
	Direction	Intensity	Direction	Intensity	Direction	Intensity	
LCNet	4.88	0.066	6.09	0.066	6.00	0.075	
GCNet	2.44	0.046	4.16	0.043	3.68	0.050	

Results on surfaces with SVBRDFs To analyze the effect of SVBRDFs, we used two different material maps to generate a synthetic dataset of surfaces with SVBRDFs following Goldman *et al.* [82]. Specifically, we rendered 100 test objects by randomly sampling two MERL BRDFs and blended the BRDFs for BUNNY using “Ramp” and “Irregular” material maps shown in Table 4.9 (b) and (c). Table 4.9 shows that although both methods perform worse on surfaces with SVBRDFs compared to uniform BRDFs, our method is still reasonably good even though it was trained on surfaces with uniform BRDFs. This might be explained by that although SVBRDFs may affect the feature extraction of some important clues such as shading, others such as attached shadows and specular highlights are less affected and can still be extracted

to estimate reliable lightings.

Table 4.10 Lighting estimation results on surface regions cropped from BUNNY.

Object Mask												
					<i>alumina-oxide</i>				<i>beige-fabric</i>			
	LCNet		GCNet		LCNet		GCNet					
	dir.	int.	dir.	int.	dir.	int.	dir.	int.				
Surface Normal	(a)	4.29	0.054	1.35	0.025	4.54	0.051	2.29	0.026			
	(b)	3.83	0.050	1.71	0.023	4.45	0.044	2.00	0.029			
	(c)	3.75	0.042	2.46	0.024	4.97	0.044	3.13	0.025			
	(d)	4.04	0.047	2.84	0.026	4.55	0.051	3.46	0.025			

Effect of the object silhouette Object silhouette can provide useful information for lighting calibration (*e.g.*, normals at the occluding contour are perpendicular to the viewing direction). To investigate the effect of the silhouette, we first rendered the BUNNY using two different types of BRDFs (*alumina-oxide* and *beige-fabric*) under 100 lightings sampled randomly from the upper hemisphere, and then cropped surface regions with different sizes for testing. Table 4.10 shows that both LCNet and our method perform robustly for surface regions with or without silhouette, while our method consistently outperforms LCNet. This is because the training data for both methods was generated by randomly cropping image patches from the Blobby and Sculpture datasets, which contains surface regions without silhouette.

4.6.2 Evaluation on Real Data

Table 4.11 Lighting estimation results on DiLiGenT benchmark. Bold font indicates the best result.

















Method											Average
Light Direction Estimation											
PF14 [88]	4.90	5.31	2.43	5.24	13.52	9.76	33.22	21.77	16.34	24.99	13.75
LCNet	3.27	4.08	5.44	3.47	2.87	4.34	10.36	4.50	4.52	6.32	4.92
GCNet	1.75	4.58	1.41	2.44	2.81	2.86	2.98	5.47	3.15	5.74	3.32
Light Intensity Estimation											
PF14 [88]	0.017	0.098	0.044	0.053	0.223	0.122	0.074	0.156	0.088	0.059	0.036
LCNet	0.039	0.095	0.058	0.061	0.048	0.048	0.067	0.105	0.073	0.082	0.068
GCNet	0.027	0.075	0.039	0.101	0.059	0.032	0.042	0.048	0.031	0.065	0.052

Table 4.12 Lighting estimation results on Light Stage Data Gallery.

model	 HELMET SIDE		 PLANT		 FIGHTING KNIGHT		 KNEELING KNIGHT		 STANDING KNIGHT		 HELMET FRONT		Average	
	Dir.	Int.	Dir.	Int.	Dir.	Int.	Dir.	Int.	Dir.	Int.	Dir.	Int.	Dir.	Int.
PF14 [88]	25.40	0.576	20.56	0.227	69.50	1.137	46.69	9.805	33.81	1.311	81.60	0.133	46.26	2.198
LCNet	6.57	0.212	16.06	0.170	15.95	0.214	19.84	0.199	11.60	0.286	11.62	0.248	13.61	0.221
GCNet	5.33	0.096	10.49	0.154	13.42	0.168	14.41	0.181	5.31	0.198	6.22	0.183	9.20	0.163

To demonstrate the proposed method’s capability to handle real-world non-Lambertian objects, we evaluated our method on the challenging *DiLiGenT benchmark* [59] and the *Light Stage Data Gallery* [61].

Results on lighting estimation We first compared our GCNet with the LCNet and non-learning method PF14 [88]. PF14 is the state-of-the-art traditional method for uncalibrated photometric stereo, and we used its publicly available code for testing. Table 4.11 shows that GCNet achieves the best average results on the DiLiGenT benchmark with an MAE of 3.32 for light directions and a relative error of 0.052 for light intensities. Although our GCNet does not achieve the best results for all objects, it exhibits the most robust performance with a maximum MAE of 5.77 and a maximum relative error of 0.101 compared with LCNet (MAE: 10.36, relative error: 0.105) and PF14 (MAE: 33.22, relative error: 0.223). Figures 4.10 (a)-(b) visualize the lighting estimation results for the POT1 and the GOBLET. The non-learning method PF14 works well for near-diffuse surfaces (*e.g.*, POT1), but quickly degenerates on highly specular surfaces (*e.g.*, GOBLET). Compared with LCNet, GCNet is more robust to surfaces with different reflectances and shapes.

Table 4.12 shows lighting estimation results on the Light Stage Data Gallery. Our GCNet significantly outperforms LCNet and PF14, and achieves an average MAE of 9.20 for light directions and a relative error of 0.163 for light intensities, improving the results of LCNet by 32.4% and 26.4% for light directions and light intensities respectively. Figures 4.10 (c)-(d) visualize lighting estimation results for the Light Stage Data Gallery’s STANDING KNIGHT and PLANT.

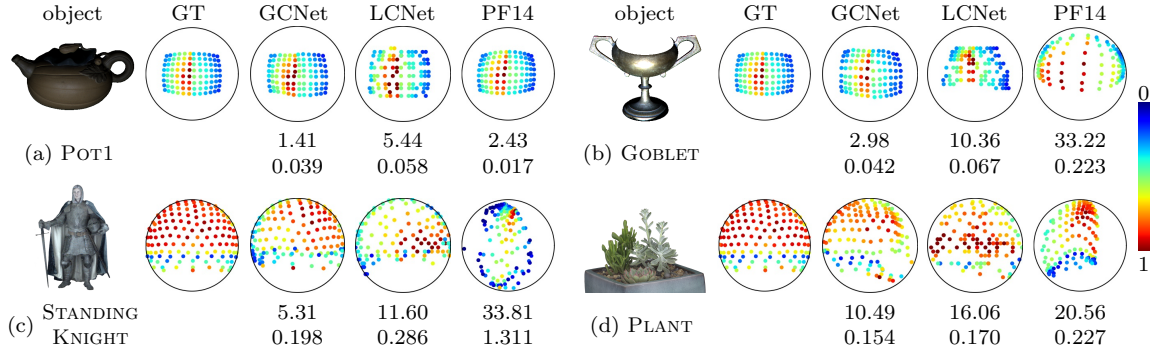


Fig. 4.10 Visualization of the ground-truth and estimated lighting distributions for the DiLiGenT benchmark and Light Stage Data Gallery.

Table 4.13 Normal estimation results on DiLiGenT benchmark.

Model	BALL	CAT	POT1	BEAR	POT2	BUDDHA	GOBLET	READING	COW	HARVEST	Average
AM07 [85]	7.3	31.5	18.4	16.8	49.2	32.8	46.5	53.7	54.7	61.7	37.3
SM10 [86]	8.9	19.8	16.7	12.0	50.7	15.5	48.8	26.9	22.7	73.9	29.6
WT13 [87]	4.4	36.6	9.4	6.4	14.5	13.2	20.6	59.0	19.8	55.5	23.9
LM13 [80]	22.4	25.0	32.8	15.4	20.6	25.8	29.2	48.2	22.5	34.5	27.6
PF14 [88]	4.8	9.5	9.5	9.1	15.9	14.9	29.9	24.2	19.5	29.2	16.7
LC18 [89]	9.3	12.6	12.4	10.9	15.7	19.0	18.3	22.3	15.0	28.0	16.3
LCNet + ST14 [69]	4.1	8.2	8.8	8.4	9.7	11.6	13.5	15.2	13.4	27.7	12.1
GCNet + ST14	2.0	7.7	7.5	5.7	9.3	10.9	10.0	14.8	13.5	26.9	10.8
LCNet + PS-FCN	3.2	7.6	8.4	11.4	7.0	8.3	11.6	14.6	7.8	17.5	9.7
GCNet + PS-FCN	2.5	7.9	7.2	5.6	7.1	8.6	9.6	14.9	7.8	16.2	8.7
LCNet + IS18 [53]	6.4	15.6	10.6	8.5	12.2	13.9	18.5	23.8	29.3	25.7	16.5
GCNet + IS18	3.1	6.9	7.3	5.7	7.1	8.9	7.0	15.9	8.8	15.6	8.6

Results on surface normal estimation We then verified that the proposed GCNet can be seamlessly integrated with existing calibrated methods to handle uncalibrated photometric stereo. Specifically, we integrated the GCNet with a state-of-the-art non-learning calibrated method ST14 [69] and two learning-based methods PS-FCN and IS18 [53]. Table 4.13 shows that these integrations can outperform existing state-of-the-art uncalibrated methods [80, 85–89] by a large margin on the DiLiGenT benchmark. We can further see that ST14, PS-FCN, as well as IS18 perform better with GCNet’s instead of LCNet’s predicted lightings: 10.8 vs. 12.1 for PS14, 8.7 vs. 9.7 for PS-FCN, and 8.6 vs. 16.5 for IS18. Figure 4.11 presents visual comparisons on the POT1 and GOBLET from the DiLiGenT benchmark.

Figure 4.12 shows the surface normals of the STANDING KNIGHT predicted by

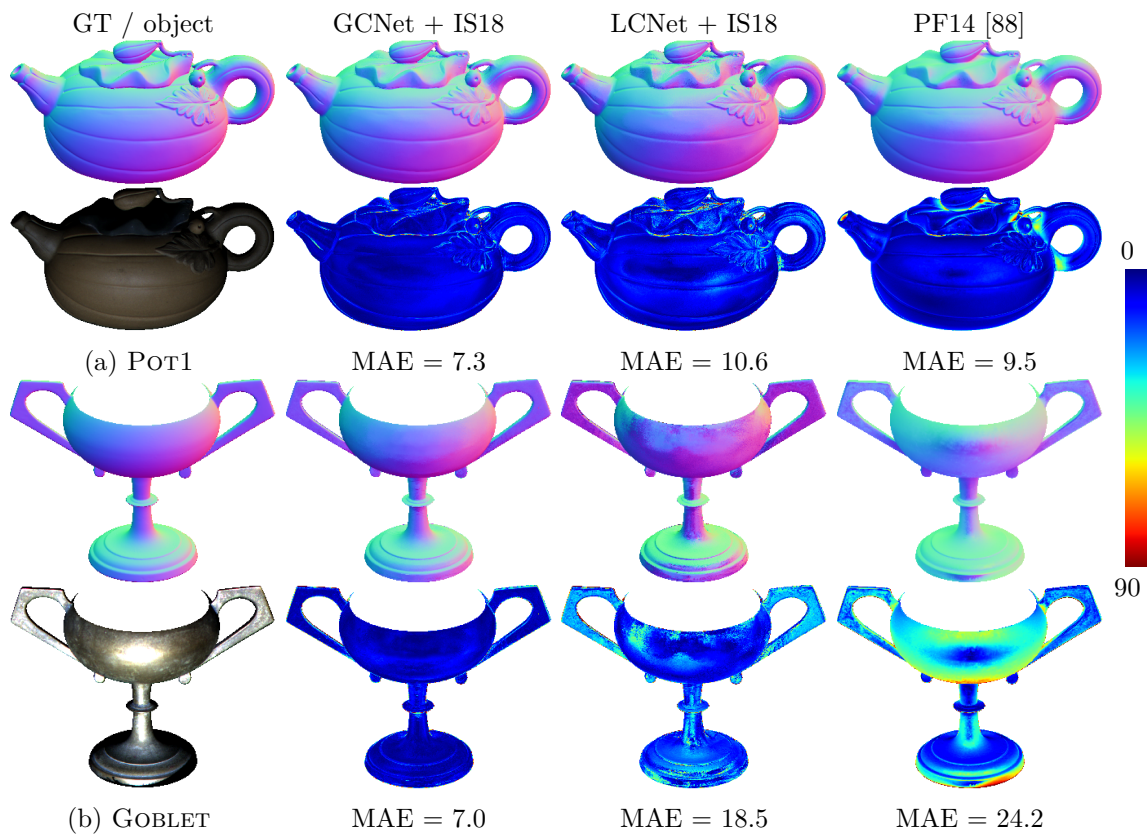


Fig. 4.11 Visual comparisons of normal estimation for GOBLET in the DiLiGenT benchmark. We compared the normal estimation results of a calibrated method IS18 [53] given lightings estimated by GCNet and LCNet.

PS-FCN given lighting estimated by GCNet and LCNet. We can see that coupled with GCNet’s more accurate lightings, PS-FCN can produce more reliable normal estimation for thin regions.

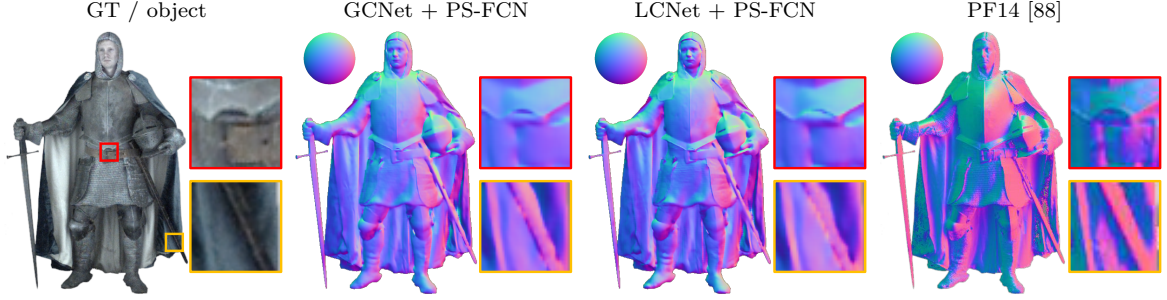
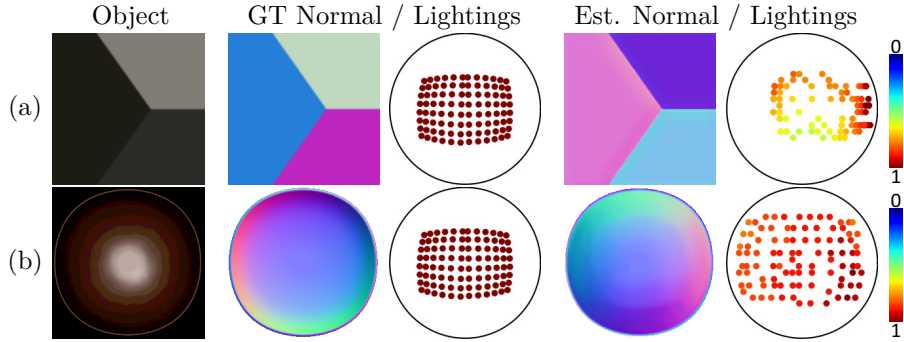


Fig. 4.12 Visual comparison of normal estimation for the Light Stage Data Gallery’s STANDING KNIGHT.

4.6.3 Failure Cases

Fig. 4.13 Failure cases. (a) Results on a piecewise planar surface with sparse normal distribution. (b) Results on a highly-concave bowl. The estimated normals are predicted by PS-FCN given GCNet’s estimated lightings.



As discussed in Section 4.4, LCNet relies on features like attached shadows, shading, and specular highlights, which is also true for GCNet. For piecewise planar surfaces with a sparse normal distribution such as the one in Fig. 4.13 (a), few useful features can be extracted and as a result GCNet cannot predict reliable lightings for such surfaces. For highly-concave shapes under directional lightings, strong cast shadows largely affect the extraction of useful features. Figure 4.13 (b) shows that

GCNet erroneously estimates a highly-concave bowl to be convex. Note that LCNet and PF14 [88] also have similar problems.

4.7 Conclusion

In this chapter, we have first introduced a lighting calibration network, named LCNet, to estimated directional lightings for uncalibrated photometric stereo. To understand what have been learned by LCNet for lighting estimation, we analyze the features learned by the network, and find that attached shadows, shadings, and specular highlights are key elements for lighting estimation. Based on our findings, we then introduced the guided calibration network, named GCNet, that explicitly leverages inter-image information of object shape and intra-image information of shading to estimate more reliable lightings. Experiments on both synthetic and real datasets showed that GCNet achieves significantly better results than LCNet, and demonstrated that our method can be integrated with existing calibrated photometric stereo methods to handle uncalibrated setups.

Chapter 5

Conclusions

5.1 Summary

This thesis has presented learning based solutions for

- transparent object matting from a single image (Chapter 2),
- calibrated photometric stereo for non-Lambertian surfaces (Chapter 3), and
- lighting calibration for uncalibrated photometric stereo (Chapter 4).

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

The problem of transparent object matting from a single image was addressed in Chapter 2. We have introduced a simple and efficient model for transparent object matting, and proposed a CNN architecture, named TOM-Net, that takes a single image as input and predicts environment matte as an object mask, an attenuation mask, and a refractive flow field in a fast feed-forward pass. We created a large-scale synthetic dataset and a real dataset as a benchmark for learning transparent object matting. We have also shown that TOM-Net can perform better by incorporating a trimap or a background image in the input. Promising results have been achieved on both synthetic and real data, which clearly demonstrate the feasibility and effectiveness of the proposed approach.

The problem of calibrated photometric stereo for non-Lambertian surfaces under directional lightings was addressed in Chapter 3. We have proposed a flexible deep fully convolutional network, named PS-FCN, that accepts an arbitrary number of images and their associated light directions as input and regresses an accurate normal map. Our PS-FCN does not require a pre-defined set of light directions during training and testing, and can handle multiple images and light directions in an order-agnostic manner. A data normalization strategy was introduced to better handle surfaces with SVBRDFs. In order to train PS-FCN, two synthetic datasets with various realistic shapes and materials have been created. Results on diverse real datasets have clearly shown that our method outperforms previous calibrated photometric stereo methods.

The problem of lighting estimation for uncalibrated photometric stereo was addressed in Chapter 4. We have first introduced a lighting calibration network, named LCNet, to estimate directional lightings for uncalibrated photometric stereo. To understand what have been learned by LCNet for lighting estimation, we analyse the features learned by the network, and find that attached shadows, shadings, and specular highlights are key elements for lighting estimation. Based on our findings, we then introduced the guided calibration network, named GCNet, that explicitly leverages inter-image information of object shape and intra-image information of shading to estimate more reliable lightings. Experiments on both synthetic and real datasets showed that GCNet achieves significantly better results than LCNet, and demonstrated that our method can be integrated with existing calibrated photometric stereo methods to handle uncalibrated setups.

5.2 Future Work

Although the methods proposed in this thesis are novel and achieve promising results on their specific tasks, there are rooms for improvement.

- Colored transparent object matting under natural illumination

First, our transparent object matting method assumes objects to be colorless and

is not applicable to colored transparent object. Second, we assume a single planar background as the only light source (following most of the previous works), and does not consider the more sophisticated refractive properties of a transparent object under natural illumination (*e.g.*, specular highlight, Fresnel effect, and acoustic shadow). It would be very useful to develop a method for colored transparent object matting under natural illumination in the future.

- Joint estimation of surface normals, reflectances, and lightings from photometric stereo images

We have proposed methods for estimating surface normals and directional lightings from multiple input images of an object. However, we did not explicitly estimate the surface reflectance of the object. By knowing the surface reflectance of an object, we can perform some interesting applications like object appearance editing. It would be helpful to consider the problem of joint estimation of surface normals, reflectances, and lightings from photometric stereo images.

- Photometric stereo under natural illumination

Most of the existing methods for photometric stereo (also in our methods) assume a directional lighting model, and this requires objects to be placed indoor with controllable illuminations. It would be interesting to consider a more general lighting model (*e.g.*, natural illumination), as it allows accurate surface normal estimation of an object outside the laboratory environment.

References

- [1] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, “A comparison and evaluation of multi-view stereo reconstruction algorithms,” in *CVPR*, 2006.
- [2] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, “Building Rome in a day,” *Communications of the ACM*, 2011.
- [3] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, “High accuracy optical flow estimation based on a theory for warping,” in *ECCV*, 2004.
- [4] D. Sun, S. Roth, and M. J. Black, “A quantitative analysis of current practices in optical flow estimation and the principles behind them,” *IJCV*, 2014.
- [5] K. Ikeuchi and B. K. Horn, “Numerical shape from shading and occluding boundaries,” *Artificial intelligence*, 1981.
- [6] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, “Shape-from-shading: a survey,” *TPAMI*, 1999.
- [7] A. R. Smith and J. F. Blinn, “Blue screen matting,” in *SIGGRAPH*, 1996.
- [8] A. Levin, D. Lischinski, and Y. Weiss, “A closed-form solution to natural image matting,” *TPAMI*, 2007.
- [9] R. J. Woodham, “Photometric method for determining surface orientation from multiple images,” *Optical Engineering*, 1980.
- [10] H. Hayakawa, “Photometric stereo under a light source with arbitrary motion,” *JOSA A*, 1994.
- [11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, 1998.
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NIPS*, 2012.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.

- [14] W. M. Silver, “Determining shape and reflectance using multiple images,” Ph.D. dissertation, Massachusetts Institute of Technology, 1980.
- [15] G. Chen, K. Han, and K.-Y. K. Wong, “TOM-Net: Learning transparent object matting from a single image,” in *CVPR*, 2018.
- [16] —, “Learning transparent object matting,” *IJCV*, 2019.
- [17] —, “PS-FCN: A flexible learning framework for photometric stereo,” in *ECCV*, 2018.
- [18] G. Chen, K. Han, B. Shi, Y. Matsushita, and K.-Y. K. Wong, “Deep photometric stereo for non-Lambertian surfaces,” *TPAMI*, 2020.
- [19] —, “Self-calibrating deep photometric stereo networks,” in *CVPR*, 2019.
- [20] G. Chen, W. Michael, B. Shi, Y. Matsushita, and K.-Y. K. Wong, “What is learned in deep uncalibrated photometric stereo?” *ECCV*, 2020.
- [21] D. E. Zongker, D. M. Werner, B. Curless, and D. H. Salesin, “Environment matting and compositing,” in *SIGGRAPH*, 1999.
- [22] Y.-Y. Chuang, D. E. Zongker, J. Hindorff, B. Curless, D. H. Salesin, and R. Szeliski, “Environment matting extensions: Towards higher accuracy and real-time capture,” in *SIGGRAPH*, 2000.
- [23] Y. Wexler, A. W. Fitzgibbon, A. Zisserman *et al.*, “Image-based environment matting,” in *Rendering Techniques*, 2002.
- [24] P. Peers and P. Dutré, “Wavelet environment matting,” in *Eurographics workshop on Rendering*, 2003.
- [25] J. Zhu and Y.-H. Yang, “Frequency-based environment matting,” in *Computer Graphics and Applications*, 2004.
- [26] Q. Duan, J. Zheng, and J. Cai, “Flexible and accurate transparent-object matting and compositing using refractive vector field,” in *Computer Graphics Forum*, 2011.
- [27] Q. Duan, J. Cai, and J. Zheng, “Compressive environment matting,” *The Visual Computer*, 2015.
- [28] Q. Duan, J. Cai, J. Zheng, and W. Lin, “Fast environment matting extraction using compressive sensing,” in *ICME*, 2011.
- [29] Y. Qian, M. Gong, and Y.-H. Yang, “Frequency-based environment matting by compressive sensing,” in *ICCV*, 2015.

- [30] S.-K. Yeung, C.-K. Tang, M. S. Brown, and S. B. Kang, “Matting and compositing of transparent and refractive objects,” *TOG*, 2011.
- [31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *TIP*, 2004.
- [32] X. Shen, X. Tao, H. Gao, C. Zhou, and J. Jia, “Deep automatic portrait matting,” in *ECCV*, 2016.
- [33] D. Cho, Y.-W. Tai, and I. Kweon, “Natural image matting using deep convolutional neural networks,” in *ECCV*, 2016.
- [34] N. Xu, B. Price, S. Cohen, and T. Huang, “Deep image matting,” in *CVPR*, 2017.
- [35] Y. Zhang, L. Gong, L. Fan, P. Ren, Q. Huang, H. Bao, and W. Xu, “A late fusion cnn for digital matting,” in *CVPR*, 2019.
- [36] H. Lu, Y. Dai, C. Shen, and S. Xu, “Indices matter: Learning to index for deep image matting,” in *ICCV*, 2019.
- [37] J. Shi, Y. Dong, H. Su, and S. X. Yu, “Learning non-Lambertian object intrinsics across shapenet categories,” in *CVPR*, 2017.
- [38] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.
- [39] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in *NIPS*, 2014.
- [40] 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 *ICCV*, 2015.
- [41] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, “Spatial transformer networks,” in *NIPS*, 2015.
- [42] D. Kingma and J. Ba, “ADAM: A method for stochastic optimization,” in *ICLR*, 2015.
- [43] J. Kim, J. Kwon Lee, and K. Mu Lee, “Accurate image super-resolution using very deep convolutional networks,” in *CVPR*, 2016.
- [44] S. Nah, T. H. Kim, and K. M. Lee, “Deep multi-scale convolutional neural network for dynamic scene deblurring,” in *CVPR*, 2017.
- [45] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “Flownet 2.0: Evolution of optical flow estimation with deep networks,” in *CVPR*, 2017.

- [46] “Persistence of vision (tm) raytracer,” <http://www.povray.org/>.
- [47] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *ECCV*, 2014.
- [48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *CVPR*, 2009.
- [49] S. Tozza, R. Mecca, M. Duocastella, and A. Del Bue, “Direct differential photometric stereo shape recovery of diffuse and specular surfaces,” *Journal of Mathematical Imaging and Vision*, 2016.
- [50] H.-S. Chung and J. Jia, “Efficient photometric stereo on glossy surfaces with wide specular lobes,” in *CVPR*, 2008.
- [51] R. Ruiters and R. Klein, “Heightfield and spatially varying BRDF reconstruction for materials with interreflections,” in *Computer Graphics Forum*, 2009.
- [52] H. Santo, M. Samejima, Y. Sugano, B. Shi, and Y. Matsushita, “Deep photometric stereo network,” in *ICCV Workshops*, 2017.
- [53] S. Ikehata, “CNN-PS: CNN-based photometric stereo for general non-convex surfaces,” in *ECCV*, 2018.
- [54] T. Taniai and T. Maehara, “Neural inverse rendering for general reflectance photometric stereo,” in *ICML*, 2018.
- [55] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, 2015.
- [56] M. K. Johnson and E. H. Adelson, “Shape estimation in natural illumination,” in *CVPR*, 2011.
- [57] O. Wiles and A. Zisserman, “SilNet: Single-and multi-view reconstruction by learning from silhouettes,” in *BMVC*, 2017.
- [58] W. Matusik, H. Pfister, M. Brand, and L. McMillan, “A data-driven reflectance model,” in *SIGGRAPH*, 2003.
- [59] B. Shi, Z. Mo, Z. Wu, D. Duan, S.-K. Yeung, and P. Tan, “A benchmark dataset and evaluation for non-Lambertian and uncalibrated photometric stereo,” *TPAMI*, 2019.
- [60] N. G. Alldrin, T. Zickler, and D. J. Kriegman, “Photometric stereo with non-parametric and spatially-varying reflectance,” in *CVPR*, 2008.

- [61] P. Einarsson, C.-F. Chabert, A. Jones, W.-C. Ma, B. Lamond, T. Hawkins, M. Bolas, S. Sylwan, and P. Debevec, “Relighting human locomotion with flowed reflectance fields,” in *EGSR*, 2006.
- [62] S. Herbort and C. Wöhler, “An introduction to image-based 3D surface reconstruction and a survey of photometric stereo methods,” *3D Research*, 2011.
- [63] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma, “Robust photometric stereo via low-rank matrix completion and recovery,” in *ACCV*, 2010.
- [64] Y. Mukaigawa, Y. Ishii, and T. Shakunaga, “Analysis of photometric factors based on photometric linearization,” *JOSA A*, 2007.
- [65] D. Miyazaki, K. Hara, and K. Ikeuchi, “Median photometric stereo as applied to the segonko tumulus and museum objects,” *IJCV*, 2010.
- [66] T.-P. Wu and C.-K. Tang, “Photometric stereo via expectation maximization,” *TPAMI*, 2010.
- [67] S. Ikehata, D. Wipf, Y. Matsushita, and K. Aizawa, “Robust photometric stereo using sparse regression,” in *CVPR*, 2012.
- [68] A. S. Georgiades, “Incorporating the Torrance and Sparrow model of reflectance in uncalibrated photometric stereo,” in *ICCV*, 2003.
- [69] B. Shi, P. Tan, Y. Matsushita, and K. Ikeuchi, “Bi-polynomial modeling of low-frequency reflectances,” *TPAMI*, 2014.
- [70] S. Ikehata and K. Aizawa, “Photometric stereo using constrained bivariate regression for general isotropic surfaces,” in *CVPR*, 2014.
- [71] M. Holroyd, J. Lawrence, G. Humphreys, and T. Zickler, “A photometric approach for estimating normals and tangents,” in *TOG*, 2008.
- [72] A. Hertzmann and S. M. Seitz, “Example-based photometric stereo: Shape reconstruction with general, varying BRDFs,” *TPAMI*, 2005.
- [73] Z. Hui and A. C. Sankaranarayanan, “A dictionary-based approach for estimating shape and spatially-varying reflectance,” in *ICCP*, 2015.
- [74] J. Li, A. Robles-Kelly, S. You, and Y. Matsushita, “Learning to minify photometric stereo,” in *CVPR*, 2019.
- [75] Q. Zheng, Y. Jia, B. Shi, X. Jiang, L.-Y. Duan, and A. C. Kot, “SPLINE-Net: Sparse photometric stereo through lighting interpolation and normal estimation networks,” in *ICCV*, 2019.
- [76] X. Wang, D. Fouhey, and A. Gupta, “Designing deep networks for surface normal estimation,” in *CVPR*, 2015.

- [77] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, “3d-r2n2: A unified approach for single and multi-view 3d object reconstruction,” in *ECCV*, 2016.
- [78] W. Hartmann, S. Galliani, M. Havlena, L. Van Gool, and K. Schindler, “Learned multi-patch similarity,” in *ICCV*, 2017.
- [79] I. Sato, T. Okabe, Q. Yu, and Y. Sato, “Shape reconstruction based on similarity in radiance changes under varying illumination,” in *ICCV*, 2007.
- [80] F. Lu, Y. Matsushita, I. Sato, T. Okabe, and Y. Sato, “Uncalibrated photometric stereo for unknown isotropic reflectances,” in *CVPR*, 2013.
- [81] W. Jakob, “Mitsuba renderer,” 2010.
- [82] D. B. Goldman, B. Curless, A. Hertzmann, and S. M. Seitz, “Shape and spatially-varying BRDFs from photometric stereo,” *TPAMI*, 2010.
- [83] A. Paszke, S. Gross, S. Chintala, and G. Chanan, “PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration,” 2017.
- [84] Z. Hui and A. C. Sankaranarayanan, “Shape and spatially-varying reflectance estimation from virtual exemplars,” *TPAMI*, 2017.
- [85] N. G. Alldrin, S. P. Mallick, and D. J. Kriegman, “Resolving the generalized bas-relief ambiguity by entropy minimization,” in *CVPR*, 2007.
- [86] B. Shi, Y. Matsushita, Y. Wei, C. Xu, and P. Tan, “Self-calibrating photometric stereo,” in *CVPR*, 2010.
- [87] Z. Wu and P. Tan, “Calibrating photometric stereo by holistic reflectance symmetry analysis,” in *CVPR*, 2013.
- [88] T. Papadhimetri and P. Favaro, “A closed-form, consistent and robust solution to uncalibrated photometric stereo via local diffuse reflectance maxima,” *IJCV*, 2014.
- [89] F. Lu, X. Chen, I. Sato, and Y. Sato, “SymPS: BRDF symmetry guided photometric stereo for shape and light source estimation,” *TPAMI*, 2018.
- [90] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille, “The bas-relief ambiguity,” *IJCV*, 1999.
- [91] F. Lu, I. Sato, and Y. Sato, “Uncalibrated photometric stereo based on elevation angle recovery from BRDF symmetry of isotropic materials,” in *CVPR*, 2015.
- [92] A. L. Yuille, D. Snow, R. Epstein, and P. N. Belhumeur, “Determining generative models of objects under varying illumination: Shape and albedo from multiple images using SVD and integrability,” *IJCV*, 1999.

- [93] M. K. Chandraker, F. Kahl, and D. J. Kriegman, “Reflections on the generalized bas-relief ambiguity,” in *CVPR*, 2005.
- [94] O. Drbohlav and M. Chantler, “Can two specular pixels calibrate photometric stereo?” in *ICCV*, 2005.
- [95] P. Tan, S. P. Mallick, L. Quan, D. J. Kriegman, and T. Zickler, “Isotropy, reciprocity and the generalized bas-relief ambiguity,” in *CVPR*, 2007.
- [96] C. H. Esteban, G. Vogiatzis, and R. Cipolla, “Multiview photometric stereo,” *TPAMI*, 2008.
- [97] D. Cho, Y. Matsushita, Y.-W. Tai, and I. Kweon, “Photometric stereo under non-uniform light intensities and exposures,” in *ECCV*, 2016.
- [98] T. Okabe, I. Sato, and Y. Sato, “Attached shadow coding: Estimating surface normals from shadows under unknown reflectance and lighting conditions,” in *ICCV*, 2009.
- [99] K. Midorikawa, T. Yamasaki, and K. Aizawa, “Uncalibrated photometric stereo by stepwise optimization using principal components of isotropic BRDFs,” in *CVPR*, 2016.
- [100] D. Cho, Y. Matsushita, Y. W. Tai, and I. S. Kweon, “Semi-calibrated photometric stereo,” *TPAMI*, 2018.
- [101] Y. Quéau, T. Wu, F. Lauze, J.-D. Durou, and D. Cremers, “A non-convex variational approach to photometric stereo under inaccurate lighting,” in *CVPR*, 2017.
- [102] R. Basri, D. Jacobs, and I. Kemelmacher, “Photometric stereo with general, unknown lighting,” *IJCV*, 2007.
- [103] Z. Mo, B. Shi, F. Lu, S.-K. Yeung, and Y. Matsushita, “Uncalibrated photometric stereo under natural illumination,” in *CVPR*, 2018.
- [104] B. Haefner, Z. Ye, M. Gao, T. Wu, Y. Quéau, and D. Cremers, “Variational uncalibrated photometric stereo under general lighting,” in *ICCV*, 2019.
- [105] M.-A. Gardner, K. Sunkavalli, E. Yumer, X. Shen, E. Gambaretto, C. Gagné, and J.-F. Lalonde, “Learning to predict indoor illumination from a single image,” *TOG*, 2017.
- [106] Y. Hold-Geoffroy, K. Sunkavalli, S. Hadap, E. Gambaretto, and J.-F. Lalonde, “Deep outdoor illumination estimation,” in *CVPR*, 2017.
- [107] H. Weber, D. Prévost, and J.-F. Lalonde, “Learning to estimate indoor lighting from 3d objects,” in *3DV*, 2018.

- [108] H. Zhou, J. Sun, Y. Yacoob, and D. W. Jacobs, “Label denoising adversarial network (LDAN) for inverse lighting of faces,” in *CVPR*, 2018.