Beyond Photo-Consistency: Shape, Reflectance, and Material Estimation Using Light-Field Cameras

By

Ting-Chun Wang

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ravi Ramamoorthi, Chair
Professor Alexei A. Efros, Co-Chair
Professor Martin S. Banks

Spring 2017

Beyond Photo-Consistency: Shape, Reflectance, and Material Estimation Using Light-Field Cameras

Abstract

Beyond Photo-Consistency: Shape, Reflectance, and Material Estimation Using Light-Field Cameras

by

Ting-Chun Wang

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Ravi Ramamoorthi, Chair
Professor Alexei A. Efros, Co-Chair

Light-field cameras have recently become easily accessible in the consumer market, making applications such as post-shot refocusing and viewpoint parallax possible. An important benefit of light-field cameras for computer vision is that multiple viewpoints or sub-apertures are available in a single light-field image, enabling passive depth estimation. However, most existing approaches are based on *photo-consistency*, i.e., all viewpoints exhibit the same color when focused to the correct depth. This assumption does not hold in a number of circumstances, e.g., in the presence of occlusions, and when the surface is not Lambertian.

In this thesis, we refrain from assuming photo-consistency, and explicitly deal with the situations where it fails to hold. First, we propose a novel framework that can handle occlusions when estimating depth, so we are able to get sharper occlusion boundaries in the obtained depth maps. Next, we extend traditional optical flow to the case of glossy surfaces, and derive a spatially-varying (SV) BRDF invariant equation. Using this equation, we can then recover both shape and reflectance simultaneously using light-field cameras. Finally, we show an application of recognizing materials in light-field images, based on the reflectance information extracted using learning-based methods. By looking beyond photo-consistency, we are able to estimate better depths and recover reflectance and material types, which can be useful for a variety of vision and graphics applications.

To My Parents and Friends

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisors, Prof. Ravi Ramamoorthi and Prof. Alyosha Efros, for guiding me through a fantastic journey of my Ph.D. career. I had all the freedom to explore all different possibilities, finding things that truly interest me. Looking back, advisors are just like alchemists: any stones I had, they can always turn them into precious gems. This trip becomes full of surprises because of them.

I thank my co-authors and those who helped in all of my publications: Michael Tao for building my elementary knowledge of light-fields; Jun-Yan Zhu for advice on network structures and building applications; Manmohan Chandraker for his valuable advice on the reflectance and material projects; Ebi Hiroaki for collecting the light-field dataset; Manohar Srikanth for ideas of the multi-camera system; Jong-Chyi Su for his help on our specularity journal paper; Jitendra Malik for his advice on both specularity papers; and Nima Khademi Kalantari for his ideas on the view synthesis papers.

I would also like to thank my labmates Jiamin Bai, Lingqi Yan, Pratul Srinivasan, Weilun Sun, and Cecilia Zhang for inspiring me for project ideas; Tai-Hsiang Huang, Homer Chen, Yi-Nung Liu, and Shao-Yi Chien for building the foundations in my undergraduate years; and Shirley Salanio for help with all the graduate student matters. Life would really become a mess without any of you.

I also acknowledge the financial support from a Berkeley Fellowship, ONR grant N00014152013, and support by Google, Draper Lab, Intel, Nokia, Samsung and Sony.

Finally, I thank my parents and all my friends, for their unyielding applauses and shoulders for me through all the ups and downs. A Ph.D. life could be tough, but becomes better because of all of you. You are the reason why I became who I am today.

Chapter 1

Introduction

To reconstruct a scene, usually there are many important properties we need to know about it. For example, which objects are further or closer than the others? What are the shapes of these objects? Are the objects shiny? Are they made of metals or wood? Humans can very easily answer these questions by a simple glance at the scene. However, computers usually have a hard time estimating these properties. In this dissertation, we introduce a series of frameworks to estimate the shapes, the reflectances, and the materials of objects given images of the scene.

Much effort has been devoted to recovering the 3D structures from images of the scene; We can mainly divide these works into two large areas: using a single view (monocular cues) or using multiple views (stereo cues). A single view is basically one picture captured by a conventional camera (e.g. phone cameras or DSLRs). Since only one view is captured, recovering shapes often relies on many assumptions of the scene, and usually adopts a training procedure. Multiple views, on the other hand, can correspond to multiple pictures (or a video which consists of multiple frames) taken with one camera, or pictures taken with multiple cameras at the same time. A very popular case is when we have two cameras in the system, which is often referred to as a stereo camera, just like humans have two eyes to help us estimate the distances of objects. In general, multi-views often perform better than single views due to the extra information; However, the drawback of using multiple views is the extra time for taking multiple shots, or the complicated setup of adding more cameras.

In this work, we approach this problem using a special kind of camera, called light-field cameras, which can be considered as a special case of using multiple views. Moreover, it combines the advantages of both methods: it contains the multi-view information, but alleviates the extra exposure or hardware problems using a technique called micro-lens. By using light-field cameras, multiple views can be captured in one single shot using one single camera, so shapes can be more robustly estimated with minimal effort.

Indeed, many methods have been proposed to recover depth using stereo or light-field cameras.

To make the estimation easier, most existing approaches rely on an assumption called photo-consistency, which means all cameras should see the same color when they are all focused on the same point in the scene. However, in the real world this assumption often fails to hold, especially in two very important situations: around occlusion boundaries, and on glossy surfaces.

In this work, we try to get rid of this assumption, and explicitly deal with these situations. We show that by doing so, depth can be better estimated around the occlusion boundaries (Ch. 3). Next, by extending the scope to specular objects, we also enable our system to recover reflectance in addition to shape (Ch. 4), which can be used for many applications. For example, we develop a system to determine material types, and can segment an image into different regions according to their material classes (Ch. 5). A light-field dataset is also collected to verify the correctness and robustness of our system.

Therefore, by looking beyond photo-consistency, we are able to estimate shapes, reflectances, and materials of the objects, which are three essential properties to reconstruct a scene. These works open a new door for light-field related problems, and can be used to inspire further research in this field. Below we introduce the basic idea of light-fields (Ch. 1.1), the photo-consistency assumption and its problems (Ch. 1.2), and the overview of this dissertation (Ch. 1.3).

1.1 Light-field

In this section, we briefly introduce the concept of a light-field camera, the difference between a conventional camera and a light-field camera, and the advantages of using light-field cameras.

We first start with the simplest camera – a pinhole camera. In a pinhole camera, the lens that all rays pass through is just a pinhole; Therefore, each pixel on the sensor corresponds to exactly one direction from one point in the scene, as shown in Fig. 1.1.[1] Note that only single view information can be recorded using a pinhole camera.

Since pinhole cameras can only let very little light pass through, the resulting image is usually too dark unless hours of exposure time is given. Therefore, pinhole cameras are seldom used in practice. For a conventional real-world camera, the pinhole is replaced with a lens. This changes two things dramatically. First, the passage for light is now larger, so more light can be gathered without exploiting very long exposure time. Sec-

[1] In reality, the pixel itself also spans some finite distance on the sensor, and the final color value is determined by light rays summed over the width of this pixel. To make explanation easier, this is omitted in the main text.

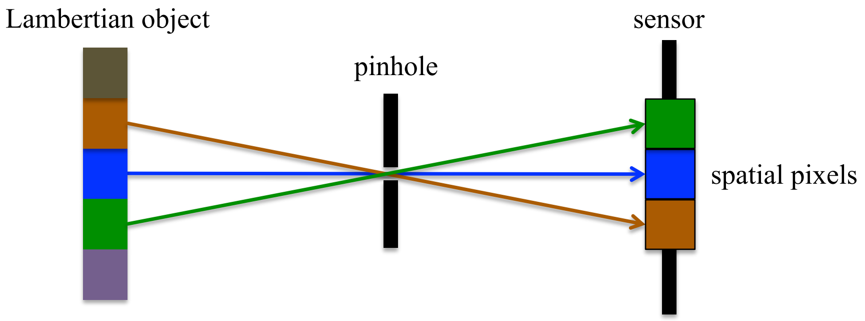


Figure 1.1: *Recorded rays in a pinhole camera. For each pixel in the camera, it records exactly one direction coming from one point in the scene.*

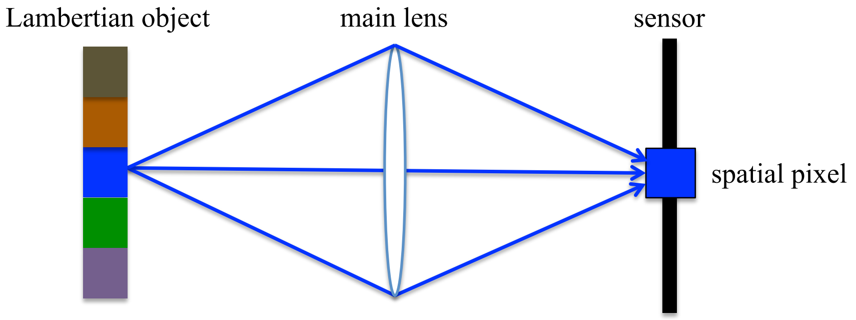


Figure 1.2: *Recorded rays in a conventional camera, which has a finite lens. For a pixel in a conventional camera, it integrates all light rays from different directions (three directions shown in the figure), and only records the integrated value as the pixel color. Therefore, distinguishing rays from different directions is ambiguous.*

ond, since we now have a hole of finite width, light rays of different directions from the same scene point can now reach the sensor. Each pixel in the picture now represents a summation of light rays from a range of different directions, hitting a particular point on the sensor. Since we now record rays from different directions, ideally we should be able to extract multi-view information from the image. However, since only the integration is recorded instead of the individual rays, lights coming from different directions become indistinguishable, and it is impossible to tell what portion of the pixel is contributed to by a particular direction (Fig. 1.2). This is the drawback of using conventional cameras; To utilize multi-view information, we must take multiple pictures from different positions.

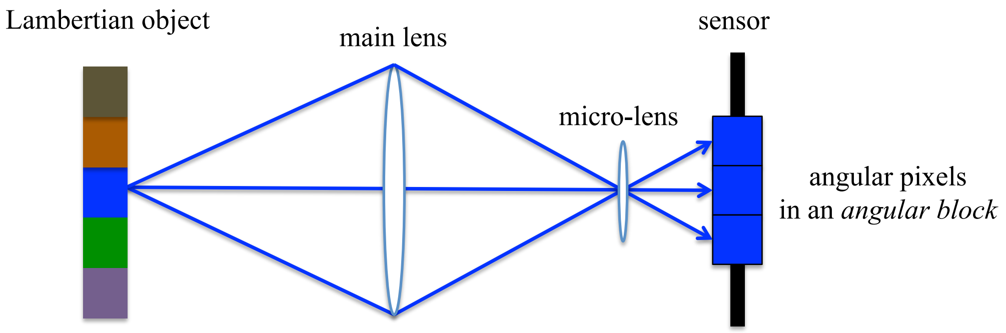


Figure 1.3: *Recorded rays in a light-field camera. In a light-field camera, there is a layer of micro-lens in front of the sensor, which splits rays coming from different directions to different (angular) pixels. Since rays of different directions are now recorded separately, we can extract the multi-view information from the image pixels.*

Fortunately, that is not the only solution. To distinguish rays from different directions, another approach is to use light-field cameras. The concept of light-fields can date back to [1]. Originally, it was proposed for use in rendering. More recently, a technique called micro-lens is proposed [2] to implement light-fields using a conventional camera. The basic idea is to add a layer of micro-lens in front of the sensor. The micro-lens then splits rays coming from different directions to different pixels on the sensor. These pixels, dubbed angular pixels, then record these different directions of rays (Fig. 1.3). Since different viewpoints are now recorded separately, we are able to extract the multi-view information from the resulting image easily. This is the advantage of using light-field cameras compared to conventional cameras. Next, we discuss about common assumptions adopted by most multi-view methods to estimate depth, the cases that the assumptions fail to hold, and how we deal with them.

1.2 Photo-consistency

Photo-consistency is an assumption commonly adopted for multi-view methods. Basically, it assumes that if we focus the camera on a particular point in the scene, all light rays (in the corresponding angular block) should come from that point, and thus exhibit the same color. For example, in Fig. 1.3, the camera is focused on the blue point of the object, so the recorded angular pixels should reflect that and have the same blue color. Thus, if we observe a uniform color in the angular block, we know that we are focused at the point, and the focused distance is the depth of that point. On the other hand, if the

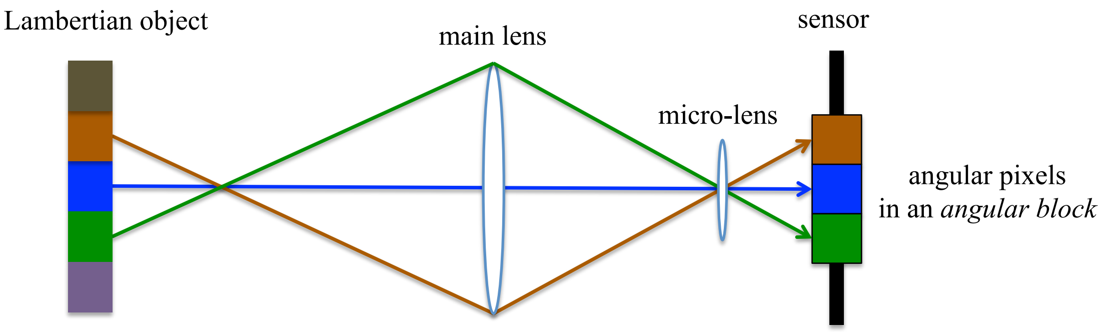


Figure 1.4: *Recorded rays when a light-field camera is out of focus.*

camera is focused incorrectly, different angular pixels will correspond to different points in the scene, thus exhibiting different colors (Fig. 1.4). Based on this assumption, we can estimate the depth of objects by looking at the coherence in the angular block.

However, this assumption does not always hold, and fails in two very common and important cases: around occlusion boundaries and on glossy objects. Below we briefly discuss why these situations would cause the assumption to fail.

Occlusion issue Photo-consistency assumes that all rays come from the same point in the scene when refocused on it. However, this is not the case when there exists an occluder between the point and the camera, where the point is occluded in some views. For example, in Fig. 1.5, we have a red occluder in front of the object, and the upper ray is occluded by it, so the corresponding angular pixel actually records a different color (red) than the other angular pixels (blue). This is an important issue since occlusions are very common in real-world images, and obtaining sharp boundaries across depth discontinuities is essential for many applications, e.g. image segmentations. However, due to the assumption failure, most existing methods will get smoothed depths across occlusion boundaries, and thus rely on heavy regularization or post processing to correct that.

To address this issue, we explicitly model this phenomenon in light-fields, and propose a corresponding framework to handle it. The proposed system enables us to predict potential regions in the angular block that will fail the photo-consistency metric. These parts are then excluded in the depth estimation process. Since the remaining pixels all come from a single point, photo-consistency holds on them and previous approaches can be adopted to estimate depth. This dramatically improves the generated depth maps around occlusions. Moreover, we also propose a framework to explicitly estimate occlusion boundaries in the image. These boundaries are then used for depth regularization, and possibly other applications as well.

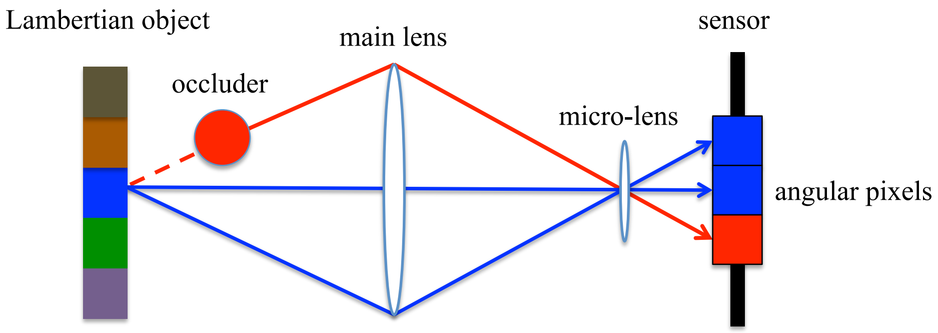


Figure 1.5: *Photo-consistency fails in the presence of occlusion. Were there not the occluder, all angular pixels should exhibit the same color (blue). However, since the occluder exists, the upper ray gets occluded, and the corresponding angular pixel actually records a different color (red) than the other angular pixels (blue).*

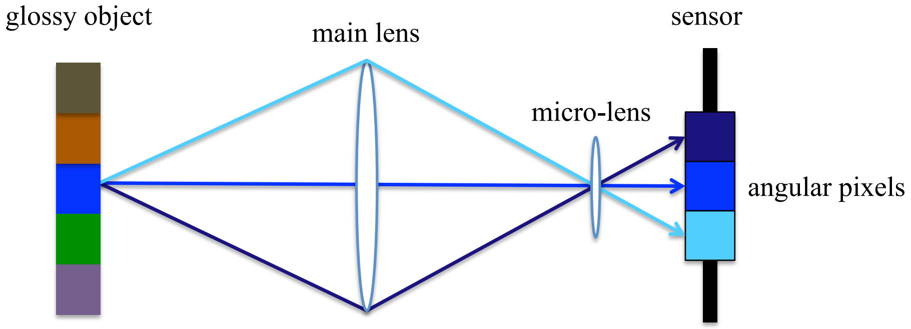


Figure 1.6: *Photo-consistency fails on glossy objects. For glossy objects, the observed intensities would be different when viewed from different perspectives. Therefore, even if all rays come from the same point, they are still not necessarily the same color.*

Glossy objects In addition to occlusions, photo-consistency also fails on glossy objects. To understand why, first we need to know the difference between Lambertian and glossy objects. For a Lambertian object, its appearance is the same regardless of the observer's viewpoint. This means no matter which direction the light is coming from, the observer/sensor will always see the same color. However, this is not the case for glossy objects, e.g. metals or plastics. For these objects, their appearances will change when viewed from different directions (Fig. 1.6). The function that describes this appearance variation is commonly referred to as Bidirectional Reflectance Distribution Function (BRDF).

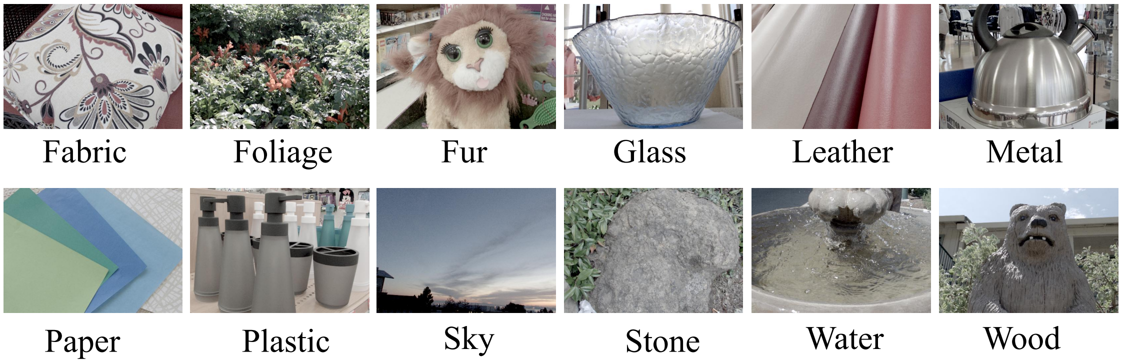


Figure 1.7: *Example images in our dataset of* 12 *material classes.*

Since lights from different directions result in different colors, photo-consistency fails to hold, and methods based on that cannot estimate depth robustly.

To address the problem, we incorporate this color variation caused by different viewpoints into the estimation system. We modify the traditional optical flow framework to model this phenomenon, and show that we will introduce two additional unknowns into the system. By adopting a half-angle BRDF model and a locally smooth prior on shape, we are able to solve depth directly without alternating between solving shape and solving reflectance.

Moreover, by extending the system to non-Lambertian objects, we also enable estimation of reflectance in addition to shapes. The recovered reflectance is helpful in a number of applications, e.g. classifying materials in the scene. When determining material types, reflectance is often a more informative cue compared to colors. This is because reflectance is an intrinsic property of materials, while colors of different instances may vary substantially. However, current approach for estimating object reflectances often requires using gantries, and thus takes many efforts. Instead, by extracting this information implicitly using multiple views, we develop a learning based system to recognize materials using light-field images. We show that the results compare favorably to using single 2D images. Compared with other reflectance-based methods, ours also has minimal capture effort due to the single shot nature of light-field cameras. Finally, to verify our system, we also collect a light-field dataset to train and test different architectures. At the time of writing, this is the largest light-field dataset, which contains 1200 images. Example dataset images are shown in Fig. 1.7, and an example classification result is shown in Fig. 1.8.

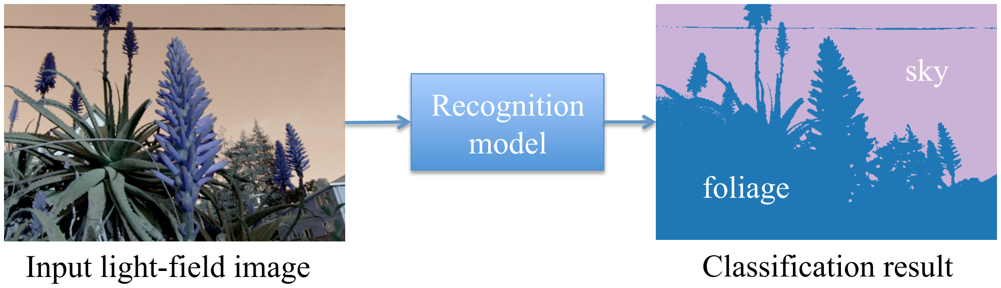


Figure 1.8: *Material segmentation example. Using a neural network model, we are able to classify different pixels in a light-field image into different material classes, thus performing material segmentation.*

1.3 Dissertation Overview

In this dissertation, we demonstrate the benefits by not making the photo-consistency assumption, and show how depth and reflectance estimation can be improved. In particular, since occlusions violate photo-consistency, by treating this explicitly, we are able to get sharper depth boundaries around occlusions than previous methods. Furthermore, we are also able to deal with glossy objects in addition to Lambertian objects, and estimate both shape and reflectance simultaneously. Finally, we utilize the estimated reflectance and apply a neural network approach to classify material types in an image. The description for each chapter is summarized below.

Chapter 2. Light-Field Cameras: Hardware and Applications We first briefly introduce the light-field camera we use in this thesis, and describe some potential benefits and applications. Then, we discuss current main research areas using light-field cameras, including super-resolution and depth estimation.

Chapter 3. Depth Estimation with Occlusion Modeling In this chapter, we derive an occlusion model which explicitly models the phenomenon when occlusion occurs in light field images. We start with the traditional pinhole model, and show that occlusion issues in the light-field setup can be expressed using a "reversed" pinhole model. Using this model, we are able to identify which cameras might observe the occluded view, and discard them while computing the cost functions for depth estimation. Based on the model, occlusion boundaries can also be predicted and used to better regularize the depth map. We show that the obtained depths have much sharper boundaries around occlusion regions compared to current state-of-the-art approaches.

Chapter 4. SVBRDF-Invariant Shape and Reflectance Estimation Next, we develop a depth estimation framework which works on glossy objects, and can also estimate the reflectance in addition to shape. Most previous works handle this problem by solving an iterative optimization, which alternates between solving depth and reflectance. Our approach, on the other hand, can directly solve them without any alternating steps, and can deal with spatially-varying (SV) BRDFs. We do this by incorporating an additional term into the traditional optical flow, which accounts for the intensity change caused by the viewpoint change. We then solve this new introduced ambiguity by adopting a special class of BRDF model, as well as applying a locally polynomial shape prior on the observed objects. After shape is recovered, reflectance can also be recovered easily. Experimental results demonstrate that our approach works better compared to previous methods on a variety of different materials, especially at specular regions.

Chapter 5. Material Recognition with Light-Field Cameras Light-field images inherently capture the reflectance information of objects by taking multiple views at once. This information should be very useful when determining material types, since reflectance, as opposed to colors, is an intrinsic property of materials. Thus, instead of using 2D images to perform material recognition, we believe using light-field images should result in better performance. To perform the classification, we adopt the convolutional neural network (CNN) approach, which is currently the state-of-the-art for material recognition on 2D images. The biggest challenge, however, is to adapt 4D light-field data to these 2D pre-trained networks. We designed and experimented with several architectures to adapt to this change. To train and evaluate our networks, we also collect the first mid-sized light field dataset with ground truth material labels to validate our proposition. We show that we indeed gain higher classification accuracy with light-field images, which can spur further research in this direction.

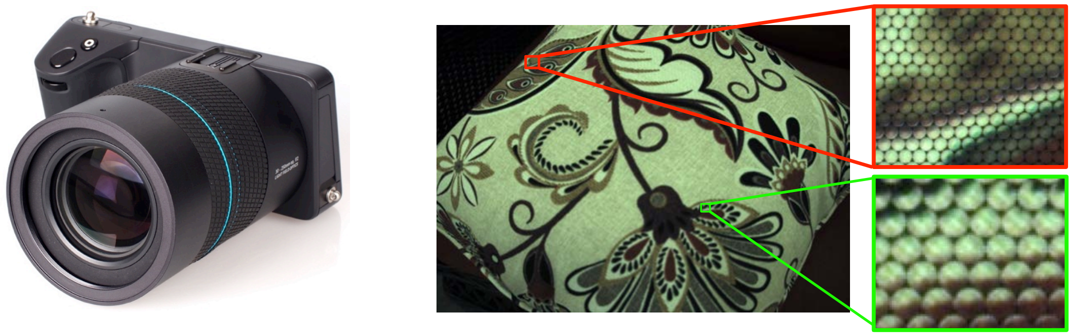
Chapter 6. Conclusion Finally, this chapter summarizes the work in this thesis, namely reconstructing shape, reflectance, and materials using light-field cameras by looking beyond photo-consistency. We also discuss several possible directions for future work.

Chapter 2

Light-Field Cameras: Hardware and Applications

Since the introduction of light-fields, there have been many different light-field cameras on the market, including Pelican, Raytrix, and Lytro. Probably the most well-known for consumers are the Lytro cameras, which include two generations: the first generation and the ILLUM generation. In this thesis, we only focus on using the Lytro ILLUM camera.

2.1 Lytro ILLUM camera

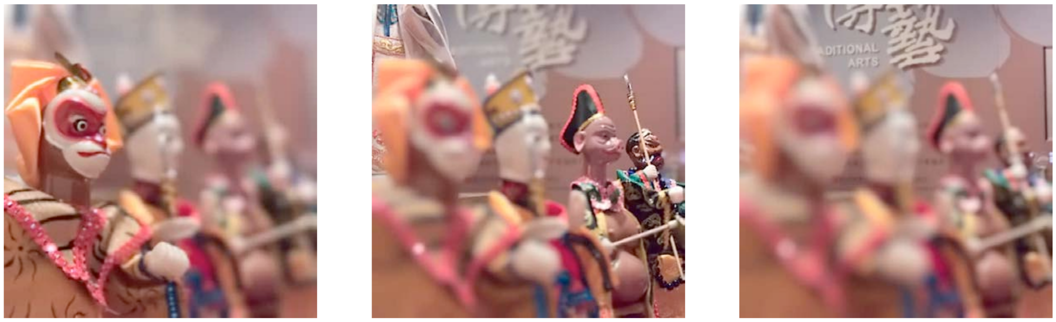


(a) Lytro ILLUM camera

(b) Example micro-lens image

Figure 2.1: *Lytro ILLUM camera and an example micro-lens image taken with it. Each hexagonal block represents different viewpoints of the object.*

The ILLUM camera and an example image taken with it are shown in Fig. 2.1. The camera adopts the micro-lens technology introduced in Ch. 1.1. Each raw image taken



Near focus Middle focus Far focus

Figure 2.2: *Post-shot image refocusing. After we take a photo, we can still refocus to any depth we want.*

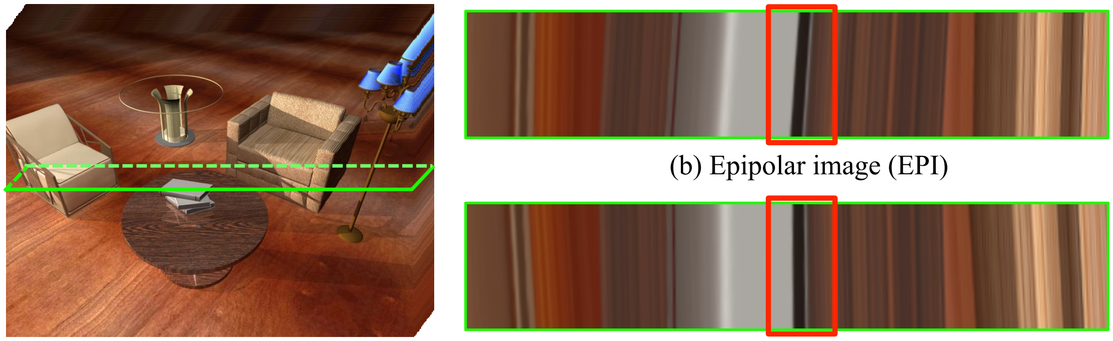
with the ILLUM camera is roughly 5368×7728 pixels. The image contains an array of hexagonal blocks, where each pixel block is a collection of different viewpoints of objects in the scene. To utilize the multi-view information, we must first decode the image into a standard format. Using the official toolbox released by Lytro [3], we are able to extract images with 14×14 angular resolution and 376×541 spatial resolution. These images can then be used for a number of different applications, which are described in the following subsection.

2.1.1 Applications

Given a light-field image, some very common applications are post-shot refocusing, (effective) aperture changing, and view changing.

Refocusing After a light-field image is taken, we can still refocus to any point in the image. A refocusing example is shown in Fig. 2.2. This is achieved by resampling the light-field image using a 4D shearing operation.

To see this, let's first consider a 3D case where the camera viewpoint only shifts in 1D. As shown in Fig. 2.3, if we stack all images into a large volume, and look at the side view profile, we get another image, which is often called the epipolar plane image (EPI). If we collapse the volume and average all the images, we get a normal 2D image. In this case, if a pixel in the final image corresponds to a vertical line in the original EPI, then all its colors come from the same point in the scene. Thus, this pixel is in focus in the final image. On the other hand, if that pixel corresponds to a tilted line in the original EPI (Fig. 2.3b), it will mix with colors from other points, and become out of focus in the final image.



(a) Input light-field volume

(b) Epipolar image (EPI)

(c) Sheared epipolar image

Figure 2.3: *The epipolar plane image (EPI). (a) The input light-field volume. (b) The side view profile of one horizontal cut on (a). Note that the black stripe in the red box is tilted, so if we average the image along the vertical direction, that region will become out of focus. (c) After we shear the EPI in (b) so that the black stripe becomes a vertical line, the region will be in focus once we average along the vertical direction.*

To refocus to a particular point in the scene, we can first "shear" the volume so that the corresponding line becomes vertical in the EPI (Fig. 2.3c). Then when we collapse the volume, this new point will become in focus instead. This implies that refocusing can be done using a 3D shearing on the light-field data.

Finally, when the camera moves viewpoints in 2D instead of 1D, we can apply a 4D shearing on the light-field data instead of a 3D shearing, and collapse the 4D volume to obtain the final refocused image.

Aperture changing We can also change the effective aperture size to create different depths of field using light-field images. Since the micro-lens approach splits rays into different directions, each sub-aperture view is close to a pinhole image. Therefore, if we just pick the central view image, the effective aperture is very small, and we can get an image where everything is in focus. However, as we average more and more views to get the final image, the effective aperture becomes larger and larger, and we are able to get an image with a very shallow depth of field.

View changing Finally, given a light-field image we can change our viewpoints and see different parallax around the objects. This is done by simply picking different views in the light-field image.

2.2 Previous work

Given the recent popularity on light-field cameras, much effort has been devoted to researches on this topic. Most existing light-field works focus on two main aspects: increasing the image resolution and estimating depth. Below we briefly describe the work that has been developed in these two areas.

2.2.1 Super-resolution

For most existing light-field cameras, the micro-lens approach is adopted. However, note that the product of the spatial resolution and the angular resolution is just the sensor resolution, so there is an inherent tradeoff between these two resolutions. Therefore, light-field images usually have a much lower spatial resolution compared to traditional 2D images. To deal with the low resolution, numerous super-resolution approaches have been proposed. They aim to increase either the spatial resolution, the angular resolution, or both.

Spatial super-resolution To increase the spatial resolution, Bishop et al. [4] formulate a variational Bayesian framework to reconstruct both the surface depth and a higher resolution light-field. Cho et al. [5] explicitly model the calibration pipeline of Lytro cameras and propose a learning based interpolation method to obtain higher spatial resolution. Boominathan et al. [6] combine a Lytro camera with a DSLR to apply a dictionary-based super-resolution technique.

Angular super-resolution To reconstruct novel views from sparse angular samples, some methods require the input to follow a specific pattern, or to be captured in a carefully designed way. For example, the work by Levin and Durand [7] takes in a 3D focal stack sequence and reconstructs the light-field, using a prior based on the dimensionality gap. Shi et al. [8] leverage sparsity in the continuous Fourier spectrum to reconstruct a dense light-field from a 1D set of viewpoints. Marwah et al. [9] propose a dictionary-based approach to reconstruct light-fields from a coded 2D projection.

For super-resolution with more flexible inputs, Zhang et al. [10] propose a phase-based approach to reconstruct light-fields from a micro-baseline stereo pair. Kalantari et al. [11] develop a learning-based framework to synthesize arbitrary views given four corner views of the light-field.

Joint super-resolution To simultaneously increase the spatial and the angular resolutions, Mitra and Veeraraghavan [12] introduce a patch-based approach where they model the light-field patches using a Gaussian mixture model. Wanner and Goldluecke [13] propose an optimization approach to reconstruct images at novel views with higher resolution from an input light-field. Yoon et al. [14] perform spatial and angular super-resolution on light-fields using convolutional neural networks (CNN).

2.2.2 Depth estimation

In addition to super-resolution, depth estimation is also a popular research area in light-fields. Below we briefly introduce two main approaches for estimating depth: epipolar plane image (EPI) based and multi-view based methods.

EPI based methods As we have seen in Fig. 2.3, an object is in focus if the corresponding line in the EPI is vertical. The more tilted the line is, the farther away the object is from the current focused distance. Thus, the slope of the line indicates the depth of the object. Therefore, given a light-field image, EPI based methods first try to find the line directions in the EPIs. Once the line directions are estimated, the depth of the objects can be determined accordingly. For example, Wanner et al. [15, 13] propose using structure tensors to measure the line directions for each pixel in an EPI. Tosic and Berkner [16] utilize the normalized second derivative of the Ray Gaussian kernel to detect rays in EPIs. To estimate the line orientations, Zhang et al. [17] propose a spinning parallelogram operator (SPO) and compute the distribution distance. In general, these methods are faster to compute, but are more vulnerable to noise compared to multi-view based methods.

Multi-view based methods As opposed to EPI methods, multi-view methods first define a number of discrete depth levels. Then for each depth level, images of different views are warped to the central view, and the corresponding error on the warped images is computed using certain metric. For example, Tao et al. [18] adopt a metric which combines correspondence and defocus cues, which can complement the disadvantages of each other. Heber and Pock [19] assume the matrix containing all warped images should have low rank, and solve a convex optimization problem. On the other hand, Jeon et al. [20] propose a phase-based interpolation method to replace bilinear interpolation, thus increasing the accuracy for sub-pixel shift. Most of these methods do not explicitly take occlusion issues into account, which motivates our work in the next chapter.

Chapter 3

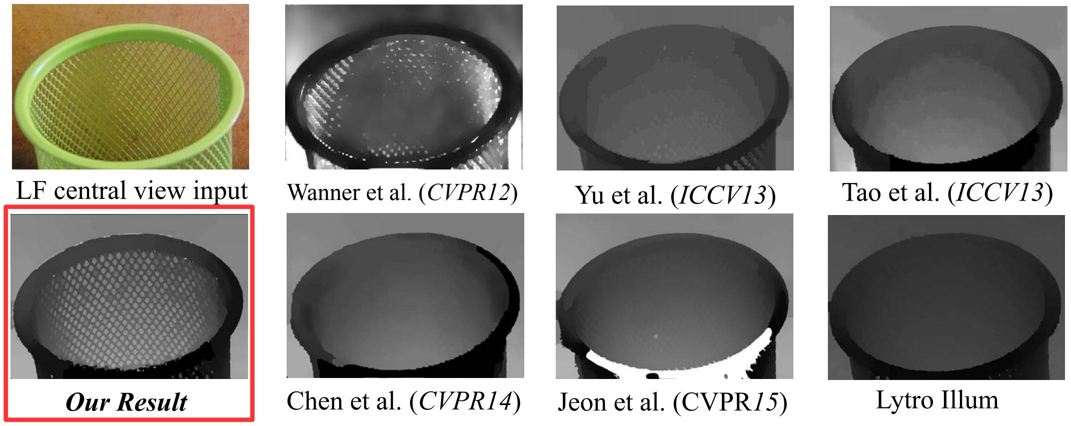
Depth Estimation with Occlusion Modeling

As discussed in the last chapter, most previous depth estimation approaches do not model occlusions explicitly, and therefore fail to capture sharp object boundaries. A common assumption is that for a Lambertian scene, a pixel will exhibit photo-consistency, which means all viewpoints converge to a single point when focused to its depth. However, in the presence of occlusions this assumption fails to hold, making most current approaches unreliable precisely where accurate depth information is most important – at depth discontinuities.

In this chapter, an occlusion-aware depth estimation algorithm is developed; the method also enables identification of occlusion edges, which may be useful in other applications. It can be shown that although photo-consistency is not preserved for pixels at occlusions, it still holds in approximately half the viewpoints. Moreover, the line separating the two view regions (occluded object vs. occluder) has the same orientation as that of the occlusion edge in the spatial domain. By ensuring photo-consistency in only the occluded view region, depth estimation can be improved. Occlusion predictions can also be computed and used for regularization. Experimental results show that our method outperforms current state-of-the-art light-field depth estimation algorithms, especially near occlusion boundaries.

3.1 Introduction

Light-field cameras from Lytro [21] and Raytrix [22] are now available for consumer and industrial use respectively, bringing to fruition early work on light field rendering [23, 1]. An important benefit of light field cameras for computer vision is that multiple



LF central view input

Wanner et al. (*CVPR12*)

Yu et al. (*ICCV13*)

Tao et al. (*ICCV13*)

Our Result

Chen et al. (*CVPR14*)

Jeon et al. (CVPR*15*)

Lytro Illum

Figure 3.1: *Comparison of depth estimation results of different algorithms from a light field input image. Darker represents closer and lighter represents farther. It can be seen that only our occlusion-aware algorithm successfully captures most of the holes in the basket, while other methods either smooth over them, or have artifacts as a result.*

viewpoints or sub-apertures are available in a single light-field image, enabling passive depth estimation [24]. Indeed, Lytro Illum and Raytrix software produces depth maps used for tasks like refocusing after capture, and recent work [18] shows how multiple cues like defocus and correspondence can be combined.

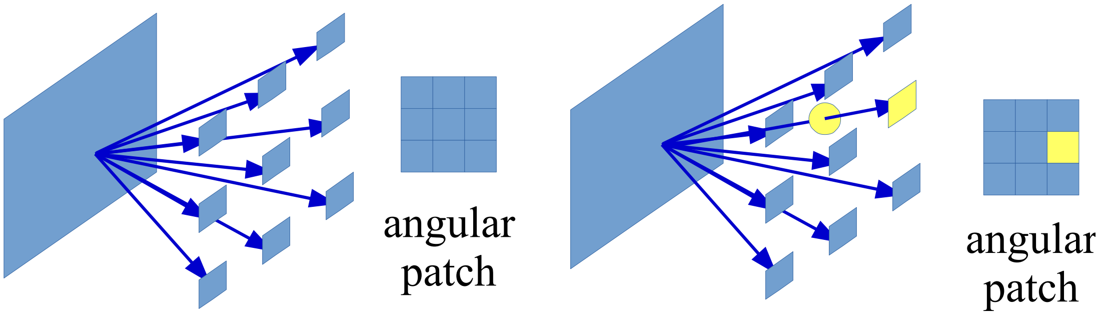
However, very little work has explicitly considered occlusion before. A common assumption is that, when refocused to the correct depth (i.e., the depth of the center view), angular pixels corresponding to a single spatial pixel represent viewpoints that converge to the same point in the scene. If we collect these pixels to form an *angular patch* (Eq. 3.6), they exhibit photo-consistency for Lambertian surfaces, which means they all share the same color (Fig. 3.2a). However, this assumption is not true when occlusion occurs at a pixel; photo-consistency no longer holds since some viewpoints will now be blocked by the occluder (Fig. 3.2b). Enforcing photo-consistency on these pixels will often lead to incorrect depth results, causing smooth transitions around sharp occlusion boundaries.

In this chapter, we explicitly model occlusions, by developing a modified version of the photo-consistency condition on angular pixels [25, 26]. Our main contributions are:

1. An occlusion prediction framework on light field images that uses a modified angular photo-consistency.

2. A robust depth estimation algorithm which explicitly takes occlusions into account.

We show (Sec. 3.3) that around occlusion edges, the angular patch can be divided into



(a) Non-occluded pixels

(b) Occluded pixels

Figure 3.2: *Non-occluded vs. occluded pixels. (a) At non-occluded pixels, all view rays converge to the same point in the scene if refocused to the correct depth. (b) However, photo-consistency fails to hold at occluded pixels, where some view rays will hit the occluder.*

two regions, where only one of them obeys photo-consistency. A key insight (Fig. 3.3) is that the line separating the two regions in the *angular domain* (correct depth vs. occluder) has the same orientation as the occlusion edge does in the *spatial domain*. This observation is specific to light-fields, which have a dense set of views from a planar camera array or set of sub-apertures. Although a stereo camera also satisfies the model, the sampling in angular domain is not sufficient to observe an orientation of the occlusion boundary.

We use the modified photo-consistency condition, and the means/variances in the two regions, to estimate initial occlusion-aware depth (Sec. 3.4). We also compute a predictor for the occlusion boundaries, that can be used as an input to determine the final regularized depth (Sec. 3.5). These occlusion boundaries could also be used for other applications like segmentation or recognition. As seen in Fig. 5.1, our depth estimates are more accurate in scenes with complex occlusions (previous results smooth object boundaries like the holes in the basket). In Sec. 3.6, we present extensive results on both synthetic data (Figs. 3.9, 3.10), and on real scenes captured with the consumer Lytro Illum camera (Fig. 3.11), demonstrating higher-quality depth recovery than previous work [27, 18, 15, 28].

3.2 Related Work

(Multi-View) Stereo with Occlusions: Multi-view stereo matching has a long history, with some efforts to handle occlusions. For example, the graph-cut framework [29] used an occlusion term to ensure visibility constraints while assigning depth labels. Woodford et al. [30] imposed an additional second order smoothness term in the optimization, and solved it using Quadratic Pseudo-Boolean Optimization [31]. Based on this, Bleyer et al. [32] assumed a scene is composed of a number of smooth surfaces and proposed a soft segmentation method to apply the asymmetric occlusion model [33]. However, significant occlusions still remain difficult to address even with a large number of views.

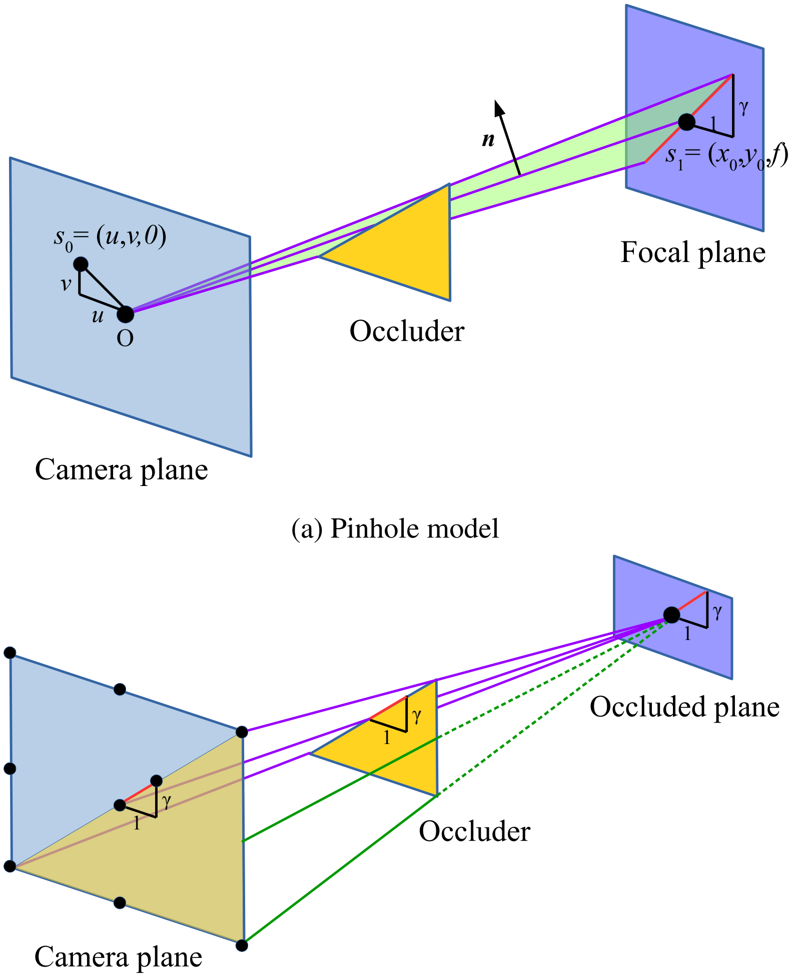
Depth from Light Field Cameras: Perwass and Wietzke [22] proposed using correspondence techniques to estimate depth from light-field cameras. Tao et al. [18] combined correspondence and defocus cues in the 4D Epipolar Image (EPI) to complement the disadvantages of each other. Neither method explicitly models occlusions. McCloskey [34] proposed a method to remove partial occlusion in color images, which does not estimate depth. Wanner and Goldluecke [15] proposed a globally consistent framework by applying structure tensors to estimate the directions of feature pixels in the 2D EPI. Yu et al. [28] explored geometric structures of 3D lines in ray space and encoded the line constraints to further improve the reconstruction quality. However, both methods are vulnerable to heavy occlusion: the tensor field becomes too random to estimate, and 3D lines are partitioned into small, incoherent segments. Kim et al. [35] adopted a fine-to-coarse framework to ensure smooth reconstructions in homogeneous areas using dense light fields. Jeon et al. [20] proposed a phase-based interpolation method to increase the accuracy for sub-pixel shift. We build on the method by Tao et al. [18], which works with consumer light field cameras, to improve depth estimation by taking occlusions into account. Although Tao et al. have a more recent method for depth estimation [36], it aims at combining the shading cue which is not applicable in our case.

Chen et al. [27] proposed a new bilateral metric on angular pixel patches to measure the probability of occlusions by their similarity to the central pixel. However, as noted in their discussion, their method is biased towards the central view as it uses the color of the central pixel as the mean of the bilateral filter. Therefore, the bilateral metric becomes unreliable once the input images get noisy. In contrast, our method uses the mean of about half the pixels as the reference, and is thus more robust when the input images are noisy, as shown in our result section.

3.3 Light-Field Occlusion Theory

We first develop our new light-field occlusion model, based on the physical image formation. We show that at occlusions, some of the angular patch remains photo-consistent, while the other part comes from occluders and exhibits no photo consistency. By treating these two regions separately, occlusions can be better handled.

For each pixel on an occlusion edge, we assume it is occluded by only one occluder among all views. We also assume that we are looking at a spatial patch small enough, so that the occlusion edge around that pixel can be approximated by a line. We show that if



(a) Pinhole model

(b) "Reversed" pinhole model

Figure 3.3: *Light-field occlusion model. (a) Pinhole model for central camera image formation. An occlusion edge on the imaging plane corresponds to an occluding plane in the 3D space. (b) The "reversed" pinhole model for light field formation. It can be seen that when we refocus to the occluded plane, we get a projection of the occluder on the camera plane, forming a reversed pinhole camera model.*

we refocus to the occluded plane, the angular patch will still have photo-consistency in a subset of the pixels (unoccluded). Moreover, the edge separating the unoccluded and occluded pixels in the angular patch has the same orientation as the occlusion edge in the spatial domain (Fig. 3.3). In Secs. 3.4 and 3.5, we use this idea to develop a depth estimation and regularization algorithm.

Consider a pixel at (x_0, y_0, f) on the imaging focal plane (the plane in focus), as shown in Fig. 3.3a. An edge in the central pinhole image with 2D slope γ corresponds to a plane P in 3D space (the green plane in Fig. 3.3a). The normal $\mathbf{n}$ to this plane can be obtained by taking the cross-product,

$$\mathbf{n} = (x_0, y_0, f) \times (x_0 + 1, y_0 + \gamma, f) = (-\gamma f, f, \gamma x_0 - y_0). \tag{3.1}$$

Note that we do not need to normalize the vector. The plane equation is $P(x, y, z) \equiv \mathbf{n} \cdot (x_0 - x, y_0 - y, f - z) = 0$,

$$P(x, y, z) \equiv \gamma f(x - x_0) - f(y - y_0) + (y_0 - \gamma x_0)(z - f) = 0. \tag{3.2}$$

In our case, one can verify that $\mathbf{n} \cdot (x_0, y_0, f) = 0$ so a further simplification to $\mathbf{n} \cdot (x, y, z) = 0$ is possible,

$$P(x, y, z) \equiv \gamma f x - f y + (y_0 - \gamma x_0)z = 0. \tag{3.3}$$

Now consider the occluder (yellow triangle in Fig. 3.3a). The occluder intersects $P(x, y, z)$ with $z \in (0, f)$ and lies on one side of that plane. Without loss of generality, we can assume it lies in the half-space $P(x, y, z) \geq 0$. Now consider a point $(u, v, 0)$ on the camera plane (the plane where the camera array lies on). To avoid being shadowed by the occluder, the line segment connecting this point and the pixel (x_0, y_0, f) on the image must not hit the occluder,

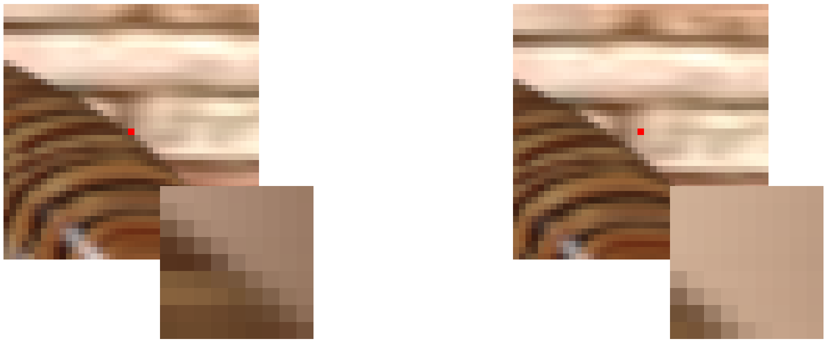
$$P(\mathbf{s}_0 + (\mathbf{s}_1 - \mathbf{s}_0)t) \leq 0 \quad \forall t \in [0, 1], \tag{3.4}$$

where $\mathbf{s}_0 = (u, v, 0)$ and $\mathbf{s}_1 = (x_0, y_0, f)$. When $t = 1$, $P(\mathbf{s}_1) = 0$. When $t = 0$,

$$P(\mathbf{s}_0) \equiv \gamma f u - f v \leq 0. \tag{3.5}$$

The last inequality is satisfied if $v \geq \gamma u$, i.e., the *critical slope on the angular patch $v/u = \gamma$ is the same as the edge orientation in the spatial domain*. If the inequality above is satisfied, both endpoints of the line segment lie on the other side of the plane, and hence the entire segment lies on that side as well. Thus, the light ray will not be occluded.

We also give an intuitive explanation of the above proof. Consider a plane being occluded by an occluder, as shown in Fig. 3.3b. Consider a simple 3×3 camera array. When we refocus to the occluded plane, we can see that some views are occluded by the occluder. Moreover, the occluded cameras on the camera plane are the projection of the occluder on the camera plane. Thus, we obtain a "reversed" pinhole camera model, where the original imaging plane is replaced by the camera plane, and the original pinhole becomes the pixel we are looking at. When we collect pixels from different cameras to form an angular patch, the edge separating the two regions will correspond to the same edge the occluder has in the spatial domain.



(a) Occlusion in central view

(b) Occlusion in other views

Figure 3.4: *Occlusions in different views. The insets are the angular patches of the red pixels when refocused to the correct depth. At the occlusion edge in the central view, the angular patch can be divided evenly into two regions, one with photo-consistency and one without. However, for pixels around the occlusion edge, although the central view is not occluded, some other views will still get occluded. Hence, the angular patch will not be photo-consistent, and will be unevenly divided into occluded and visible regions.*

Therefore, we can predict the edge orientation in the angular domain using the edge in the spatial image. Once we divide the patch into two regions, we know photo consistency holds in one of them since they all come from the same (assumed to be Lambertian) spatial pixel.

3.4 Occlusion-Aware Initial Depth

In this section, we show how to modify the initial depth estimation from Tao et al. [18], based on the theory above. First, we apply edge detection on the central view image. Then for each edge pixel, we compute initial depths using a modified photo-consistency constraint. The next section will discuss computation of refined occlusion predictors and regularization to generate the final depth map.

3.4.1 Edge detection

We first apply Canny edge detection on the central view (pinhole) image. Then an edge orientation predictor is applied on the obtained edges to get the orientation angles at each edge pixel. These pixels are candidate occlusion pixels in the central view. However, some pixels are not occluded in the central view, but are occluded in other views, as shown in Fig. 3.4, and we want to mark these as candidate occlusions as well. We identify these pixels by dilating the edges detected in the center view.

3.4.2 Depth Estimation

For each pixel, we refocus to various depths using a 4D shearing of the light-field data [2],

$$L_\alpha(x, y, u, v) = L(x + u(1 - \frac{1}{\alpha}), y + v(1 - \frac{1}{\alpha}), u, v), \tag{3.6}$$

where L is the input light field image, α is the ratio of the refocused depth to the currently focused depth, L_α is the refocused light field image, (x, y) are the spatial coordinates, and (u, v) are the angular coordinates. The central viewpoint is located at $(u, v) = (0, 0)$. This gives us an angular patch for each depth, which can be averaged to give a refocused pixel. In our implementation, we use a simple linear interpolation to perform the resampling in Eq. 3.6. However, more advanced resampling techniques, e.g. the phase-based method in [20], could be used and could potentially lead to better results.

When an occlusion is not present at the pixel, the obtained angular patch will have photo-consistency, and hence exhibits small variance and high similarity to the central view. For pixels that are not occlusion candidates, we can simply compute the variance and the mean of this patch to obtain the correspondence and defocus cues, similar to the method by Tao et al. [18].

However, if an occlusion occurs, photo-consistency will no longer hold. Instead of dealing with the entire angular patch, we divide the patch into two regions. The angular edge orientation separating the two regions is the same as in the spatial domain, as proven in Sec. 3.3. Since at least half the angular pixels come from the occluded plane (otherwise it will not be seen in the central view), we place the edge passing through the central pixel, dividing the patch evenly. Note that only one region, corresponding to the partially occluded plane focused to the correct depth, exhibits photo-consistency. The other region contains angular pixels that come from the occluder, which is not focused at the proper depth, and might also contain some pixels from the occluded plane. We therefore replace the original patch with the region that has the minimum variance to compute the correspondence and defocus cues.

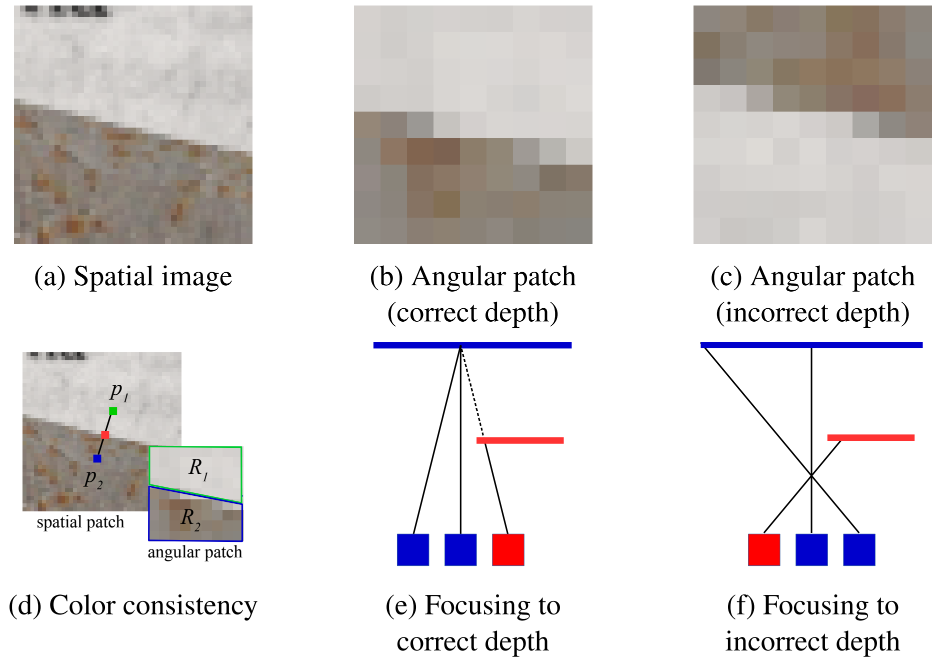
To be specific, let (u_1, v_1) and (u_2, v_2) be the angular coordinates in the two regions, respectively. We first compute the means and the variances of the two regions,

$$\bar{L}_{\alpha,j}(x, y) = \frac{1}{N_j} \sum_{u_j, v_j} L_\alpha(x, y, u_j, v_j), \quad j = 1, 2 \tag{3.7}$$

$$V_{\alpha,j}(x, y) = \frac{1}{N_j - 1} \sum_{u_j, v_j} \left(L_\alpha(x, y, u_j, v_j) - \bar{L}_{\alpha,j}(x, y)\right)^2, \tag{3.8}$$

where N_j is the number of pixels in region j. Let

$$i = \arg\min_{j=1,2} \left\{V_{\alpha,j}(x, y)\right\} \tag{3.9}$$



(a) Spatial image

(b) Angular patch (correct depth)

(c) Angular patch (incorrect depth)

(d) Color consistency

(e) Focusing to correct depth

(f) Focusing to incorrect depth

Figure 3.5: *Color consistency constraint. (b)(e) We can see that when we refocus to the correct depth, we get low variance in half the angular patch. However, in (c)(f) although we refocused to an incorrect depth, it still gives low variance response since the occluded plane is very textureless, so we get a "reversed" angular patch. To address this, we add another constraint that p_1 and p_2 should be similar to the averages of R_1 and R_2 in (d), respectively.*

be the index of the region that exhibits smaller variance. Then the correspondence response is given by

$$C_\alpha(x,y) = V_{\alpha,i}(x,y) \tag{3.10}$$

Similarly, the defocus response is given by

$$D_\alpha(x,y) = \left(\bar{L}_{\alpha,i}(x,y) - L(x,y,0,0)\right)^2 \tag{3.11}$$

Finally, the optimal depth is determined as

$$\alpha^*(x,y) = \arg\min_\alpha \left\{C_\alpha(x,y) + D_\alpha(x,y)\right\} \tag{3.12}$$

3.4.3 Color Consistency Constraint

When we divide the angular patch into two regions, it is sometimes possible to obtain a "reversed" patch when we refocus to an incorrect depth, as shown in Fig. 3.5. If the occluded plane is very textureless, this depth might also give a very low variance response, even though it is obviously incorrect. To address this, we add a color consistency constraint that the averages of the two regions should have a similar relationship with respect to the current pixel as they have in the spatial domain. Mathematically,

$$|\bar{L}_{\alpha,1} - p_1| + |\bar{L}_{\alpha,2} - p_2| < |\bar{L}_{\alpha,2} - p_1| + |\bar{L}_{\alpha,1} - p_2| + \delta, \tag{3.13}$$

where p_1 and p_2 are the values of the pixels shown in Fig. 3.5d, and δ is a small value (threshold) to increase robustness. If refocusing to a depth violates this constraint, this depth is considered invalid, and is automatically excluded in the depth estimation process.

3.5 Occlusion-Aware Depth Regularization

After the initial local depth estimation phase, we refine the results with global regularization using a smoothness term. We improve on previous methods by reducing the effect of the smoothness/regularization term in occlusion regions. Our occlusion predictor, discussed below, may also be useful independently for other vision applications.

3.5.1 Occlusion Predictor Computation

We compute a predictor P_{occ} for whether a particular pixel is occluded, by combining cues from depth, correspondence and refocus.

Depth Cues

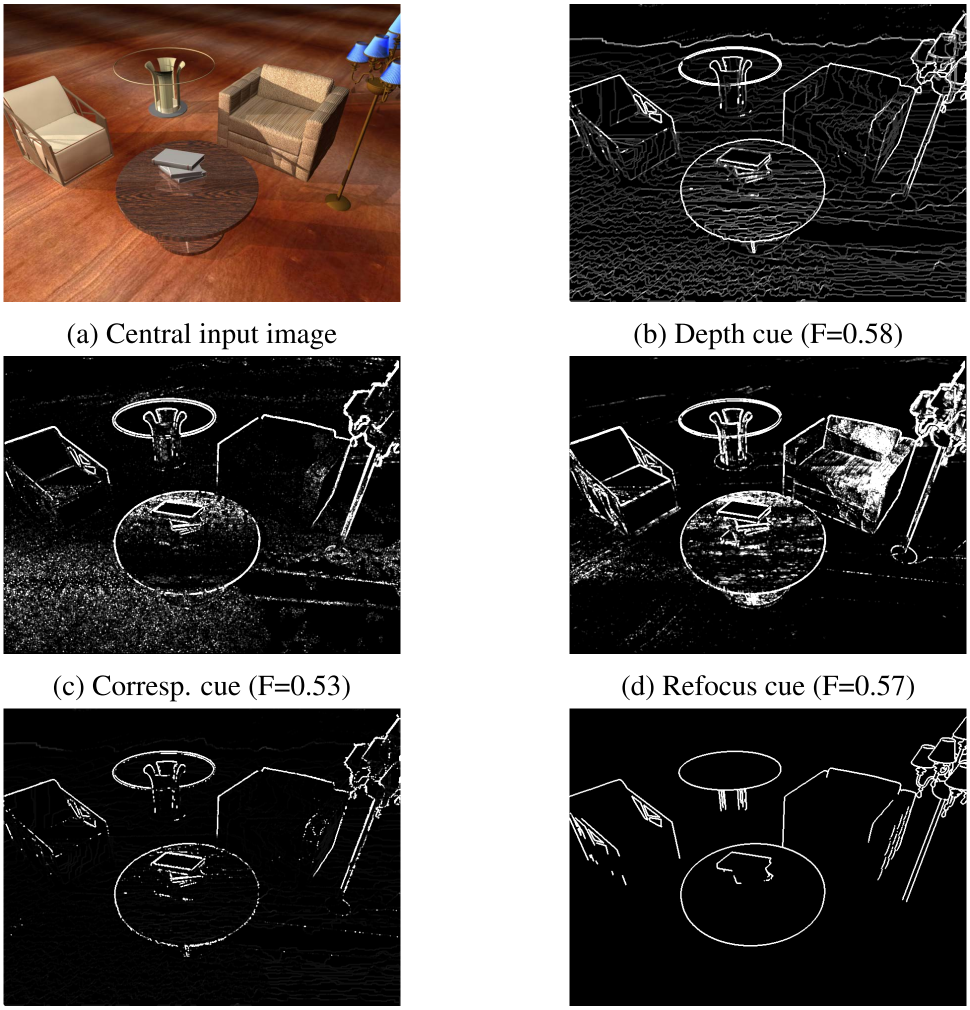
First, by taking the gradient of the initial depth, we can obtain an initial occlusion boundary,

$$P_{\text{occ}}^d = f\big(\nabla d_{\text{ini}} / d_{\text{ini}}\big) \tag{3.14}$$

where d_{ini} is the initial depth, and $f(\cdot)$ is a robust clipping function that saturates the response above some threshold. We divide the gradient by d_{ini} to increase robustness since for the same normal, the depth change across pixels becomes larger as the depth gets larger.

Correspondence Cues

In occlusion regions, we have already seen that photo-consistency will only be valid in approximately half the angular patch, with a small variance in that region. On the other hand, the pixels in the other region come from different points on the occluding object,



(a) Central input image

(b) Depth cue (F=0.58)

(c) Corresp. cue (F=0.53)

(d) Refocus cue (F=0.57)

(e) Combined cue (F=0.65)

(f) Occlusion ground truth

Figure 3.6: *Occlusion Predictor (Synthetic Scene). The intensities are adjusted for better contrast. F-measure is the harmonic mean of precision and recall compared to the ground truth. By combining three cues from depth, correspondence and refocus, we can obtain a better prediction of occlusions.*

and thus exhibit much higher variance. By computing the ratio between the two variances, we can obtain an estimate of how likely the current pixel is to be at an occlusion,

$$P_{\text{occ}}^{\text{var}} = f\left(\max\left\{\frac{V_{\alpha^*,1}}{V_{\alpha^*,2}}, \frac{V_{\alpha^*,2}}{V_{\alpha^*,1}}\right\}\right), \tag{3.15}$$

where α^* is the initial depth we get.

Refocus Cues

Finally, note that the variances in both the regions will be small if the occluder is textureless. To address this issue, we also compute the means of both regions. Since the two regions come from different objects, their colors should be different, so a large difference between the two means also indicates a possible occlusion occurrence. In other words,

$$P_{\text{occ}}^{\text{avg}} = f(|\bar{L}_{\alpha^*,1} - \bar{L}_{\alpha^*,2}|) \tag{3.16}$$

Finally, we compute the combined occlusion response or prediction by the product of these three cues,

$$P_{occ} = \mathcal{N}(P_{\text{occ}}^{\text{d}}) \cdot \mathcal{N}(P_{\text{occ}}^{\text{var}}) \cdot \mathcal{N}(P_{\text{occ}}^{\text{avg}}) \tag{3.17}$$

where $\mathcal{N}(\cdot)$ is a normalization function that subtracts the mean and divides by the standard deviation. The threshold values of the f function for depth, correspondence and refocus cues are set to 1, 100, and 0.01, respectively.

3.5.2 Depth Regularization

Finally, given initial depth and occlusion cues, we regularize with a Markov Random Field (MRF) for a final depth map. We minimize the energy:

$$E = \sum_{p} E_{\text{unary}}(p, d(p)) + \lambda \sum_{p,q} E_{\text{binary}}(p, q, d(p), d(q)). \tag{3.18}$$

where d is the final depth p, q are neighboring pixels, and λ is a weight which we set to 5. We adopt the unary term similar to Tao et al. [18]. The binary energy term is defined as

$$E_{binary}(p, q, d(p), d(q)) = \frac{\exp\left[- (d(p) - d(q))^2/(2\sigma^2)\right]}{(|\nabla I(p) - \nabla I(q)| + k|P_{occ}(p) - P_{occ}(q)|)} \tag{3.19}$$

where ∇I is the gradient of the central pinhole image, and k is a weighting factor. The numerator encodes the smoothness constraint, while the denominator reduces the strength of the constraint if two pixels are very different or an occlusion is likely to be between them. The minimization is solved using a standard graph cut algorithm [37, 38, 39]. We can then apply the occlusion prediction procedure again on this regularized depth map. A sample result is shown in Fig. 3.6. In this example, the F-measure (harmonic mean of precision and recall compared to ground truth) increased from 0.58 (depth cue), 0.53 (correspondence cue), and 0.57 (refocus cue), to 0.65 (combined cue).

3.6 Results

In this section, we first show results of different stages of our algorithm (Sec. 3.6.1), and then demonstrate the superiority of our method by comparing to different state-of-the-art algorithms (Sec. 3.6.2). Finally, we show limitations and some failure cases of our method (Sec. 3.6.3).

3.6.1 Algorithm Stages

We show results of different stages of our algorithm in Fig. 3.7. First, edge detection is applied on the central pinhole image (Fig. 3.7a) to give all possible edge boundaries (Fig. 3.7b). As can be seen, although the output captures the occlusion boundaries, it also contains lots of false positives. We then compute the initial depth (Fig. 3.7c) and occlusion prediction (Fig. 3.7d) using the method described in Sec. 3.4. Note that the false positives in the obtained occlusion are dramatically reduced. Finally, using the initial depth and occlusion detection, we further regularize the depth (Sec. 3.5) to get the final depth (Fig. 3.7e) and occlusion detection (Fig. 3.7f). Note that the final occlusion detection realistically captures the true occlusion boundaries. For runtime, on a 2.4 GHz Intel i7 machine with 8GB RAM, our MATLAB implementation takes about 3 minutes on a Lytro Illum Image (7728×5368 pixels). This is comparable to [18], since all the additional steps are marginal to the computation.

3.6.2 Comparisons

We compare our results to the methods by Wanner et al. [15], Tao et al. [18], Yu et al. [28], Chen et al. [27], Jeon et al. [20], and the results by Lytro Illum. For Chen et al., since code is not available, we used our own implementation. Since ground truth at occlusions is difficult to obtain, we perform extensive tests using the synthetic dataset created by Wanner et al. [40] as well as new scenes modeled by us. Our dataset is generated from 3dsMax [41] using models from the Stanford Computer Graphics Laboratory [42, 43, 44] and models freely available online [45]. The dataset can be found online at http://cseweb.ucsd.edu/~viscomp/projects/LF/papers/ICCV15/dataset.zip. While the dataset by [40] only provides ground truth depth, ours provides ground truth depth, normals, specularity, lighting, etc, which we believe will be useful for a wider variety of applications. In addition to synthetic datasets, we also validate our algorithm on real-world scenes of fine objects with occlusions, taken by the Lytro Illum camera.

Occlusion Boundaries

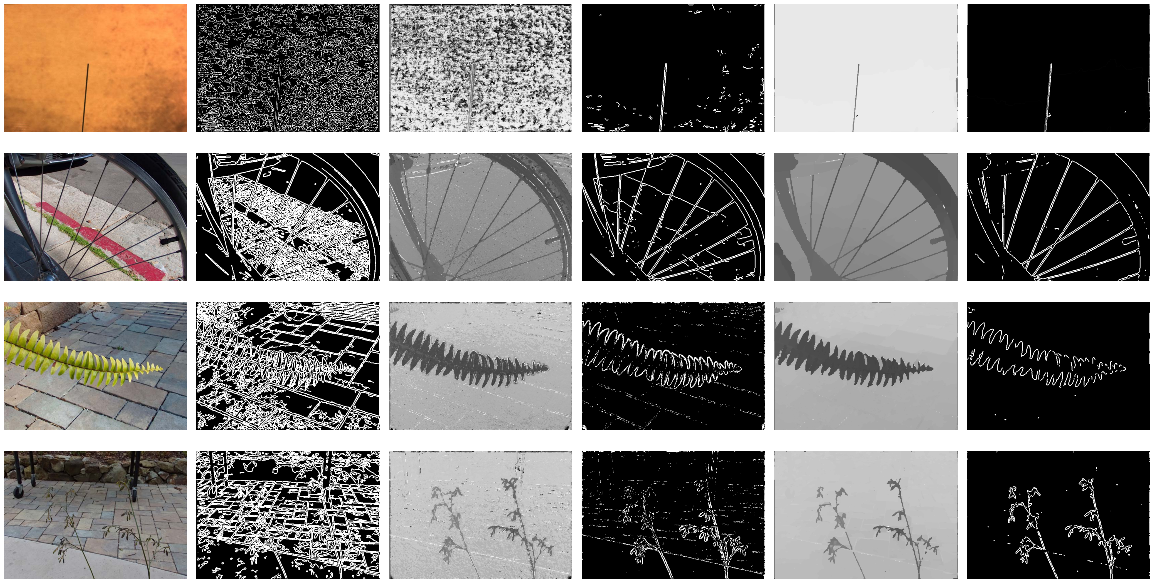
For each synthetic scene, we compute the occlusion boundaries from the depth maps generated by each algorithm, and report their precision-recall curves by picking different thresholds. For our method, the occlusions are computed using only the depth cue instead of the combined cue in Sec. 3.5, to compare the depth quality only. A predicted occlusion pixel is considered correct if its error is within one pixel. The results on both synthetic datasets are shown in Figs. 3.8a,3.8b. Our algorithm achieves better performance than current state-of-the-art methods. Next, we validate the robustness of our system by adding noise to a test image, and report the F-measure values of each algorithm, as shown in Fig. 3.8c. Although Chen et al. [27] performs very well in the absence of noise, their quality quickly degrades as the noise level is increased. In contrast, our algorithm is more tolerant to noise.

Depth Maps for Synthetic Scenes

Figure 3.9 shows the recovered depths on the synthetic dataset by Wanner et al. [40]. It can be seen that our results show fewer artifacts in heavily occluded areas. We obtain the correct shape of the door and window in the top row, accurate boundaries along the twig and leaf in the second row, and realistic antenna shape and wing boundaries in the bottom row. Other methods smooth the object boundaries and are noisy in some regions. Figure 3.10 shows the results on our synthetic dataset. Notice that we capture the boundaries of the leaves, fine structures like the lamp and holes in the chair, and thin shapes of the lamp and the chandelier. Other methods smooth over these occlusions or generate thicker structures. The RMSE of the depth maps compared to the ground truth are also shown in Table 3.1. However, note that RMSE is not the best metric for the improvements on thin occluded structures provided by our method.

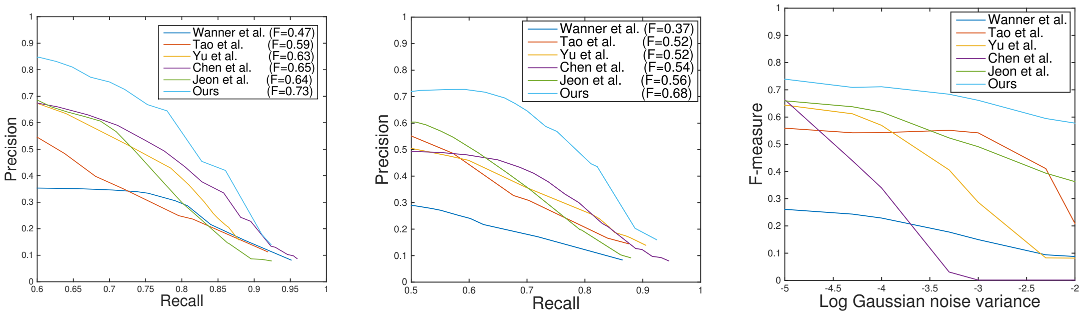
Depth Maps for Real Scenes

Figures 5.1 and 3.11 compare results on real scenes with fine structures and occlusions, captured with Lytro Illum light field camera. Our method performs better around occlusion boundaries, especially for thin objects. Ours is the only method that captures the basket holes in Fig. 5.1. In Fig. 3.11, our method properly captures the thin structure of the lead (first row), reproduces the fine petals of the flower (second row), captures the holes behind the leafs without over-smoothing (third and fourth row), obtains realistic shape of the stem(fifth row), and reproduces the complicated structure of the strap (final row).



(a) Input (b) Edge detect (c) Initial depth (d) Initial occ (e) Final depth (f) Final occ

Figure 3.7: *Real-world results of different stages of our algorithm. We first apply edge detection on the central input, run our depth estimation algorithm on the light-field image to get an initial depth and an occlusion response prediction, and finally use the occlusion to regularize the initial depth to get a final depth map. We can then run the occlusion predictor on this final depth again to get a refined occlusion.*



(a) PR curve (dataset by Wanner) (b) PR curve (our dataset) (c) F-measure vs. noise

Figure 3.8: *(a) PR-curve of occlusion boundaries on dataset of Wanner et al. [40] (b) PR-curve on our dataset. (c) F-measure vs. noise level. Our method achieves better results than current state-of-the-art methods, and is robust to noise.*

	Wanner et al.	Tao et al.	Yu et al.	Chen et al.	Jeon et al.	Our method
Dataset by Wanner	0.0470	0.0453	0.0513	0.0375	0.0443	***0.0355***
Our dataset	0.1256	0.1253	0.1006	0.1019	0.1062	***0.0974***

Table 3.1: *Depth RMSE on synthetic scenes. Our method achieves lowest RMSE on both datasets. Note that RMSE is not the best metric for the improvements on thin occluded structures provided by our method.*

3.6.3 Limitations and Future Work

First, our algorithm cannot handle situations where the occluded plane is very small relative to the angular patch size, or if the single occluder assumption fails to hold (Fig. 3.12). If the occluded area is very small, there is no simple line that can separate the angular patch into two regions. If we have multiple edges intersecting at a point, its angular patch needs to be divided into more than two regions to achieve photo consistency. This may be addressed by inspecting the spatial patch around the current pixel instead of just looking at the edges. Second, our algorithm cannot perform well if the spatial edge detector fails or outputs an inaccurate orientation. We also assume the light-field is bandlimited [46], so aliasing does not occur and we can always find consistent correspondences in the original light-field representation. Finally, similar to previous stereo methods, our algorithm cannot perform well at textureless regions. In addition, since we only use half the angular patch around edges, it might introduce some confusion in certain cases. For example, a special case would be a plane which is uniform on one side and textured on the other side. Using previous methods, the depth around the separating edge can be uniquely determined using the entire angular patch. However, no matter which depth we refocus to, the angular patch will be uniform on one side, and our method will not be able to find the correct depth. In this case, the unary cost will be indiscernible, and we will rely on neighboring pixels in the textured region to determine its depth (by smoothness constraint), just as previous methods rely on neighboring pixels to determine the depths in uniform regions.

3.7 Conclusion

In this chapter, we propose an occlusion-aware depth estimation algorithm. We show that although pixels around occlusions do not exhibit photo-consistency in the angular patch when refocused to the correct depth, they are still photo-consistent for part of the patch. Moreover, the line separating the two regions in the angular domain has the same orientation as the edge in the spatial domain. Utilizing this information, the depth estimation process can be improved in two ways. *First*, we can enforce photo-consistency on only the region that is coherent. *Second*, by exploiting depth, correspondence and refocus cues, we can perform occlusion prediction, so smoothing over these boundaries can be avoided in the regularization. We demonstrate the benefits of our algorithm on various synthetic datasets as well as real-world images with fine structures, extending the range of objects that can be captured in 3D with consumer light-field cameras.

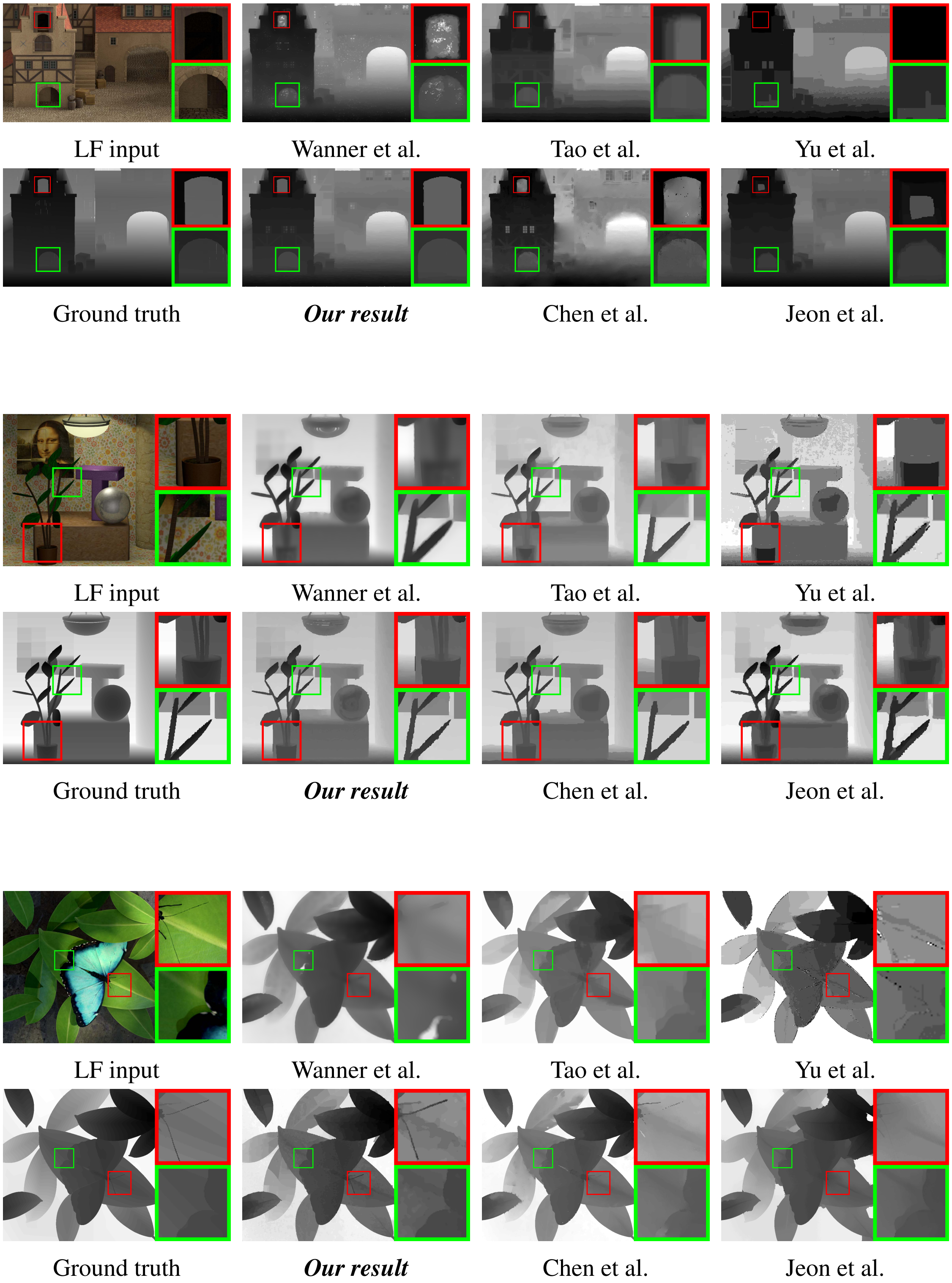
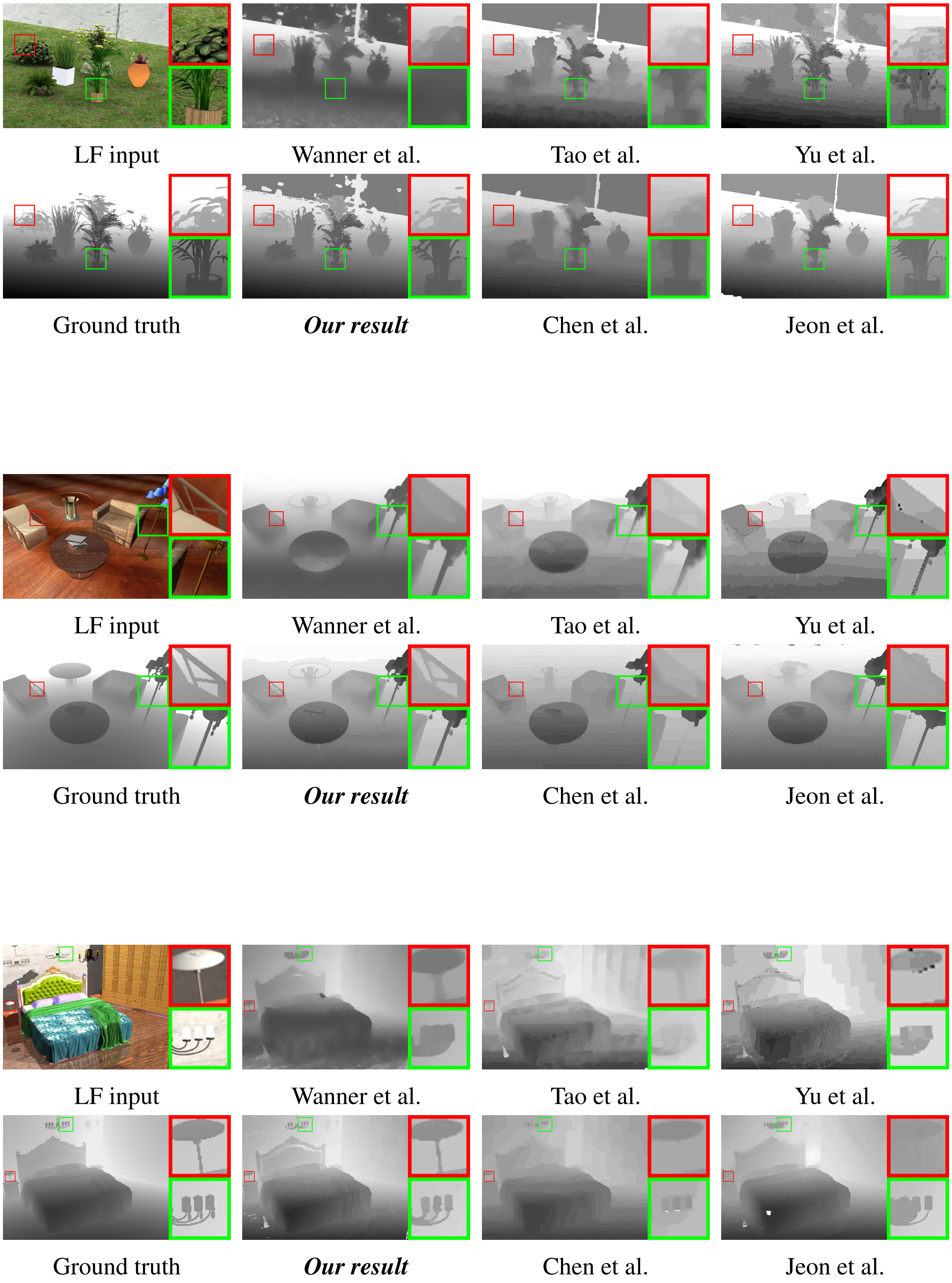


Figure 3.9: *Depth estimation results on synthetic data by Wanner et al. [40]. Some intensities in the insets are adjusted for better contrast. In the first example, note that our method correctly captures the shape of the door/window, while all other algorithms fail and produce smooth transitions. Similarly, in the second example our method reproduces accurate boundaries along the twig/leaf, while other algorithms generate smoothed results or fail to capture the details, and have artifacts. Finally, in the last example, our*



LF input Wanner et al. Tao et al. Yu et al.

Ground truth ***Our result*** Chen et al. Jeon et al.

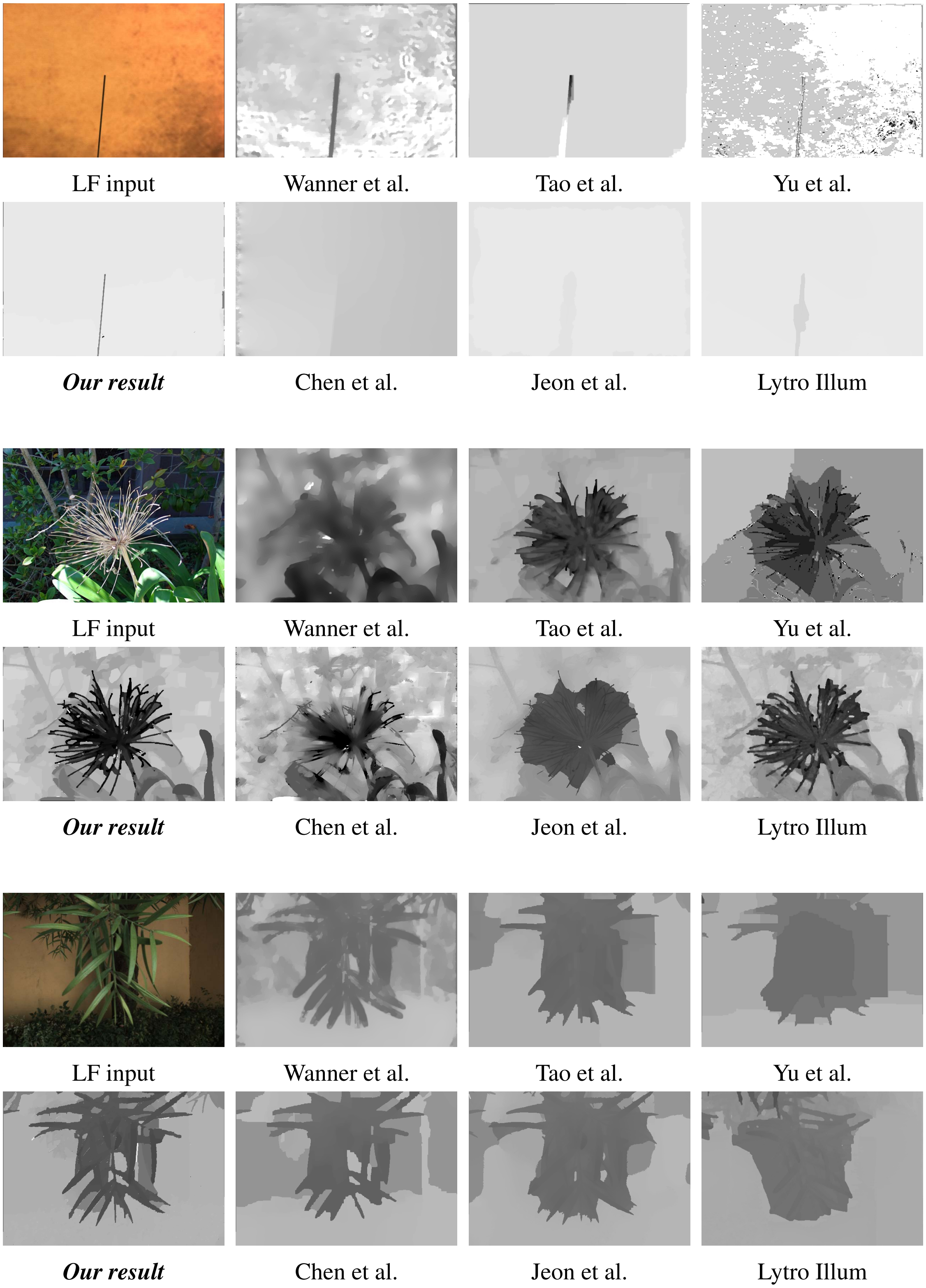
LF input Wanner et al. Tao et al. Yu et al.

Ground truth ***Our result*** Chen et al. Jeon et al.

LF input Wanner et al. Tao et al. Yu et al.

Ground truth ***Our result*** Chen et al. Jeon et al.

Figure 3.10: *Depth estimation results on our synthetic dataset. Some intensities in the insets are adjusted for better contrast. In the first example, our method successfully captures the shapes of the leaves, while all other methods generate smoothed results. In the second example, our method captures the holes in the chair as well as the thin structure of the lamp, while other methods obtain smoothed or thicker structures. In the last example, our method captures the thin structure of the lamp and the chandelier, while other*



LF input Wanner et al. Tao et al. Yu et al.

Our result Chen et al. Jeon et al. Lytro Illum

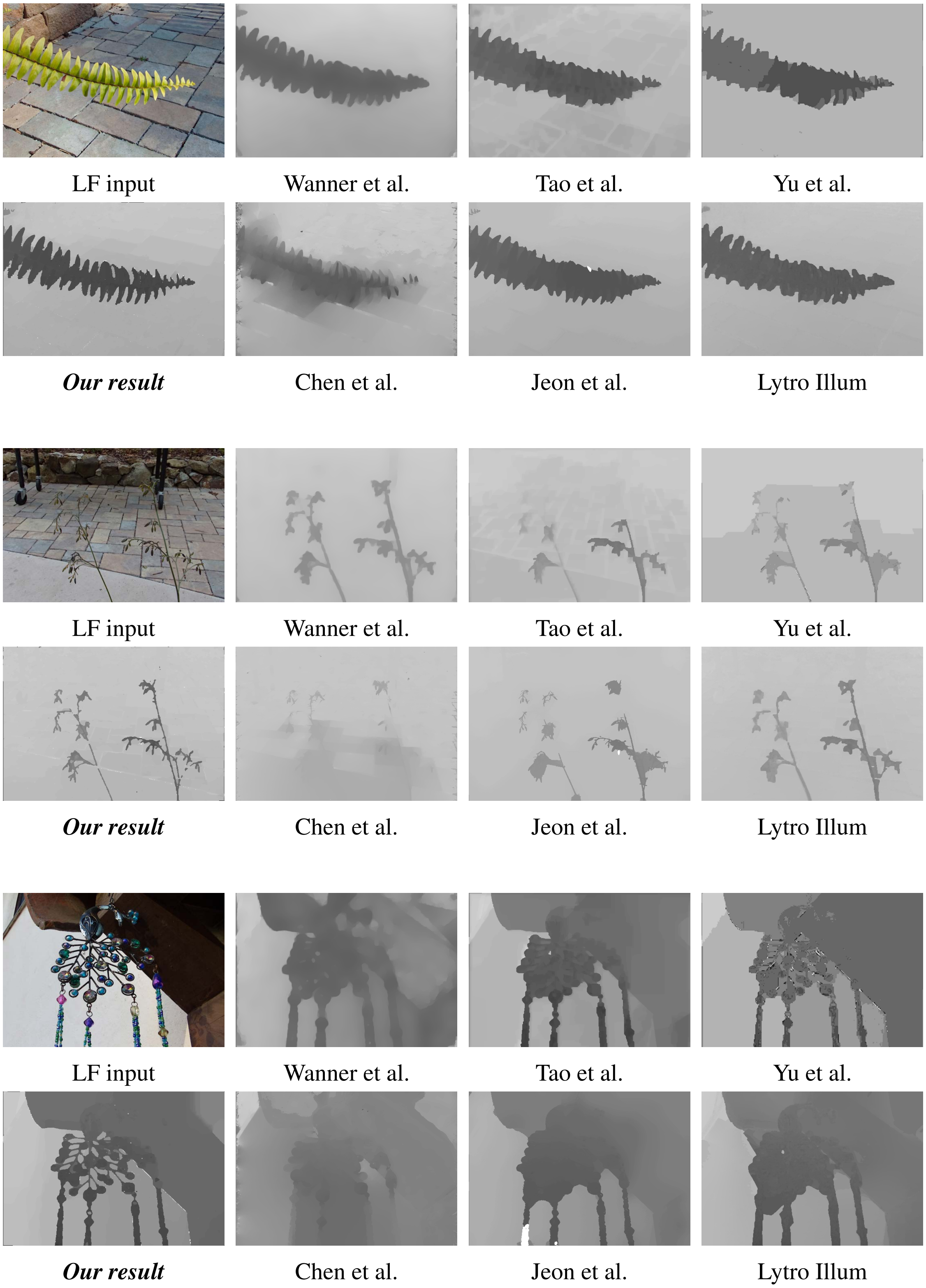
LF input Wanner et al. Tao et al. Yu et al.

Our result Chen et al. Jeon et al. Lytro Illum

LF input Wanner et al. Tao et al. Yu et al.

Our result Chen et al. Jeon et al. Lytro Illum

Figure 3.11: *Depth estimation results on real data taken by the Lytro Illum light field camera. It can be seen that our method realistically captures the thin structures and occlusion boundaries, while other methods fail, or generate dilated structures.*



LF input Wanner et al. Tao et al. Yu et al.

Our result Chen et al. Jeon et al. Lytro Illum

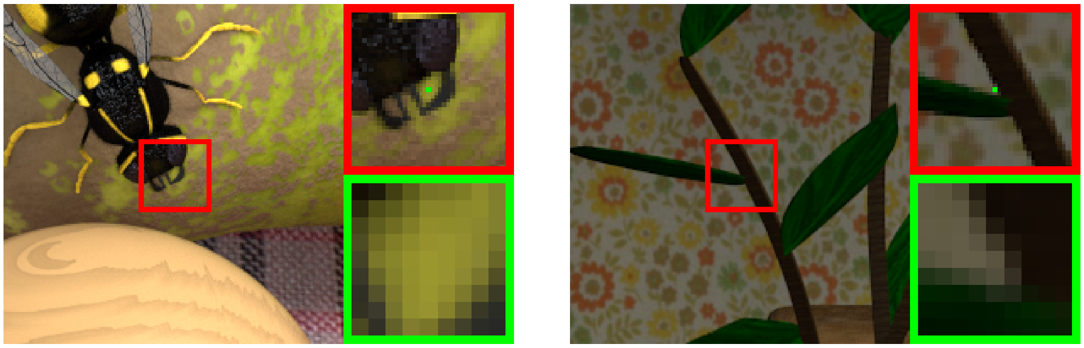
LF input Wanner et al. Tao et al. Yu et al.

Our result Chen et al. Jeon et al. Lytro Illum

LF input Wanner et al. Tao et al. Yu et al.

Our result Chen et al. Jeon et al. Lytro Illum

Figure 3.11: *Depth estimation results on real data taken by the Lytro Illum light field camera (continued). It can be seen that our method realistically captures the thin structures and occlusion boundaries, while other methods fail, or generate dilated structures.*



(a) Small area occlusion

(b) Multi-occluder occlusion

Figure 3.12: *Limitations. The upper insets show close-ups of the red rectangle, while the lower insets show the angular patches of the green (central) pixels when refocused to the correct depth. (a) Our algorithm cannot handle occlusions where the occluded area is very small, so that there is no simple line that can separate the angular patch. (b) Also, if more than one occluder is present around the pixel, it is not enough to just divide the angular domain into two regions.*

Chapter 4

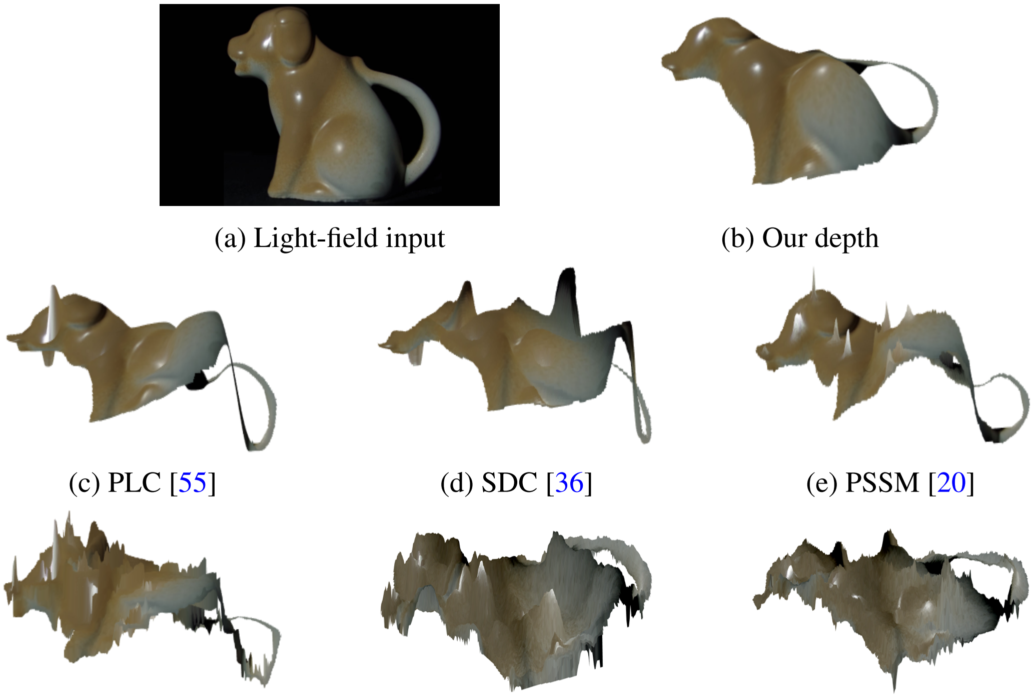
SVBRDF-Invariant Shape and Reflectance Estimation

We have seen how we can estimate a better depth map around occlusions in the previous chapter. However, note that so far we have only focused on Lambertian objects, and many real-world objects are actually glossy, such as metals, plastics, and paper. Obtaining the shape of these glossy objects remains challenging, since the photo-consistency assumption cannot be applied again.

In this chapter, instead of dealing with occlusions, we try to handle this glossy surface problem. We derive a spatially-varying (SV)BRDF-invariant theory for recovering 3D shape and reflectance from light-field cameras. Our key theoretical insight is a novel analysis of diffuse plus single-lobe SVBRDFs under a light-field setup. We show that, although direct shape recovery is not possible, an equation relating depths and normals can still be derived. Using this equation, we then propose using a polynomial (quadratic) shape prior to resolve the shape ambiguity. Once shape is estimated, we also recover the reflectance. We present extensive synthetic data on the entire MERL BRDF dataset, as well as a number of real examples to validate the theory, where we simultaneously recover shape and BRDFs from a single image taken with a Lytro Illum camera.

4.1 Introduction

Using motions of the object, the light source or the camera to recover object shapes have been extensively studied in computer vision. For example, many works have been developed in optical flow for exploiting object motion [47, 48], photometric stereo for light source motion [49] and multi-view stereo for camera motion [50]. However, dealing with the complex behavior of the bidirectional reflectance distribution function (BRDF)



(a) Light-field input

(b) Our depth

(c) PLC [55]

(d) SDC [36]

(e) PSSM [20]

(f) Lytro Illum

(g) SMRM [52, 53]

(h) IAMO [54]

Figure 4.1: *Comparison of depth estimation results of different algorithms. We texture map the input image onto the depth maps, so we can clearly see where each method fails. It can be seen that our method correctly handles the glossy surface, while other methods generate visible artifacts, especially around the specular parts.*

is hard, and the photo-consistency assumption is often adopted assuming a Lambertian surface. In particular, very robust and efficient algorithms have been introduced in multi-view stereo based on diffuse brightness constancy [51]. However, many common materials such as metals, plastics or ceramics are not diffuse and do not follow these assumptions, so acquiring shapes for these materials is still a difficult problem. Although theories for recovering shapes with general BRDFs have been proposed by Chandraker [52, 53, 54], they are still not as robust compared to traditional Lambertian methods, and the setup requires multiple shots of the object and is thus inconvenient.

In recent years, light-fields have emerged as a powerful tool for shape recovery. Using light-field cameras (e.g. Lytro [21] and Raytrix [22]), shape can be recovered in a single shot with minimal effort, offering a practical and convenient alternative to traditional multi-view approaches. However, most current depth estimation methods still support only Lambertian scenes, making them unreliable for glossy surfaces.

In this chapter, we present a depth and reflectance estimation algorithm that explicitly models spatially varying BRDFs (SVBRDFs) from light-field cameras (Fig. 5.1) [56, 57]. Since the problem is under-constrained, we assume a known distant light source. We think of a light-field camera as a multi-camera array (of virtual viewpoints), and follow the shape estimation framework using camera motion in [52, 53, 54]. However, note that the theory in [52, 53, 54] is not directly applicable to the light-field case; in fact, we show that in the case of light-fields, shape cannot be directly recovered (Sec. 4.3). However, in many instances where the BRDF depends on only the half-angle, we derive an SVBRDF-invariant equation relating depths and normals (Sec. 4.4). Note that we are able to include a generalized diffuse term (including textures) in addition to the specular single-lobe model, and that our theory applies generally to *spatially-varying* BRDFs (Figs. 4.7 and 4.12), whereas the work by Chandraker [52, 53, 54] was limited to homogeneous materials.

After this equation is derived, we recover the shape by applying a locally polynomial shape prior (Sec. 4.5.1). To ease the optimization, we require the normal at one seed pixel to be specified. Then, we solve for the BRDF derivatives and integrate them to recover the reflectance (Sec. 4.5.2). Finally, we demonstrate extensive real-world examples of shape and reflectance estimation using commercial light-field cameras (Figs. 5.1, 4.11, and 4.12). Our main contributions are:

1) A generalization of optical flow to the non-Lambertian case in light-field cameras (Secs. 4.3 and 4.4).

2) A depth estimation algorithm for light-field cameras that handles diffuse plus specular 1-lobe BRDFs (Sec. 4.5.1).

3) A reflectance estimation approach that recovers BRDFs for up to 2-lobes once shape is given (Sec. 4.5.2).

4) An extensive synthetic evaluation on the entire MERL BRDF dataset [58] (Sec. 4.6, Figs. 4.7,4.9 and 4.10).

5) A practical realization of our algorithm on images taken with the Lytro Illum camera (Sec. 4.6).

4.2 Related Work

Depth from Light-Field Cameras: Many depth estimation methods for light-field cameras have been proposed. However, most of them rely on the Lambertian assumption and work poorly on glossy surfaces [27, 20, 35, 36, 25, 26, 15]. Recently, there are some works that try to deal with specularity. Tao et al. [59] proposed a clustering method that eliminates specular pixels when enforcing photo consistency. However, they attempt a binary classification of pixels into either Lambertian or specular, which cannot handle general glossy surfaces. A follow-up work [55] adopts the dichromatic model and combines point and line consistency to deal with Lambertian and specular surfaces respectively. However, the dichromatic model fails to hold for materials like metals [60]. Therefore, their method fails if the BRDFs in different views do not lie on a line as in the dichromatic model, which is discussed in Sec. 4.4.1. Moreover, line consistency is not robust if neighboring pixels have a similar color. In contrast, our model can work on general 1-lobe BRDFs, and can also recover reflectance in addition to shape (Sec. 4.5.2), which has not been achieved by previous light-field shape acquisition approaches.

Multi-View Stereo: Multi-view stereo methods based on diffuse photo-consistency have a long history [50]. In recent years, the robustness for these reconstruction algorithms have been dramatically improved [51]. Several extensions have also been proposed to handle severe situations such as textureless regions or specularity, including priors from Manhattan constraints [61, 62] or architectural schema [63]. In contrast, we explicitly account for SVBRDF-dependence in image formation for shape recovery under the light-field setup, which can be considered as differential translations of the camera.

Methods dealing with non-Lambertian materials have also been introduced. For instance, the Helmholtz reciprocity principle is adopted by Zickler et al. [64] to reconstruct shapes with arbitrary BRDFs. Bonfort and Sturm [65] use a voxel carving approach to handle specular surfaces. Yang et al. [66] extend the Space Carving framework with a photo-consistency measure that works for both specular and diffuse surfaces. Treuille et al. [67] present an example-based stereo approach that uses reference shapes of known geometry to infer the unknown shapes. Jin et al. [68] derive a rank constraint on the radiance tensor field to estimate the surface shape and the appearance. Yu et al. [69] reduce biases in 3D reconstruction using a new iterative graph cut algorithm based on Surface Distance Grid. In contrast, we explore how a light-field image informs about shape with unknown SVBRDFs, regardless of the reconstruction method.

Differential Motion Theory: Our theoretical contributions are most closely related to the differential theory proposed by Chandraker [52, 53, 54]. He constructs a mathematical model to recover depth and reflectance using differential camera motion or object motion. Our work has three major differences. *First*, in contrast to the differential motions he uses, which contain both translations and rotations, we only have translations in light-field cameras. While this changes the form of equations obtainable through differential motions, we show that a BRDF-invariant equation of similar form as in [52, 53, 54] can still be obtained for half-angle BRDFs (Sec. 4.4). *Second*, the work by Chandraker then assumes a constant viewing direction (i.e., $(0, 0, -1)^\top$) for all pixels to solve for depth directly. In contrast, for our purely translational light-field setup, we must account for viewpoint variations. This is necessary because if the view directions do not differ between cameras, it inherently implies photo-consistency in the Lambertian case. As we show, accounting for viewpoint changes results in the infeasibility to directly obtain depth, and we try to solve the BRDF-invariant equation by applying a polynomial shape prior instead (Sec. 4.5.1). *Finally*, to obtain depth directly Chandraker also assumes a homogeneous BRDF. Since we are solving the BRDF-invariant equation instead of computing depth directly, this change also enables us to deal with spatially-varying BRDFs.

BRDF Estimation: BRDF estimation has been studied for many years and different models have been proposed [70]. Parametric models [71] can achieve good accuracy by modeling the BRDF as a statistical distribution on the unit sphere. Non-parametric [72, 73] and data-driven methods [58] are also popular, but rely on complex estimation or require a large amount of data. Semi-parametric approaches [74, 75] have also been proposed.

For joint shape and BRDF estimation, the closest to our work is [52] described above. Alldrin et al. [76] proposed an alternating approach to recover both shape and BRDF under light source motion. The work by Oxholm and Nishino [77] also uses an alternating optimization over shape and reflectance under natural illumination. None of these methods tries to recover shape or reflectance using camera motions, and the techniques are not intended for light-field cameras.

Shape from Shading: Shape from shading has a long history. Since it is a very under-constrained problem, most work assumes a known light source to increase feasibility [78, 79]. The method by Johnson and Adelson [80] can estimate shape under natural illumination, but requires a known reflectance map, which is hard to obtain. Barron and Malik [81, 82] described a framework to recover shape, illumination, reflectance, and shading from an image, but many constraints are needed for both geometry and illumination. Since shape from shading is usually prone to noise, recent methods [83, 84] assumed that the shape is locally polynomial for a small patch, and thus increased robustness. We adopt this strategy in our final optimization procedure. However, note that our case is harder, since most shape from shading methods are limited to Lambertian surfaces. In the Lambertian case, if both the pixel value and the light source are given, the normal must be lying on a cone around the light direction. In our case, since the BRDF is an unknown function, we do not have this condition.

4.3 Differential Stereo

Since light-field cameras can be considered as a multi-camera array corresponding to the set of virtual viewpoints, we first consider a simple two-camera case in Sec. 4.3.1. The idea is then extended to a multi-camera array in Sec. 4.3.2. Finally, the BRDF invariant equation is derived in Sec. 4.4.

4.3.1 Two-camera System

Consider a camera in the 3D spatial coordinates, where the origin is the principal point of its image plane. The camera is centered at $\mathbf{p} = (0, 0, -f)^\top$, where f is the focal length of the camera. Let $\beta \equiv 1/f$. Then for a perspective camera, a 3D point $\mathbf{x} = (x, y, z)^\top$ is imaged at pixel $\mathbf{u} = (u, v)^\top$, where

$$u = \frac{x}{1 + \beta z}, v = \frac{y}{1 + \beta z}. \tag{4.1}$$

Let $\mathbf{s}$ be the known distant light source direction. Given a 3D point $\mathbf{x}$, let $\mathbf{n}$ be its corresponding normal, and $\mathbf{v}$ be its (unnormalized) viewing direction from the camera center, $\mathbf{v} = \mathbf{p} - \mathbf{x}$. Then the image intensity at pixel $\mathbf{u}$ for the camera at position $\mathbf{p}$ is

$$I(\mathbf{u}, \mathbf{p}) = \rho(\mathbf{x}, \mathbf{n}, \mathbf{s}, \mathbf{v}) \tag{4.2}$$

where ρ is the BRDF function, and the cosine falloff term is absorbed into ρ. Note that unlike most previous work, ρ can be a general *spatially-varying* BRDF. Practical solutions will require a general diffuse plus 1-lobe specular form (Sec. 4.4), but the BRDF can still be spatially-varying.

Now suppose there is another camera centered at $\mathbf{p} + \boldsymbol{\tau}$, where $\boldsymbol{\tau} = (\tau_x, \tau_y, 0)^\top$. Also suppose a point at pixel $\mathbf{u}$ in the first camera image has moved to pixel $\mathbf{u}+\delta\mathbf{u}$ in the second camera image. Since the viewpoint has changed, the brightness constancy constraint in traditional optical flow no longer holds. Instead, since the view direction has changed by a small amount $\boldsymbol{\tau}$ and none of $\mathbf{x}, \mathbf{n}, \mathbf{s}$ has changed, the intensities of these two pixels can be related by a first-order approximation

$$I(\mathbf{u} + \delta\mathbf{u}, \mathbf{p} + \boldsymbol{\tau}) \cong I(\mathbf{u}, \mathbf{p}) + (\nabla_{\mathbf{v}}\rho)^\top \boldsymbol{\tau} \tag{4.3}$$

We can also model the intensity of the second image by,

$$I(\mathbf{u} + \delta\mathbf{u}, \mathbf{p} + \boldsymbol{\tau}) \cong I(\mathbf{u}, \mathbf{p}) + (\nabla_{\mathbf{u}}I)^\top \delta\mathbf{u} + (\nabla_{\mathbf{p}}I)^\top \boldsymbol{\tau} \tag{4.4}$$

Note that $(\nabla_{\mathbf{p}}I)^\top \boldsymbol{\tau}$ is just the difference between the image intensities of the two cameras, $I(\mathbf{u}, \mathbf{p} + \boldsymbol{\tau}) - I(\mathbf{u}, \mathbf{p})$. Let ΔI be this intensity difference. Combining (4.3) and (4.4) then gives

$$(\nabla_{\mathbf{u}}I)^\top \delta\mathbf{u} + \Delta I = (\nabla_{\mathbf{v}}\rho)^\top \boldsymbol{\tau} \tag{4.5}$$

Finally, since the second camera has moved by $\boldsymbol{\tau}$, all objects in the scene can be considered as equivalently moved by $\delta\mathbf{x} = -\boldsymbol{\tau}$ while assuming the camera is fixed. Using (4.1), we can write

$$\delta\mathbf{u} = \frac{\delta\mathbf{x}}{1 + \beta z} = \frac{-\boldsymbol{\tau}}{1 + \beta z} \tag{4.6}$$

Substituting this term for $\delta\mathbf{u}$ in (4.5) yields

$$(\nabla_{\mathbf{u}} I)^\top \frac{-\boldsymbol{\tau}}{1+\beta z} + \Delta I = (\nabla_{\mathbf{v}} \rho)^\top \boldsymbol{\tau} \tag{4.7}$$

Let I_u, I_v be the spatial derivatives of image $I(\mathbf{u}, \mathbf{p})$. Then multiplying the vector form out in (4.7) gives

$$\Delta I = (\nabla_{\mathbf{v}} \rho)_x \tau_x + (\nabla_{\mathbf{v}} \rho)_y \tau_y + I_u \frac{\tau_x}{1+\beta z} + I_v \frac{\tau_y}{1+\beta z} \tag{4.8}$$

where $(\cdot)_x$ and $(\cdot)_y$ mean the x- and y-components of $(\cdot)$, respectively.

An intuition for the above equation is given in Fig. 4.2. Consider the 1D case where two cameras are separated by distance τ_x. The 2D case can be derived similarly. First, an object is imaged at pixel u on camera 1 and u' on camera 2. The difference of the two images at pixel u, $\Delta I(u) = I(u, \tau_x) - I(u, 0)$ in Fig. 4.2a, will be the difference caused by the *view change* (from $I(u, 0)$ to $I(u', \tau_x)$ in Fig. 4.2b), plus the difference caused by the *spatial change* (from $I(u', \tau_x)$ to $I(u, \tau_x)$ in Fig. 4.2c). The view change is modeled by $(\nabla_{\mathbf{v}} \rho)_x \cdot \tau_x$, which is how the BRDF varies with viewpoint multiplied by the view change amount. The spatial change is modeled by $I_u \cdot \tau_x/(1+\beta z)$, which is the image derivative multiplied by the change in image coordinates. Summing these two terms gives (4.8) (Fig. 4.2d).

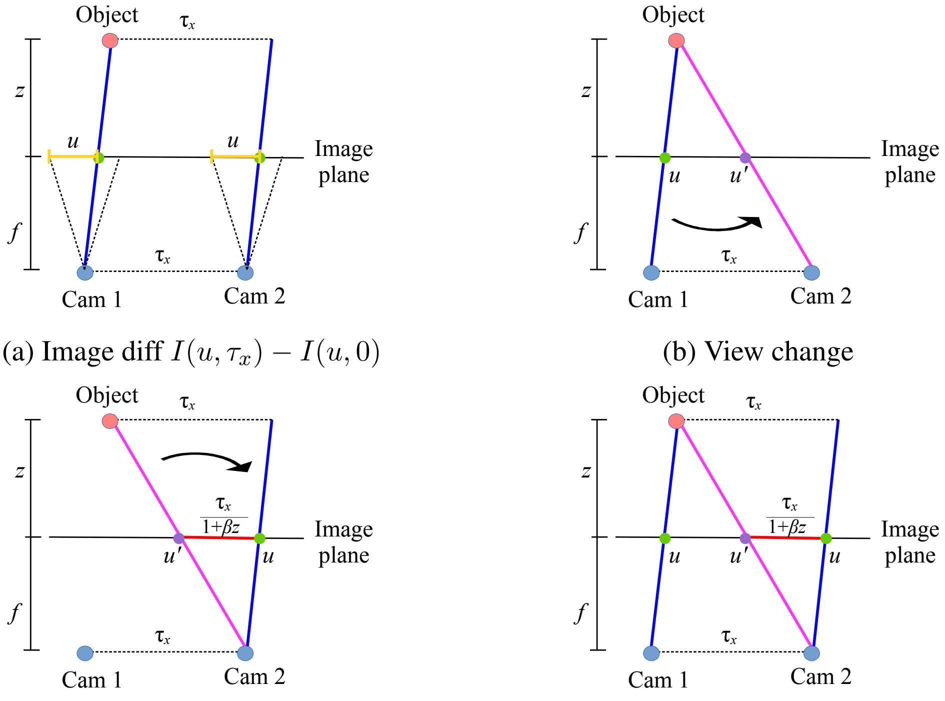
Compared with the work by Chandraker [52, 53, 54], we note that since different system setups are considered, the parameterization of the total intensity change in (4.3) is different ($(\nabla_{\mathbf{v}} \rho)^\top \boldsymbol{\tau}$ instead of $(\nabla_{\mathbf{x}} \rho)^\top \boldsymbol{\tau}$ in Appendix D of [54]). We believe this parameterization is more intuitive, since it allows the above physical interpretation of the various terms in the total intensity change.

Note that in the previous derivation we assumed all cameras are looking straight ahead, i.e., focused at infinity. For a light-field camera, however, it may be focused at some finite distance, which means if we average over all the different sub-views, objects at this distance will stay in-focus while other objects will become out-of-focus. The derivation for this case is as follows. Assume the cameras are focused at depth F. The view change will then stay the same as in the previous case. The spatial change, however, will become different since the change in pixel coordinates (distance between u' and u) changes. From Fig. 4.3b, we can see that the distance between u' and u becomes

$$\overrightarrow{u'u} = -\frac{f}{f+F}\overrightarrow{AB} = -\frac{f}{f+F} \cdot \frac{z-F}{z+f}\tau_x \tag{4.9}$$

Comparing it to Fig. 4.3a, it can be seen that $1/(1+\beta z)$ is replaced by

$$\frac{-f}{F+f}\frac{z-F}{z+f} = \frac{F-z}{F+f}\frac{1}{1+\beta z} \tag{4.10}$$



(a) Image diff $I(u, \tau_x) - I(u, 0)$

(b) View change

(c) Spatial change

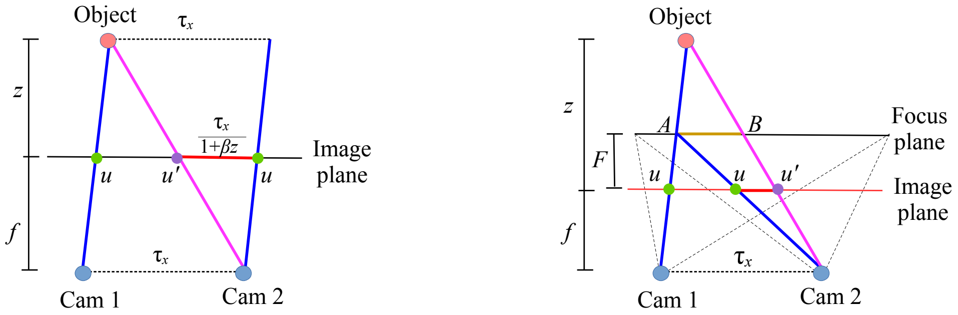
(d) Overall change

Figure 4.2: *Optical flow for glossy surfaces. (a) The difference between two images at the same pixel position, is (b) the view change plus (c) the spatial change. (d) Summing these two changes gives the overall change.*

Note that when $F \to \infty$, i.e., all cameras are looking straight ahead, the above expression reduces to $1/(1+\beta z)$. We can then replace $1/(1+\beta z)$ in (4.8) with this new term without changing anything else.

4.3.2 Multi-camera System

We now move on to consider the case of a light-field camera, which can be modeled by a multi-camera array. For a multi-camera array with $m+1$ cameras, we can form m camera pairs using the central camera and each of the other cameras. Let the translations of each pair be $\boldsymbol{\tau}^1, \boldsymbol{\tau}^2, ..., \boldsymbol{\tau}^m$ and the corresponding image differences be $\Delta I^1, \Delta I^2, ..., \Delta I^m$. Each pair will then have a stereo relation equation as in (4.8). We



(a) Cameras focused at infinity

(b) Cameras focused at depth F

Figure 4.3: *Comparison between cameras focused at infinity and focused at some finite depth.*

can stack all the equations and form a linear system as

$$\begin{bmatrix} I_u\tau_x^1 + I_v\tau_y^1 & \tau_x^1 & \tau_y^1 \\ & \dots & \\ I_u\tau_x^m + I_v\tau_y^m & \tau_x^m & \tau_y^m \end{bmatrix} \begin{bmatrix} \frac{1}{1+\beta z} \\ (\nabla_{\mathbf{v}}\rho)_x \\ (\nabla_{\mathbf{v}}\rho)_y \end{bmatrix} = \begin{bmatrix} \Delta I^1 \\ \dots \\ \Delta I^m \end{bmatrix}. \tag{4.11}$$

Let **B** be the first matrix in (4.11). If **B** is full rank, given at least three pairs of cameras (four cameras in total), we would be able to solve for depth by a traditional least squares approach. Unfortunately, it can easily be seen that **B** is rank deficient, since the first column is a linear combination of the other two columns. This should not be surprising, since we only have two degrees of freedom for translations in two directions, so the matrix is at most rank two. Adding more cameras does not add more degrees of freedom.[1] However, adding more cameras does increase the robustness of the system, as shown later in Fig. 4.5a. Finally, although directly solving for depth is not achievable, we can still obtain a relation between depth and normals for a specific form of the BRDF, which we derive next.

[1] Note that, adding translations in the z direction does not help either, since moving the camera along the viewing direction of a pixel does not change its pixel intensity ($\mathbf{v}/\|\mathbf{v}\|$ does not change), so $\nabla_{\mathbf{v}}\rho \cdot \mathbf{v} = 0$. Thus, $(\nabla_{\mathbf{v}}\rho)_z$ is just a linear combination of $(\nabla_{\mathbf{v}}\rho)_x$ and $(\nabla_{\mathbf{v}}\rho)_y$, and adding it does not introduce any new degree of freedom.

4.4 BRDF-Invariant Derivation

We first briefly discuss the BRDF model we adopt (Sec. 4.4.1), and then show how we can derive a BRDF invariant equation relating depth and normals (Sec. 4.4.2). A comparison between our work and the work by Chandraker [52, 53, 54] is given in Sec. 4.4.3.

4.4.1 BRDF model

It is commonly assumed that a BRDF contains a sum of "lobes" (certain preferred directions). Thus, the BRDF can be represented as a sum of univariate functions [74]:

$$\rho(\mathbf{x}, \mathbf{n}, \mathbf{s}, \mathbf{v}) = \sum_{i=1}^{K} f_{\mathbf{x},i}(\hat{\mathbf{n}}^\top \hat{\boldsymbol{\alpha}}_i) \cdot (\hat{\mathbf{n}}^\top \hat{\mathbf{s}}) \tag{4.12}$$

where $\hat{\mathbf{n}}$ is the normalized normal, $\hat{\boldsymbol{\alpha}}_i$ are some directions, $f_{\mathbf{x},i}$ are some functions at position $\mathbf{x}$, and K is the number of lobes. For the rest of the chapter, when we use $\hat{\mathbf{w}}$ to represent a vector $\mathbf{w}$, it means it is the normalized form of $\mathbf{w}$.

The model we adopted is similar to the Blinn-Phong BRDF; for each of the RGB channels, the BRDF is 1-lobe that depends on the half-angle direction $\hat{\mathbf{h}} = (\hat{\mathbf{s}}+\hat{\mathbf{v}})/\|\hat{\mathbf{s}}+\hat{\mathbf{v}}\|$, plus a diffuse term which is independent of viewpoint,

$$\begin{aligned} \rho^c(\mathbf{x}, \mathbf{n}, \mathbf{s}, \mathbf{v}) = \big(\rho_d^c(\mathbf{x}, \mathbf{n}, \mathbf{s}) + \rho_s^c(\mathbf{x}, \hat{\mathbf{n}}^\top \hat{\mathbf{h}})\big) \cdot (\hat{\mathbf{n}}^\top \hat{\mathbf{s}}),& \\ c \in \{\text{red,green,blue}\}& \end{aligned} \tag{4.13}$$

For the work by Tao et al. [55], it is assumed that the BRDFs of different views will lie on a line not passing the origin in the RGB space. Taking a look at, e.g., the BRDFs in Fig. 4.9, we can see that the BRDFs do not necessarily lie on a line, and passing the origin is possible for the materials whose diffuse components are not significant.

4.4.2 BRDF invariant

To derive the invariant, we first derive two expressions for $\nabla_{\mathbf{v}}\rho$, one using depth z and the other using normals $\mathbf{n}$. Combining these two expressions gives an equation which contains only z and $\mathbf{n}$ as unknowns and is invariant to the BRDF. We then show how to solve it for shape.

a. Expression using depth Continuing from (4.11), let $\boldsymbol{\gamma} = \mathbf{B}^+(\boldsymbol{\Delta I})$, where $\mathbf{B}^+$ is the Moore-Penrose pseudoinverse of $\mathbf{B}$. Then (4.11) has an infinite number of solutions,

$$\begin{bmatrix} \dfrac{1}{1+\beta z} \\ (\nabla_{\mathbf{v}}\rho)_x \\ (\nabla_{\mathbf{v}}\rho)_y \end{bmatrix} = \boldsymbol{\gamma} + \lambda \begin{bmatrix} 1 \\ -I_u \\ -I_v \end{bmatrix} \tag{4.14}$$

with $\lambda \in \mathbb{R}$. Let $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \gamma_3]^\top$. From the first row λ can be expressed as

$$\lambda = \frac{1}{1 + \beta z} - \gamma_1 \tag{4.15}$$

Thus, we can express $(\nabla_{\mathbf{v}}\rho)_y/(\nabla_{\mathbf{v}}\rho)_x$, which can be seen as the direction of the BRDF gradient, as a function of z,

$$\frac{(\nabla_{\mathbf{v}}\rho)_y}{(\nabla_{\mathbf{v}}\rho)_x} = \frac{\gamma_3 - \lambda I_v}{\gamma_2 - \lambda I_u} = \frac{\gamma_3 - (\frac{1}{1+\beta z} - \gamma_1) I_v}{\gamma_2 - (\frac{1}{1+\beta z} - \gamma_1) I_u} \tag{4.16}$$

b. Expression using normals Next, using the BRDF model in (4.13), in Appendix A.1 we show that

$$\nabla_{\mathbf{v}}\rho = \rho'_s \frac{\hat{\mathbf{n}}^\top \mathbf{H}}{\|\hat{\mathbf{s}} + \hat{\mathbf{v}}\|(1 + \beta z)\sqrt{u^2 + v^2 + f^2}} \tag{4.17}$$

where $\rho'_s = \partial\rho_s/\partial(\hat{\mathbf{n}}^\top \hat{\mathbf{h}})$ is an unknown function, and $\mathbf{H} \equiv (\mathbf{I} - \hat{\mathbf{h}}\hat{\mathbf{h}}^\top)(\mathbf{I} - \hat{\mathbf{v}}\hat{\mathbf{v}}^\top)$ is a known 3×3 matrix.

Since ρ'_s is unknown, we cannot express $\nabla_{\mathbf{v}}\rho$ as a function of $\mathbf{n}$ and z only. However, if we take the ratio between the y-component and the x-component of $\nabla_{\mathbf{v}}\rho$ corresponding to the direction of the gradient, all unknowns except $\mathbf{n}$ will disappear,

$$\frac{(\nabla_{\mathbf{v}}\rho)_y}{(\nabla_{\mathbf{v}}\rho)_x} = \frac{(\hat{\mathbf{n}}^\top \mathbf{H})_y}{(\hat{\mathbf{n}}^\top \mathbf{H})_x} = \frac{n_x H_{12} + n_y H_{22} - H_{32}}{n_x H_{11} + n_y H_{21} - H_{31}} \tag{4.18}$$

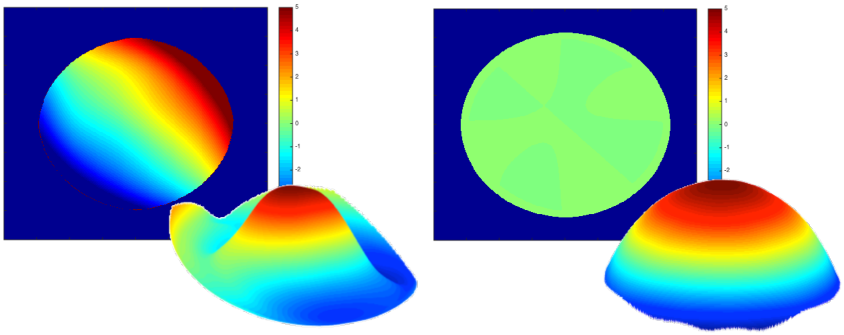
c. Combining expressions Equating the right-hand sides of (4.18) and (4.16) for the direction of the gradient $\nabla_{\mathbf{v}}\rho$ then gives

$$\frac{\gamma_3 - (\frac{1}{1+\beta z} - \gamma_1) I_v}{\gamma_2 - (\frac{1}{1+\beta z} - \gamma_1) I_u} = \frac{n_x H_{12} + n_y H_{22} - H_{32}}{n_x H_{11} + n_y H_{21} - H_{31}} \tag{4.19}$$

which is an equation of z and $\mathbf{n}$ only, since $\boldsymbol{\gamma}$ is known and $\mathbf{H}$ is known if $\mathbf{s}$ is known. Note that the spatially-varying BRDF dependent terms have been eliminated, and it is only possible for a single-lobe BRDF. Expanding (4.19) leads to solving a quasi-linear partial differential equation (PDE)

$$\boxed{(\kappa_1 + \kappa_2 z) n_x + (\kappa_3 + \kappa_4 z) n_y + (\kappa_5 + \kappa_6 z) = 0} \tag{4.20}$$

where κ_1 to κ_6 are constants specified in Appendix A.1. We call this the ***BRDF invariant*** relating depths and normals. Note that, in the case that $\nabla_{\mathbf{v}}\rho$ is zero, γ_2 and γ_3 will be zero for most solvers (e.g., `mldivide` in Matlab). Using the formulas for κ in Appendix A.1,



(a) Using infinite focus formula (b) Using finite focus formula

Figure 4.4: *(a) The BRDF invariant (4.20) result and the depth result when the cameras are focused at some finite distance but the formula for infinity focus (A.7) is used. Note that the BRDF invariant differs from zero by a large amount, and the depth reconstruction is far from accurate. (b) The results when the formula for finite focus (A.9) is used.*

(4.20) just reduces to $(\gamma_1 - 1) + (\beta\gamma_1)z = 0$, and z can be directly solved. This corresponds to the Lambertian case; the equation just stands for the photo-consistency, where the left hand side can be thought of as the intensity difference between different views. In the specular case, the same point in different views does not have the same intensity anymore; they differ by $(\nabla_{\mathbf{v}}\rho)^\top \tau$ (4.3), which can be written as a function of $\mathbf{n}$. That is where the first two normal terms in (4.20) come from.

Next we consider the case where the camera is focused at some finite distance instead of infinity. From (4.10) in Sec. 4.3.1, we know that we can replace $1/(1 + \beta z)$ by $(F - z)/[(F + f)(1 + \beta z)]$ to derive the formula for this case. Equation (4.19) then becomes

$$\frac{\gamma_3 - (\frac{F-z}{F+f}\frac{1}{1+\beta z} - \gamma_1)I_v}{\gamma_2 - (\frac{F-z}{F+f}\frac{1}{1+\beta z} - \gamma_1)I_u} = \frac{n_x H_{12} + n_y H_{22} - H_{32}}{n_x H_{11} + n_y H_{21} - H_{31}} \tag{4.21}$$

Using the same procedure, it will still lead to the same PDE as (4.20), only with different κ values. The new κ values are again specified in Appendix A.1. An example of applying this new formula on a synthetic sphere with finite-focused cameras is demonstrated in Fig. 4.4. In Fig. 4.4a, we use the old formula (assuming cameras focused at infinity), and it can be seen that it generates large errors on both the BRDF invariant result and the depth estimation result. In Fig. 4.4b, we replace it with the new formula (focused at finite distance), and we are able to get reasonable results.

4.4.3 Discussion

Compared to the work of Chandraker [52, 53, 54], we note that a similar BRDF invariant equation is derived. However, our derivation is in the light field setup and lends better physical intuition (Fig. 4.2). Moreover, our resolution of the shape ambiguity is distinct and offers several advantages. To be specific, the work of Chandraker assumes a constant viewing direction over the image, which can generate one more equation when solving the linear system (4.11), so directly recovering depth is possible. However, this is not true in the general perspective camera case. Instead, we directly solve the PDE, using a polynomial shape prior introduced next (Sec. 4.5.1). Furthermore, a homogeneous BRDF is also assumed in [52, 53, 54] to obtain depth directly. Our solution, on the other hand, is capable of dealing with spatially-varying BRDFs since we solve the PDE instead, as shown in the following section. Finally, while [52, 53, 54] are very sensitive to noise, we achieve robustness through multiple virtual viewpoints provided by the light field (Fig. 4.5a) and the polynomial regularization, as shown in the next section.

4.5 Shape and Reflectance Estimation

Given the BRDF invariant equation derived in Sec. 4.4, we utilize it to solve for shape (Sec. 4.5.1) and reflectance (Sec. 4.5.2) in this section.

4.5.1 Shape estimation

As shown in Appendix A.1, solving (4.20) mathematically requires initial conditions, so directly solving for depth is not possible. Several possible solutions can be used to address this problem. We adopt a polynomial regularization, similar to the approach proposed in [83, 84]. The basic idea is to represent z and n_x,n_y as some shape parameters, so solving (4.20) can be reduced to solving a system of quadratic equations in these parameters. Specifically, for an $\xi \times \xi$ image patch, we assume the depth can be represented by a quadratic function of the pixel coordinates u and v,

$$z(u, v) = a_1 u^2 + a_2 v^2 + a_3 uv + a_4 u + a_5 v + a_6 \tag{4.22}$$

where $a_1, a_2, ..., a_6$ are unknown parameters.

We now want to express normals using these parameters as well. However, to compute $n_x = \partial z/\partial x$, we need to know the x-distance between the 3D points imaged on those two pixels, which is not given. Therefore, we cannot directly compute n_x and n_y. Instead, we first compute the normals in the *image coordinate*,

$$
\begin{aligned}
n_u(u,v) &= \frac{\partial z}{\partial u} = 2a_1 u + a_3 v + a_4 \\
n_v(u,v) &= \frac{\partial z}{\partial v} = 2a_2 v + a_3 u + a_5
\end{aligned}
\tag{4.23}
$$

In Appendix A.2 we show that normals in the *world coordinate* n_x are related to normals in the image coordinate n_u by

$$
n_x = \frac{\partial z}{\partial x} = \frac{n_u}{1 + \beta(3z - 2a_6 - a_4 u - a_5 v)}
\tag{4.24}
$$

and n_y is computed similarly. Thus, (4.20) can be rewritten as

$$
\begin{aligned}
&(\kappa_1 + \kappa_2 z) n_u + (\kappa_3 + \kappa_4 z) n_v \\
&\quad + (\kappa_5 + \kappa_6 z)\big(1 + \beta(3z - 2a_6 - a_4 u - a_5 v)\big) = 0
\end{aligned}
\tag{4.25}
$$

Plugging (4.22)-(4.23) into (4.25) results in ξ^2 quadratic equations in $a_1, ..., a_6$, one for each pixel in the $\xi \times \xi$ patch,

$$
\begin{aligned}
(3\beta\kappa_6 u^4 + 2\kappa_2 u^3) a_1^2 + 2(3\beta\kappa_6 u^2 v^2 + \kappa_4 u^2 v + \kappa_2 u v^2) a_1 a_2 \\
+(\kappa_4 u^3 + 3\kappa_2 u^2 v + 6\beta\kappa_6 u^3 v) a_1 a_3 + \cdots + \\
(3\beta\kappa_6 v^4 + 2\kappa_4 v^3) a_2^2 + (\kappa_2 v^3 + 3\kappa_4 u v^2 + 6\beta\kappa_6 u v^3) a_2 a_3 \\
+ \cdots + \beta\kappa_6 a_6^2 + (2\kappa_1 u + \kappa_6 u^2 + 3\beta\kappa_5 u^2) a_1 + \cdots + \kappa_5
\end{aligned}
\tag{4.26}
$$

We can then factorize the above equation into the following form for easier optimization

$$
\begin{bmatrix} \mathbf{a}^\top & 1 \end{bmatrix} \mathbf{M}_i \begin{bmatrix} \mathbf{a} \\ 1 \end{bmatrix} = 0 \qquad i = 1, 2, ..., \xi^2
\tag{4.27}
$$

where $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{bmatrix}^\top$ and $\mathbf{M}_i$ is a 7×7 matrix whose formula is given in the supplementary matrix. Since we have 6 unknowns and ξ^2 equations, any patch larger than or equal to 3×3 would suffice to solve for **a**. We choose the patch size as 5×5 in our experiment.

Next, for spatial coherence we enforce neighboring pixels to have similar depths and normals. To avoid ambiguity we require the normal at one seed pixel to be specified; in practice we specify the nearest point and assume its normal is the $-z$ direction. The shape parameters for other pixels in the image are then estimated accordingly. Our final optimization thus consists of a data term D that ensures the image patch satisfies the

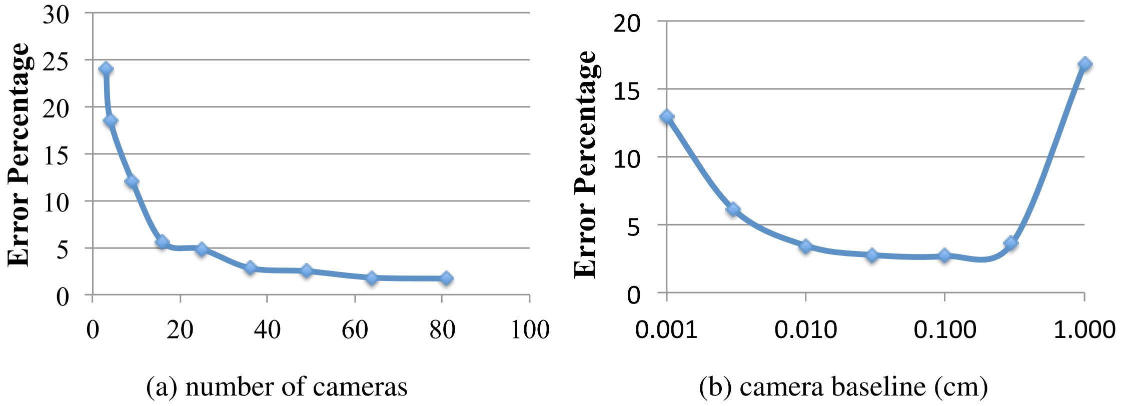


Figure 4.5: *(a) Depth error vs. number of cameras (virtual viewpoints) used. We add Gaussian noise of variance* 10^{-4} *on a synthetic sphere and test the performance when different numbers of cameras are used, from three to 81 (9×9 array). Although theoretically three cameras are enough, depth recovery is very sensitive to noise. As the number of cameras increases, the system becomes more robust. (b) Depth error vs. camera baseline. Our method performs the best when the baseline is between 0.01 cm to about 0.5 cm.*

PDE, and a smoothness term S that ensures neighboring normals and depths (a_4 to a_6) are similar,

$$\mathbf{a} = \arg\min_{\mathbf{a}} \sum_i D_i^2 + \eta \sum_j S_j^2 \tag{4.28}$$

where D_i is computed by (4.27), and

$$S_j = a_j - a_j^0 \qquad j = 4, 5, 6 \tag{4.29}$$

where a_j^0 is the average a_j of its 4-neighbors that have already been computed, and η is the weight, which is 10^3 in our experiment. We then apply standard Levenberg-Marquardt method to solve for the parameters.

Finally, note that although theoretically, three cameras are enough to solve for depth, in practice more cameras will increase the robustness against noise, as shown in Fig. 4.5a. Indeed, the multiple views provided by light-field cameras are essential to obtaining high-quality results. More cameras, along with the polynomial regularizer introduced above, helps to increase the system robustness compared to previous work [52, 53, 54]. Next, in Fig. 4.5b, we further test the effect of different baselines. We vary the baseline from 10^{-3} to 1 cm, and report their depth errors. As can be seen, our method achieves best performance when the baseline is between 0.01 cm to about 0.5 cm. When the baseline is too small, there is little difference between adjacent images; when the baseline is too large, the differential motion assumption fails. Note that the effective baseline for Lytro Illum changes with focal length and focus distance, and is in the order of 0.01 to 0.1 cm, so our method is well suited to the practical range of baselines.

4.5.2 Reflectance estimation

After the shape is recovered, reflectance can also be recovered, similar to [52]. First, $(\nabla_{\mathbf{v}}\rho)_x$ and $(\nabla_{\mathbf{v}}\rho)_y$ can be obtained using (4.14). Then (4.17) can be used to recover ρ_s'. Specifically, let $k \equiv \|\hat{\mathbf{s}} + \hat{\mathbf{v}}\|(1 + \beta z)\sqrt{u^2 + v^2 + f^2}$, then

$$\begin{aligned} \rho_s' &= k(\nabla_{\mathbf{v}}\rho)_x / (\hat{\mathbf{n}}^\top \mathbf{H})_x \\ &= k(\nabla_{\mathbf{v}}\rho)_y / (\hat{\mathbf{n}}^\top \mathbf{H})_y \end{aligned} \tag{4.30}$$

In practice we just take the average of the two expressions to obtain ρ_s'. A final integration over $\hat{\mathbf{n}}^\top \hat{\mathbf{h}}$ then suffices to generate ρ_s. Finally, subtracting ρ_s from the original image gives the diffuse component (4.13). Note that although we assumed a 1-lobe BRDF to obtain the depth information, if shape is already known, then ρ can actually be 2-lobe since two equations are given by the x- and y-component of (4.17). Specifically, from (4.17) we have

$$\begin{aligned} (\nabla_{\mathbf{v}}\rho)_x &= \rho_{s,1}' m_x + \rho_{s,2}' q_x \\ (\nabla_{\mathbf{v}}\rho)_y &= \rho_{s,1}' m_y + \rho_{s,2}' q_y \end{aligned} \tag{4.31}$$

where $\rho_{s,1}'$, $\rho_{s,2}'$ are (unknown) derivatives of the two BRDF lobes, and other variables are constants. Since we have two unknowns and two equations, we can solve for the BRDFs.

4.6 Results

We validate our algorithm using extensive synthetic scenes as well as real-world scenes. We compare our results with two methods by Tao et al., one using point and line consistency to deal with specularity (PLC) [55] and one that handles diffuse only but includes the shading cue (SDC) [36]. We also compare with the phase-shift method by Jeon et al. (PSSM) [20], results by Lytro Illum, and the results by Chandraker (SMRM [52, 53] and IAMO [54]). Since the pixel clustering method by Tao et al. [59] has been superseded by [55], we only include the comparison with [55] here.

4.6.1 Synthetic scenes

For synthetic scenes, we use a 7 × 7 camera array of 30 mm focal length. We test on a sphere of radius 10 cm positioned at 30 cm away from the cameras, and also on a randomly genrated complicated shape. Figure 4.9 shows example results on materials in the

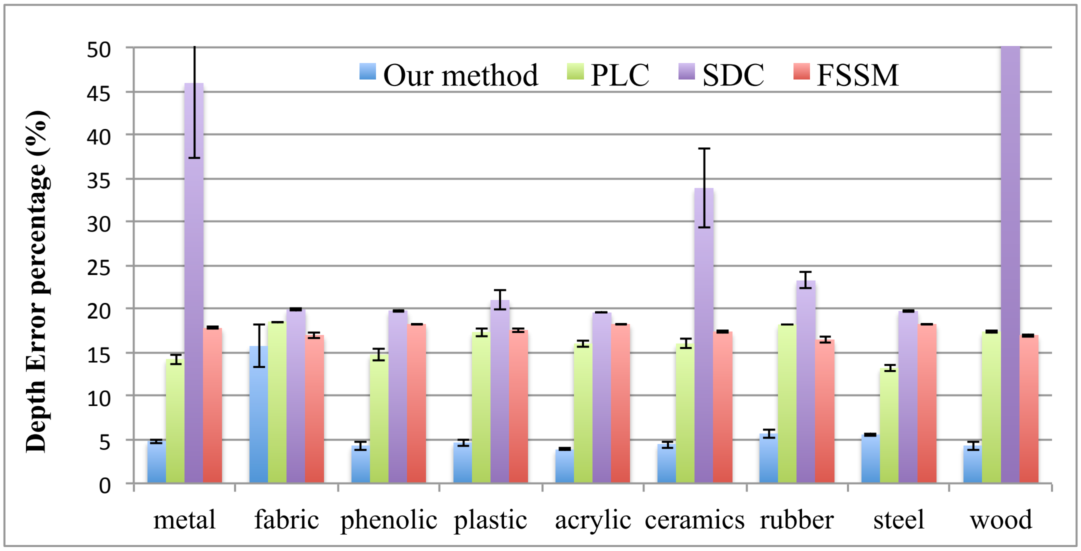
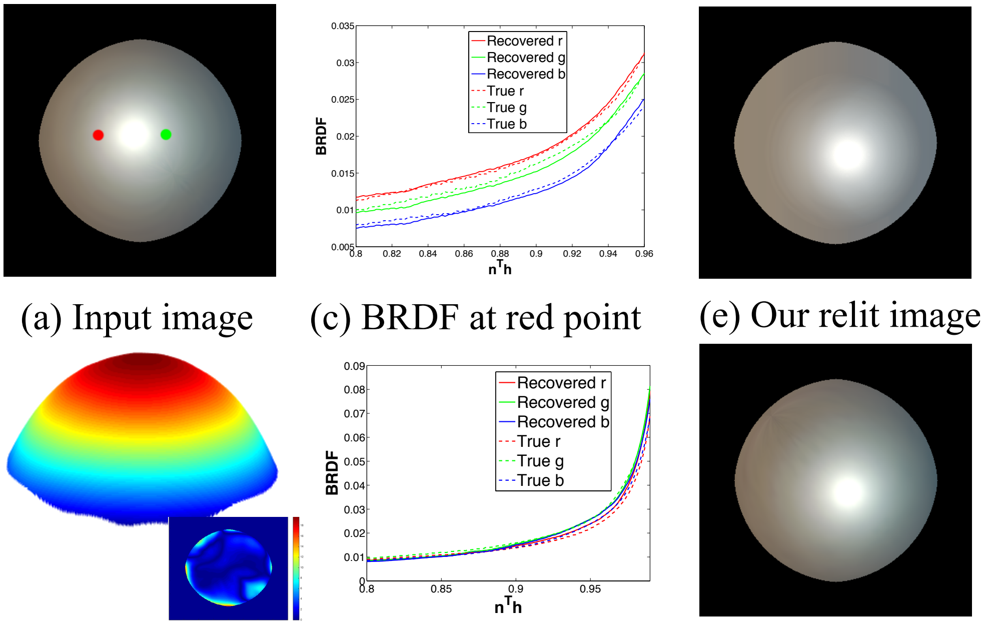


Figure 4.6: *Depth errors on different material types. Our method achieves good results on all materials except fabric. For all material types, we outperform the other methods.*

MERL BRDF dataset [58] on the sphere, while Fig. 4.10 shows results on the complicated shape. Note that spheres are not a polynomial shape ($z = \sqrt{r^2 - x^2 - y^2}$). We provide a summarized figure showing depth errors on different material types in Fig. 4.6. It can be seen that our method achieves good results on most material types except fabric, which does not follow the half-angle assumption. However, for all the material types, we still outperform the other state-of-the-art methods. For PLC [55], although it tries to handle glossy surfaces, the line consistency they adopted is not able to handle general BRDFs. In fact, from their internal result, we found that most pixels in their final result are still using point-consistency. For SDC [36] and PSSM [20], they are designed for Lambertian scenes and perform poorly on glossy objects. For SMRM [52, 53] and IAMO [54], they assume that the camera view direction is constant over the entire object, which is only an approximation for perspective projection. Finally, to evaluate our reflectance reconstruction we compute the ground truth BRDF curves by averaging BRDFs for all given half-angles. It can be seen that our curves look very similar to the ground truth BRDFs.

Next, we test our method on a sphere with a spatially-varying BRDF, where we linearly blend two materials (alum bronze and green metal) from left to right (Fig. 4.7). In addition to recovering depth, we also compute the BRDFs for each column in the image, and show results for two sample columns and a relighting example, where we accurately produce results similar to the ground truth.

Finally, to test the adaptability of our quadratic shape model (4.22), we randomly generate different shapes, and apply different sizes of Gaussian filters on the frequency



(a) Input image (c) BRDF at red point (e) Our relit image

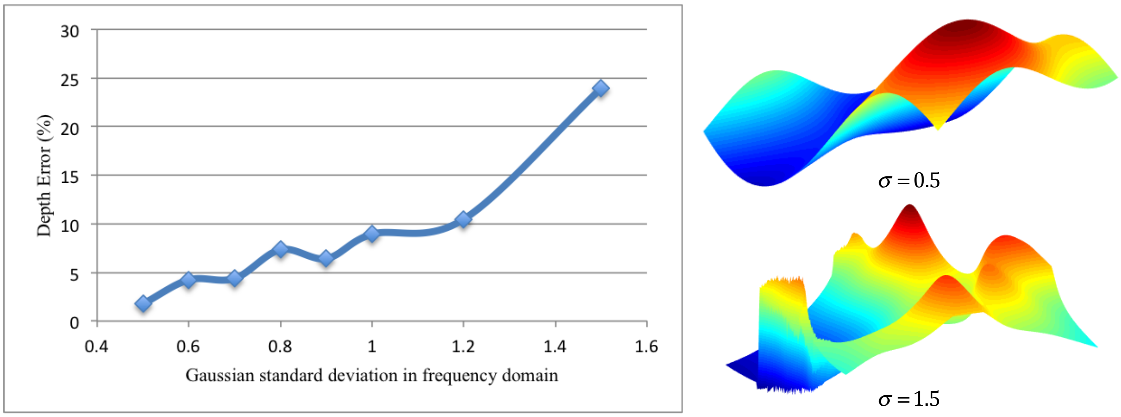
(b) Our depth (d) BRDF at green point (f) GT relit image

Figure 4.7: *Shape and reflectance estimation results on a spatially-varying example in the MERL dataset. (a) Two materials, alum bronze and green metal, are blended linearly from left to right. We reconstruct (b) the depth and (c)(d) the BRDFs for each column, where two examples are shown at the red and green points specified in (a). Finally, we show a relighting example in (e). The error percentage compared to (f) the ground truth is* 3.20%.

domain to generate shapes with different smoothnesses. The plot for depth errors vs. Gaussian standard deviations is shown in Fig. 4.8. It can be seen that as the shape becomes more and more complicated, the quadratic assumption starts to fail and leads to large errors. A possible solution would be to use higher order polynomials to better approximate the shapes.

4.6.2 Real-world results

We show results taken with the Lytro Illum in Figs. 5.1, 4.11 and 4.12. Since the effective baseline changes for different focal lengths and focus distances, we fix them and use an object at a known distance z to calibrate the baseline offline. In particular, we first compute the disparity (in pixels) between neighboring views; then the baseline is



(a) error plot

(b) example shapes

Figure 4.8: *(a) We apply Gaussians with different standard deviations (σ) on the frequency domain to generate shapes with different smoothnesses, then plot the corresponding depth errors. (b) Example shapes for $\sigma = 0.5$ and $\sigma = 1.5$.*

calculated by

$$\text{baseline} = \left|\frac{zF}{f(z-F)}p_s\right| \cdot \text{disparity} \tag{4.32}$$

where F is the focused distance, f is the focal length, and p_s is the pixel size.

In Fig. 4.11 we show reconstructed shapes and BRDFs of objects with homogeneous BRDFs. For objects that are symmetric, we obtain the ground truth by a surface of revolution using the outline curve in the image, and compute the RMSE for each method. It can be seen that our method realistically reconstructs the shape, and achieves the lowest RMSE when ground truth is available. The recovered BRDFs also seem qualitatively correct, e.g., for the bowling pin its BRDF has a very sharp specularity. In Figs. 5.1 and 4.12 we show results of objects with spatially-varying BRDFs. For the first example in Fig. 4.12, we show results for a red ball with white stripes. It can be seen that other methods generate artifacts around the specular region, while ours captures the true shape. Also, our method achieves the lowest RMSE compared to the ground truth. For the other examples, we show figurines where we spray paints of different materials on their bodies. Again, it can be seen that other methods have artifacts or produce distorted shapes around the specular regions, while our method realistically reproduces the shape.

4.7 Conclusion

In this chapter, we propose a novel BRDF-invariant shape and reflectance estimation method for glossy surfaces from light-field cameras. By utilizing the differential motion theory, we show that direct shape recovery is not possible for general BRDFs. However, for a 1-lobe BRDF that depends only on half-angle, we derive an SVBRDF-invariant equation relating depth and normals. Using a locally polynomial prior on the surface, shape can be estimated using this equation. Reflectance is then also recovered using our framework. Spatially-varying BRDFs can also be handled by our method, while this is not possible using [52, 53, 54]. Experiments validate our algorithm on most material types in the MERL dataset, as well as real-world data taken with the Lytro Illum.

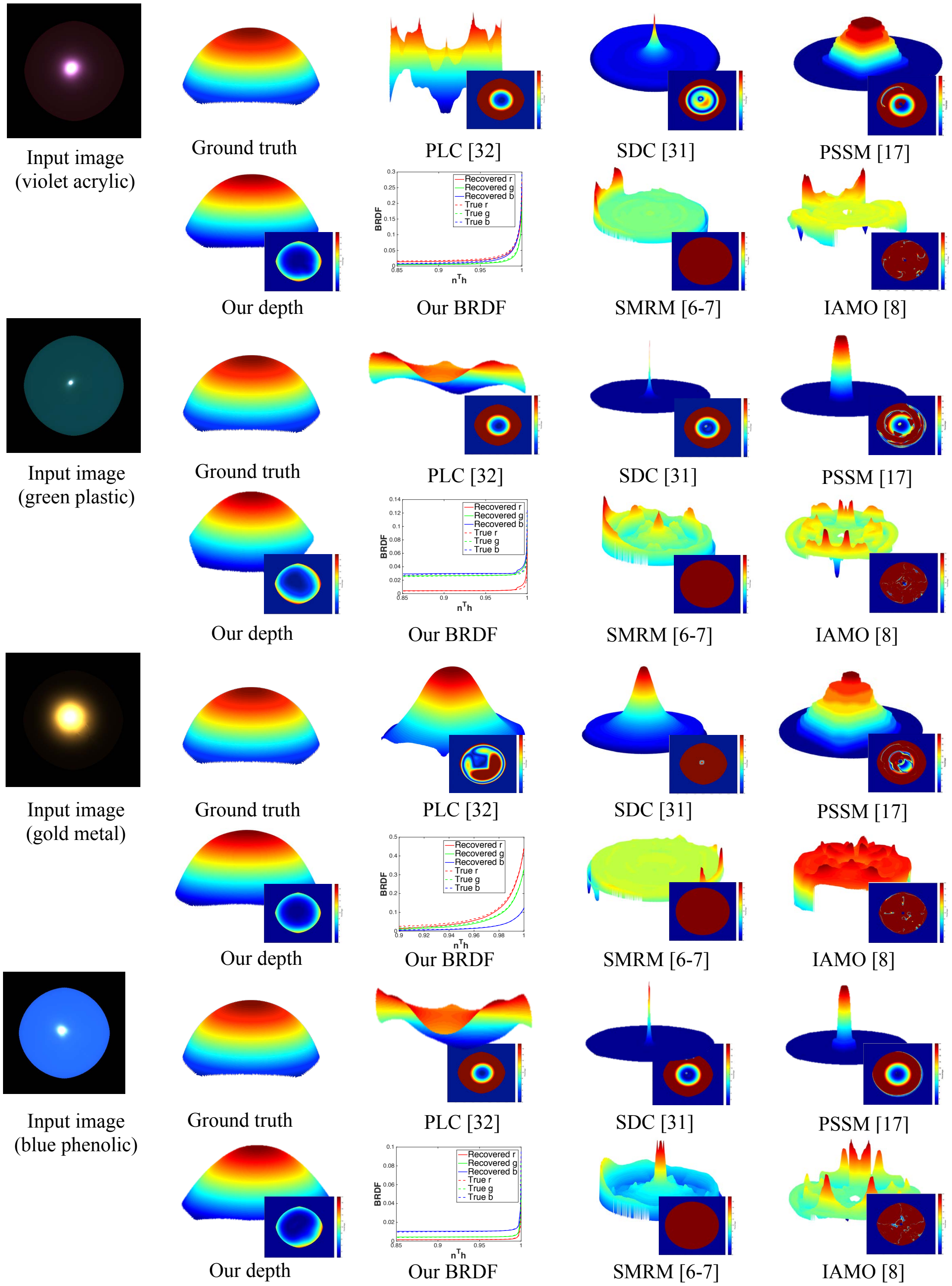


Figure 4.9: *Shape and reflectance estimation results on a sphere for example materials in the MERL dataset. For shape estimation, the upper-left shows the recovered depth, while the lower-right shows the error percentage (hotter color means larger error). For reflectance estimation, we show the recovered BRDF compared to ground truth curves.*

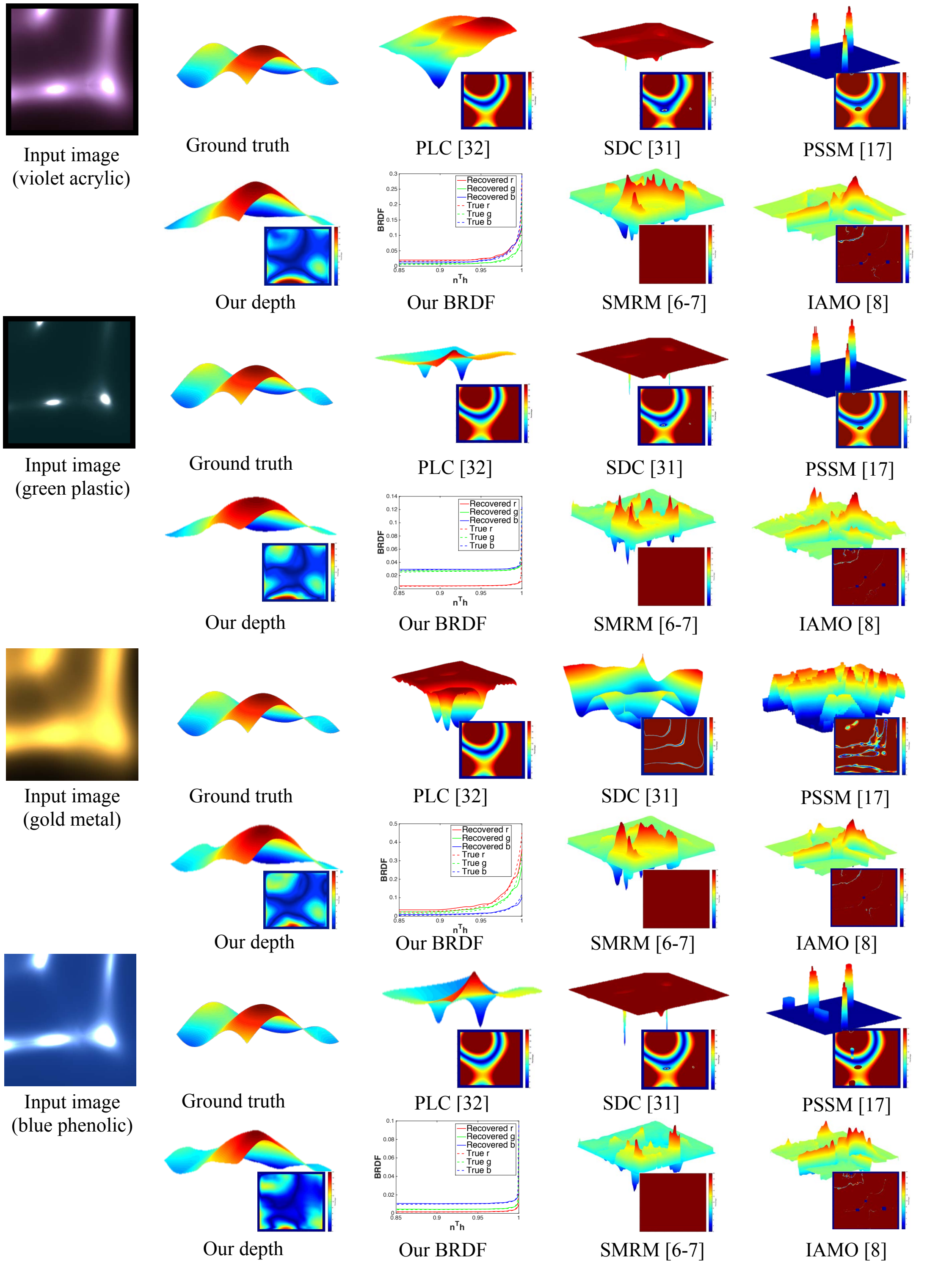


Figure 4.10: *Shape and reflectance estimation results on a more complicated shape for example materials in the MERL dataset. For shape estimation, the upper-left shows the recovered depth, while the lower-right shows the error percentage (hotter color means larger error). For reflectance estimation, we show the recovered BRDF compared to ground truth curves.*

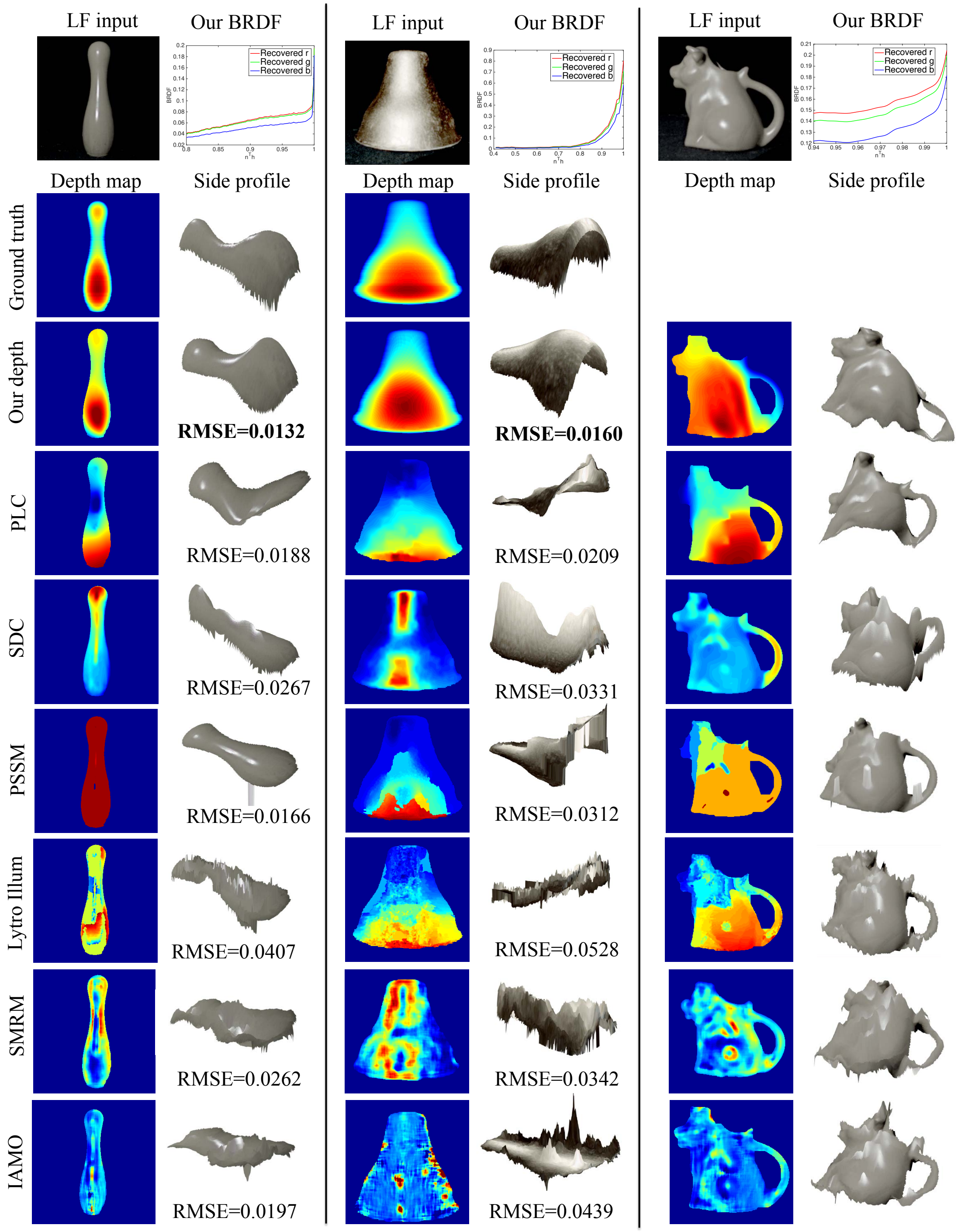


Figure 4.11: *Shape and reflectance estimation results on real data with homogeneous BRDFs. The intensities of the input images are adjusted for better contrast. For each example, we show both the depth map (hotter color means nearer) and the side view profile of the reconstructed shape. It can be seen that our method realistically reconstructs the shapes, and also achieves the lowest RMSE when ground truth is available. The recovered BRDFs also look qualitatively correct, e.g., the bowling pin has a very sharp specularity.*

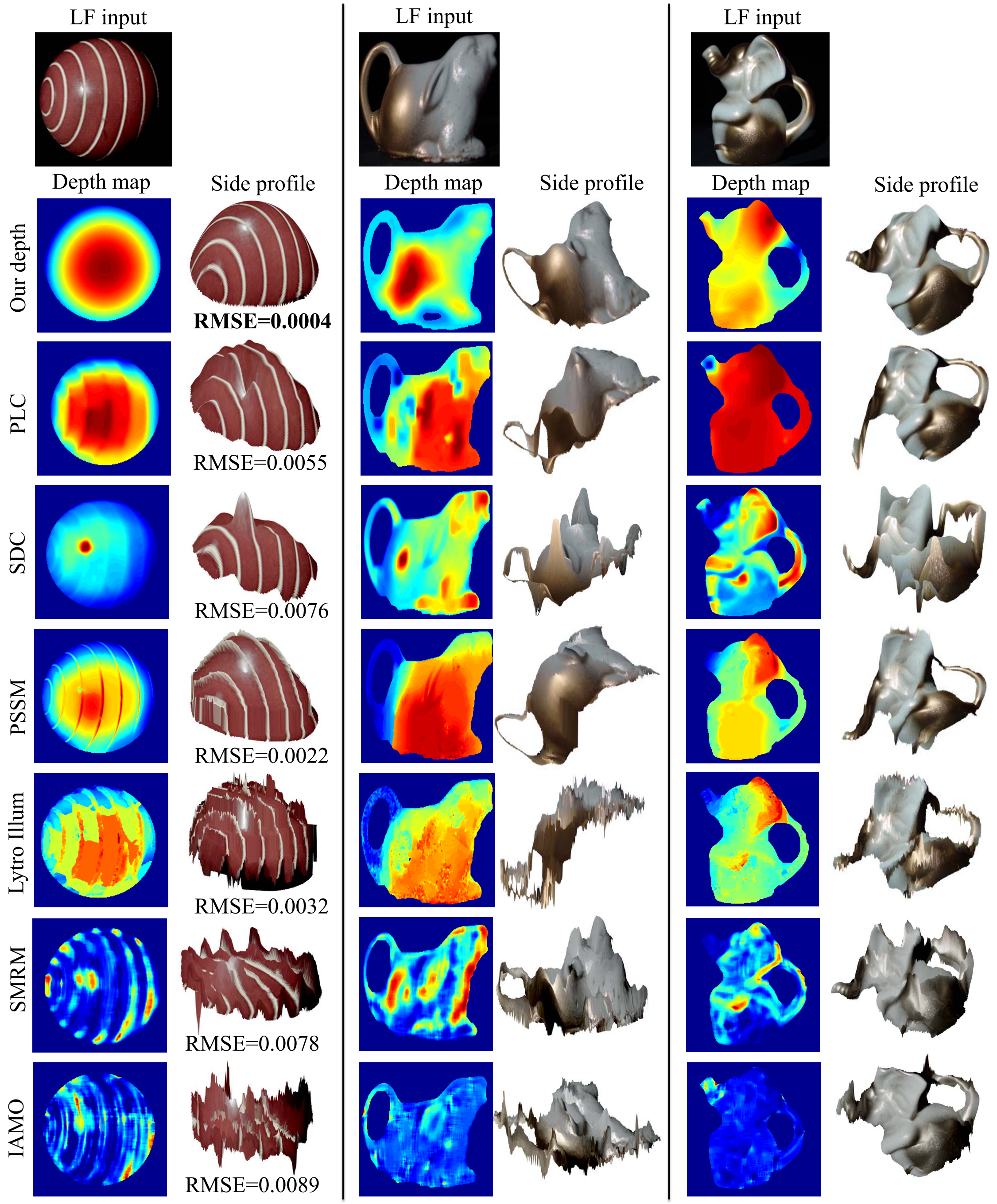


Figure 4.12: *Shape estimation results on real data with spatially-varying BRDFs. The intensities of the input images are adjusted for better contrast. For each example, we show both the depth map and the side view profile of the reconstructed shape. For the first example, we show results for a red ball with white stripes. It can be seen that other methods generate artifacts around the specular region, while ours captures the true shape. Also, our method achieves the lowest RMSE compared to the ground truth. For the other examples, we show figurines where we spray paints of different materials on their bodies. Again, it can be seen that other methods have artifacts or produce weird shapes around the specular regions, while our method realistically reproduces the shape.*

Chapter 5

Material Recognition with Light-Field Cameras

In previous chapters, we have seen how we can estimate better shapes and reflectances of objects by looking beyond photo-consistency. In particular, in Ch. 4 we validate that we can indeed extract the reflectance information from light-field images. This information can then be used in other applications. However, recall that in our BRDF model we assume a known light source, which makes it harder to be applied to real-world images.

In this chapter, we adopt a new approach to extract this hidden reflectance information, and aim to use it for material recognition. We take advantage of the recent success of deep learning methods, and collect a light-field material dataset to train and test our networks on. Our learning based approach can work on arbitrary images in the wild, thus having a greater impact in the field.

Since recognition networks have not been trained on 4D images before, we propose and compare several novel convolutional neural network (CNN) architectures to train on light-field images. Moreover, we train our system end-to-end on the classification results without explicitly estimating the reflectance, which often lead to better accuracy. In our experiments, the best performing CNN architecture achieves a 7% boost compared with 2D image classification ($70\% \rightarrow 77\%$). These results constitute important baselines that can spur further research in the use of CNNs for light-field applications, such as object detection, image segmentation and view interpolation.

5.1 Introduction

Materials affect how we perceive objects in our daily life. For example, we would not expect to feel the same when we sit on a wooden or leather chair. However, differentiating materials in an image is difficult since their appearance depends on the confounding effects of object shape and lighting. A more robust way to determine the material type is using the surface reflectance or the bidirectional reflectance distribution function (BRDF). However, measuring the reflectance is hard. Previous works use gonioreflectometers to recover the reflectance, which is cumbersome, and does not easily apply to spatially-varying BRDFs or Bidirectional Texture Functions (BTFs) [85, 86].

An alternative to directly measuring the reflectance, is to consider multiple views of a point at once. By doing so, material recognition can be improved as demonstrated by Zhang et al. [87]. We exploit the multi-views in a light-field representation instead. Light-field cameras have recently become available and are able to capture multiple viewpoints in a single shot. We can therefore obtain the intensity variation under different viewing angles with minimal effort. Therefore, one of the main goals of this chapter is to investigate whether 4D light-field information improves the performance of material recognition over 2D images [88]. This also provides a baseline method for future research in this area. We adopt the popular convolutional neural network (CNN) framework to perform material classification in this work. However, there are two key challenges: First, all previous light-field datasets include only a few images, so they are not large enough to apply the data-hungry deep learning approaches. Second, CNN architectures have previously not been adapted to 4D light-fields; Thus, novel architectures must be developed to perform deep learning with light-field inputs. Our contributions are shown in Fig. 5.1 and summarized below:

1) We introduce the first mid-size light-field image dataset (Sec. 5.3). Our dataset contains 12 classes, each with 100 images labeled with per pixel ground truth (Fig. 5.2). We then extract 30,000 patches from these images. Although we use this dataset for material recognition, it is not limited to this purpose and can be used for other light-field related applications, such as object recognition, segmentation, or view interpolation. The dataset is available online at `http://cseweb.ucsd.edu/~viscomp/projects/LF/papers/ECCV16/LF_dataset.zip`.

2) We investigate several novel CNN architectures specifically designed for 4D light-field inputs (Sec. 5.4). Since no recognition CNN has been trained on light-fields before, we implement different architectures to work on 4D data (Figs. 5.5 and 5.6). Instead of training a new network from scratch, we reuse the spatial filters from previous 2D models, while adding new angular filters into the network architecture. We also find directly training a fully convolutional network (FCN) very unstable, and thus train on extracted patches first and fine-tune on full images afterwards. The proposed architectures are not limited to material recognition, and may be used for other light-field based tasks as well.

3) Using our best-performing architecture, we achieve about 6-7% boost compared with single 2D image material classification, increasing the accuracy from 70% to 77% on extracted patches and 74% to 80% on full images (Sec. 5.5). These act as important

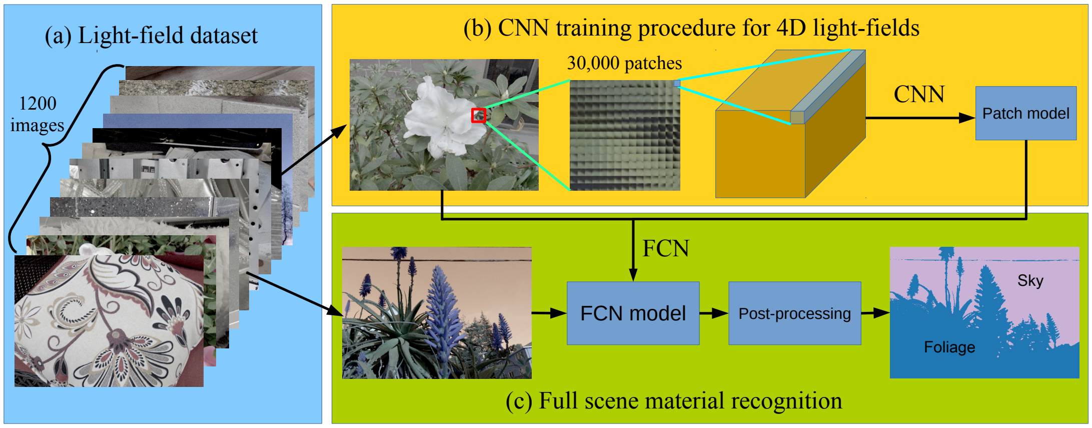


Figure 5.1: *Overview of our system and contributions. (a) We collect a new light-field dataset, which contains 1200 images labeled with 12 material classes. (b) Using (micro-lens) light-field patches extracted from this dataset, we train a CNN by modifying previous 2D models to take in 4D inputs. (c) Finally, we convert the patch model to an FCN model by fine-tuning on full images, and perform full scene material segmentation.*

baselines for future work in light-field based material recognition.

5.2 Related work

Light-field datasets: The most popular dataset is the one introduced by Wanner et al. [40], which contains 7 synthetic scenes and 5 real images captured using a gantry. Another well-known one is the Stanford light-field archive [89], which provides around 20 light-fields sampled using a camera array, a gantry and a light-field microscope. The synthetic light-field archive by Marwah et al. [9] contains 5 camera light-fields and 13 display light-fields. Other datasets contain fewer than ten images [18, 35, 90, 25] or are only suitable for particular purposes [91, 92]. Clearly, there is a lack of large light-field datasets in prior works. In this work, we use the Lytro Illum camera to build a dataset with 1200 light-field images.

Material databases: The early work on material recognition was primarily on classifying instance-level textures, such as the CUReT database [86] and the more diversified KTH-TIPS [93, 94] database. Recently, the Describable Textures Dataset (DTD) [95] features real-world material images. Some work on computer-generated synthetic datasets has also been introduced [96, 97].

For category-level material databases, the most well-known is the Flickr Material Database (FMD) [98], which contains ten categories with 100 images in each category.

Subsequently, Bell et al. [99] released OpenSurfaces which contains over 20,000 real-world scenes labeled with both materials and objects. More recently, the Materials in Context Database (MINC) [100] brought the data size to an even larger scale with 3 million patches classified into 23 materials. However, these datasets are all limited to 2D, and thus unsuitable for investigating the advantages of using multiple views. Although our dataset is not as large as the MINC dataset, it is the first mid-size 4D light-field dataset, and is an important step towards other learning based light-field research. Zhang et al. [87] also propose a reflectance disk dataset which captures intensities of different viewing angles for 20 materials. However, their dataset lacks the spatial information, and is much smaller compared to our dataset.

Material recognition: Material recognition methods can mainly be classified into two categories. The first one recognizes materials based on the object reflectance [101, 102, 103, 87]. Most work of this type requires the scene geometry or illumination to be known, or requires special measurement of the BRDF beforehand.

The other body of work extracts features directly from the image appearance, and is thus more flexible and can work on real-world images. Liu et al. [104] propose a model to combine low- and mid-level features using a Bayesian generative framework. Hu et al. [105] extend the Kernel descriptors with variances of gradient orientations and magnitudes to handle materials. Schwartz and Nishino [106] introduce visual material traits and explicitly avoid object-specific information during classification. Qi et al. [107] introduce a pairwise transform invariant feature and apply it to perform material recognition. Cimpoi et al. [95] propose a framework based on neural network descriptors and improved Fisher vectors (IFV). Recently, Cimpoi et al. [108] combine object descriptors and texture descriptors to achieve state-of-the-art results on FMD. However, none of these methods are applicable to the 4D case. In this work, we implement different methods to deal with this dimensionality change from 2D to 4D.

Convolutional neural networks: Convolutional neural networks (CNNs) have proven to be successful in modern vision tasks such as detection and recognition, and are now the state-of-the art methods in most of these problems. Since the work by Krizhevsky et al. [109] (a.k.a. AlexNet), in recent years many advanced architectures have been introduced, including GoogLeNet [110] and VGG [111]. For per-pixel segmentation, Farabet et al. [112] employ a multi-scale CNN to make class predictions at every pixel in a segmentation. A sliding window approach is adopted by Oquab et al. [113] to localize patch classification of objects. Recently, a fully convolutional framework [114] has been proposed to generate dense predictions from an image directly.

Multi-image CNNs: For CNNs trained on multiple image inputs, Yoon et al. [14] train a super-resolution network on light-field images; however, their goal is different from a high-level recognition task. Besides, only a couple of images instead of the full light-fields are sent into the network at a time, so the entire potential of the data is not exploited.

Su et al. [115] propose a “viewpooling” framework to combine multiple views of an object to perform object recognition. In their architecture, convolutional maps independently extracted from each view are maxpooled across all views. However, we find this does not work well in the light-field case. Rather, we demonstrate that it is advantageous to exploit the structure of light-fields in combining views much earlier in the network. This also has the advantage that memory usage is reduced. In this work, to ease the training of 4D light-fields, we initialize the weights with pre-trained 2D image models. We investigate different ways to map the 4D light-field onto the 2D CNN architecture, which has not been explored in previous work, and may be beneficial to learning-based methods for other light-field tasks in the future.

5.3 The light-field material dataset

While the Internet is abundant with 2D data, light-field images are rarely available online. Therefore, we capture the images ourselves using the Lytro Illum camera. There are 12 classes in our dataset: fabric, foliage, fur, glass, leather, metal, plastic, paper, sky, stone, water, and wood. Each class has 100 images labeled with material types. Compared with FMD [98], we add two more classes, fur and sky. We believe these two classes are very common in natural scenes, and cannot be easily classified into any of the ten categories in FMD.

The images in our dataset are acquired by different authors, in different locations (e.g. shops, campus, national parks), under different viewpoints and lighting conditions, and using different camera parameters (exposure, ISO, etc). The spatial resolution of the images is 376×541, and the angular resolution is 14×14. Since the pixel size of the Lytro camera is small ($1.4\mu m$), one problem we encountered is that the images are often too dark to be usable. Water is also a particularly challenging class to capture, since the corresponding scenes usually entail large motions. Overall, of the 1448 acquired images, we retain 1200 not deemed too repetitive, dim or blurred. We then manually classified and labeled the images with per pixel material category using the *Quick Selection Tool* of Photoshop. For each material region, we manually draw the boundary along the region. We check the segmentation results, and further refine the boundaries until we obtain final accurate annotation.

In Fig. 5.2 we show some example images for each category of the dataset, and in Fig. 5.3, we show some of the extracted patches.. Then, in Fig. 5.4a we show example light-field images, where each block of pixels shows different viewpoints of a 3D point. We then demonstrate the benefits of using light-fields: from the 2D images alone, it is difficult to separate sky from blue paper due to their similar appearances; However, with the aid from light-field images, it becomes much easier since paper has different reflectance from different viewpoints while sky does not. Next, in Fig. 5.4b we print out a photo of a pillow, and take a picture of the printed photo. We then test both 2D and light-field models on the picture. It is observed that the 2D model predicts the material as fabric since it assumes it sees a pillow, while the light-field model correctly identifies the material as paper.

Finally, to classify a point in an image, we must decide the amount of surrounding context to include, that is, determine the patch size. Intuitively, using small patches will lead to better spatial resolution for full scene material segmentation, but large patches contain more context, often resulting in better performance. Bell et al. [100] choose the patch scale as 23.3% of the smaller image length, although they find that scale 32% has the best performance. Since our images are usually taken closer to the objects, we use 34% of the smaller image length as the patch size, which generates about 30,000 patches of size 128×128. This is roughly 2500 patches in each class. The patch centers are separated by at least half the patch size; also, the target material type occupies at least half the patch. Throughout our experiments, we use an angular resolution of 7×7. We randomly select 70% of the dataset as training set and the rest as test set. Patches from the same image are either all in training or in test set, to ensure that no similar patches appear in both training and test sets.

5.4 CNN architectures for 4D light-fields

We now consider the problem of material recognition on our 4D light-field dataset and draw contrasts with recognition using 2D images. We train a Convolutional Neural Network for this patch classification problem. Formally, our CNN is a function f that takes a light-field image R as input and outputs a confidence score p_k for each material class k. The actual output of f depends on the parameters θ of the network that are tuned during training, i.e., $p_k = f(R; \theta)$. We adopt the softmax loss, which means the final loss for a training instance is $-\log(e^{p_t}/(\sum_{i=1}^{k} e^{p_i}))$, where t is the true label. At test time, we apply the softmax function on the output p_k, where the results can be seen as the predicted probability per class.

We use the network architecture of the recent VGG-16 model [111], a 16-layer model, as it performs the best on our dataset when using 2D images. We initialize the weights using the MINC VGG model [100], the state-of-the-art 2D material recognition model, and then fine-tune it on our dataset.

The biggest challenge, however, is we have 4D data instead of 2D. In other words, we need to find good representations for 4D light-field images that are compatible with 2D CNN models. We thus implement a number of different architectures and report their performance. The results may be used as baselines, and might be useful for designing

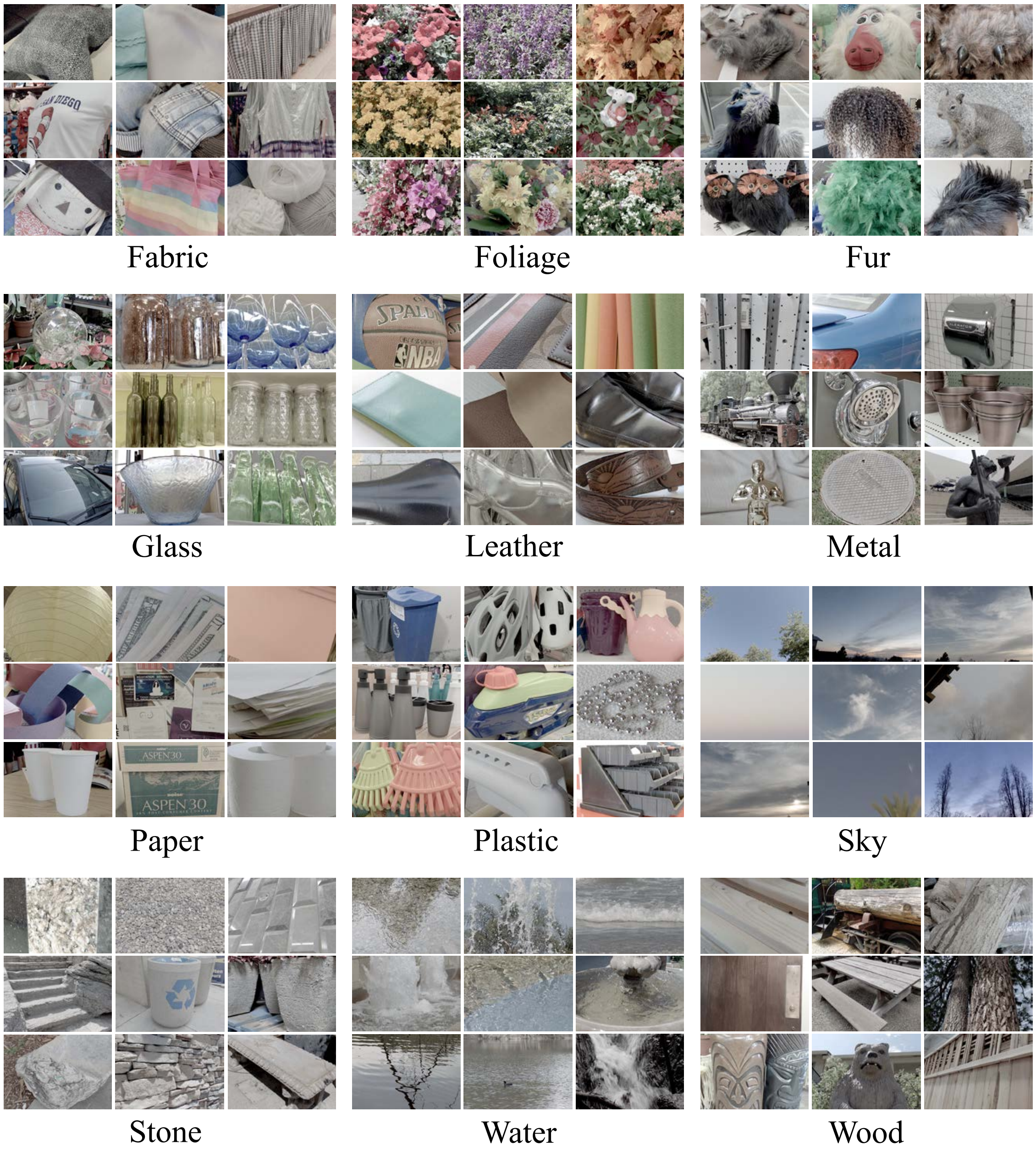


Figure 5.2: *Example images in our dataset. Each class contains 100 images.*

other learning-based methods for light-fields in the future. In our implementation, the best performing methods (angular filter and 4D filter) achieve 77% classification accuracy on the extracted patches, which is 7% higher than using 2D images only. Details of each

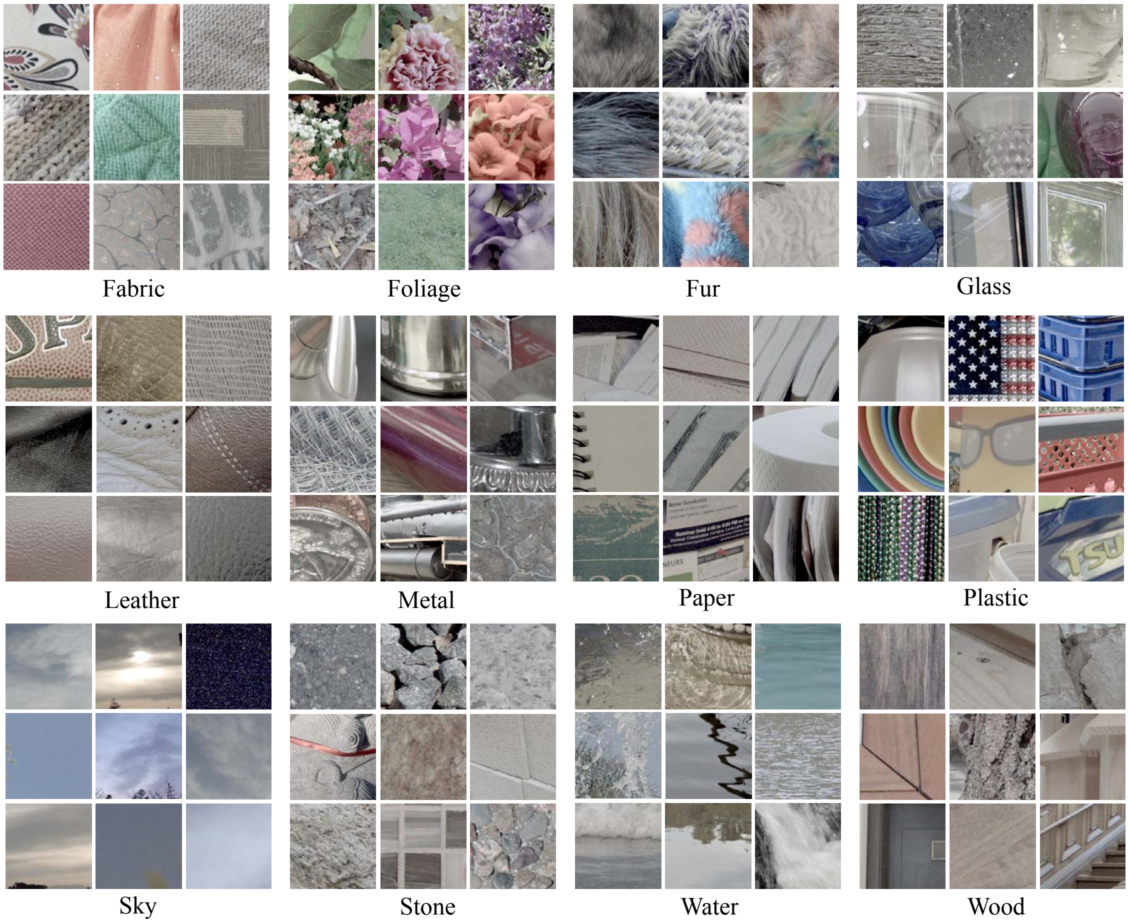
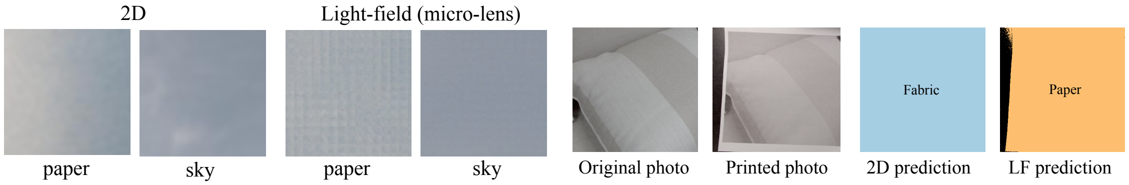


Figure 5.3: *Example image patches for the 12 materials in our dataset. Each class contains roughly 2500 image patches.*

architecture are described below.

2D average First and the simplest, we input each image independently and average the results across different views. This, however, definitely does not exploit the implicit information inside light-field images. It is also time consuming, where the time complexity grows linearly with angular size.

Viewpool Second, we leverage the recent "viewpool" method proposed in [115] (Fig. 5.5a). First, each view is passed through the convolutional part of the network separately. Next, they are aggregated at a max view-pooling layer, and then sent through the remaining (fully-connected) part of the network. This method combines information from different views at a higher level; however, max pooling only selects one input, so still only one view is chosen at each pixel. Also, since all views need to be passed through the first part



(a) 2D vs. LF images

(b) 2D vs. LF predictions

Figure 5.4: *Example benefits of using light-field images. (a) From the 2D images, it is difficult to distinguish between paper and sky. However, with the light-field images it becomes much easier. (b) We print out a picture of a pillow, and test both 2D and light-field models on the picture. The 2D model, without any reflectance information, predicts the material as fabric, while the light-field model correctly identifies the material as paper.*

of the network, the memory consumption becomes extremely large.

Stack Here, we stack all different views across their RGB channels before feeding them into the network, and change only the input channel of the first layer while leaving the rest of the architecture unchanged (Fig. 5.5b). This has the advantage that all views are combined earlier and thus takes far less memory.

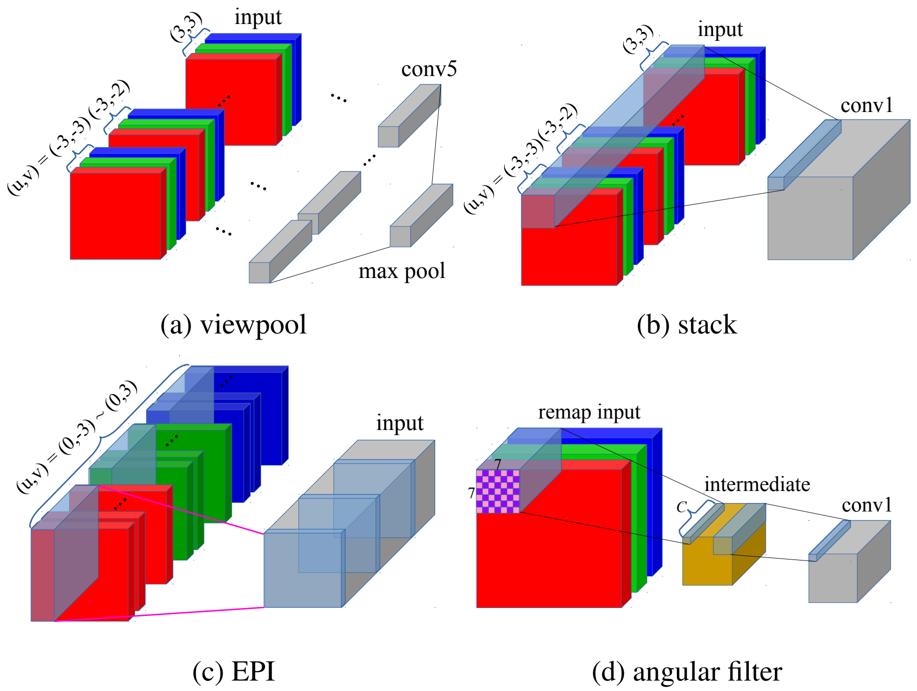
EPI For this method, we first extract the horizontal and vertical epipolar images (EPIs) for each row or column of the input light-field image. In other words, suppose the original 4D light-field is $L(x, y, u, v)$, where (x, y) are the spatial coordinates and (u, v) are the angular coordinates. We then extract 2D images from L by

$$\begin{aligned} L(x, y = y_i, u, v = 0) \quad & \forall i = 1, ..., h_s \\ L(x = x_j, y, u = 0, v) \quad & \forall j = 1, ..., w_s \end{aligned} \tag{5.1}$$

where $(u, v) = (0, 0)$ is the central view and (h_s, w_s) are the spatial size. These EPIs are then concatenated into a long cube and passed into the network (Fig. 5.5c). Again only the first layer of the pre-trained model is modified.

Angular filter on remap image The idea of applying filters on angular images was first proposed by Zhang et al. [87]. However, they only considered 2D angular images, while in our case we have 4D light-field images. Also, by incorporating the filters into the neural network, we can let the network learn the filters instead of manually designing them, which should achieve better performance.

For this method, we use the remap image instead of the standard image as input. A remap image replaces each pixel in a traditional 2D image with a block of angular pixels $h_a \times w_a$ from different views, where (h_a, w_a) are the angular size. The remap image is thus of size $(h_a \times h_s) \times (w_a \times w_s)$. It is also similar to the raw micro-lens image the Lytro camera captures; the only difference is that we eliminate the boundary viewpoints where the viewing angles are very oblique.



(a) viewpool

(b) stack

(c) EPI

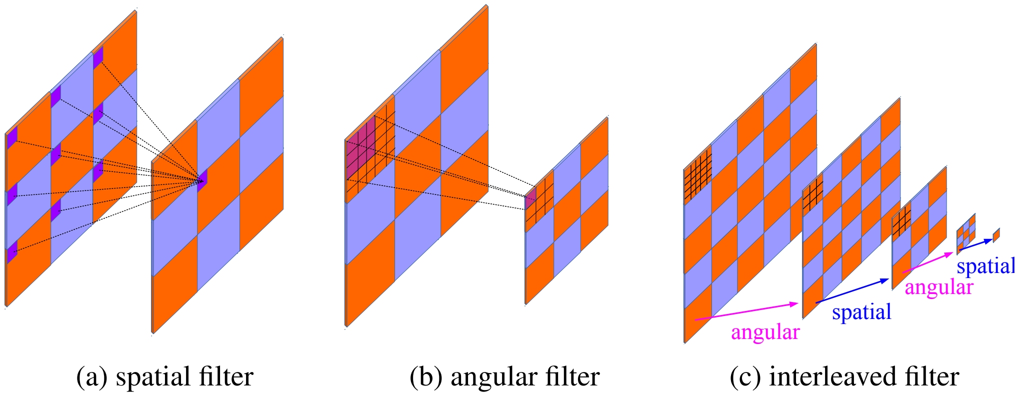
(d) angular filter

Figure 5.5: *Different CNN architectures for 4D light-field inputs. The RGB colors represent the RGB channels, while (u, v) denotes different angular coordinates (from $(-3, -3)$ to $(3, 3)$ in our experiments). (a) After each view is passed through the convolutional part, they are max pooled and combined into one view, and then sent to the fully connected part. (b) All views are stacked across the RGB channels to form the input. (c) The inputs are the horizontal and vertical EPIs concatenated together (only vertical ones are shown in the figure). (d) A 7×7 angular filter is first applied on the remap image. The intermediate output is then passed to the rest of the network.*

Before sending the remap image into the pre-trained network, we apply on it an angular filter of size $h_a \times w_a$ with stride h_a, w_a and output channel number C (Fig. 5.5d). After passing this layer, the image reduces to the same spatial size as the original 2D input. Specifically, let this layer be termed I (intermediate), then the output of this layer for each spatial coordinate (x, y) and channel j is

$$\ell^j(x,y) = g\Big(\sum_{i=r,g,b}\sum_{u,v} w_i^j(u,v)L^i(x,y,u,v)\Big) \quad \forall j = 1, ..., C \tag{5.2}$$

where L is the input (RGB) light-field, i, j are the channels for input light-field and layer I, $w_i^j(u, v)$ are the weights of the angular filter, and g is the rectified linear unit (ReLU). Afterwards, ℓ^j is passed into the pre-trained network.



(a) spatial filter (b) angular filter (c) interleaved filter

Figure 5.6: *(a)(b) New spatial and angular filters on a remap light-field image. The pooling feature is also implemented in a similar way. (c) By interleaving the angular and spatial filters (or vice versa), we mimic the structure of a 4D filter.*

4D filter Finally, since the light-field has a 4D structure, it becomes intuitive to apply a 4D filter to train the network. However, directly applying a 4D filter is problematic due to several reasons. First, a 4D filter contains far more parameters than a 2D filter. Even the smallest 4D filter ($3 \times 3 \times 3 \times 3$) contains the same number of parameters as a 9×9 2D filter. This is expensive in terms of both computation and memory. Second, a 4D filter is not present in any pre-trained network, so we need to train it from scratch, which means we cannot take advantage of the pre-trained model.

Our solution is to decompose a 4D filter into two consecutive 2D filters, a spatial filter and an angular filter, implemented on a remap image as shown in Fig. 5.6. The new spatial filter is similar to a traditional 2D filter, except that since we now work on "blocks" of pixels, it takes only one pixel from each block as input (Fig. 5.6a). This can be considered as a kind of "stride", but instead of stride in the output domain (where the filter itself moves), we have stride in the input domain (where the input moves while the filter stays). The angular filter, on the other hand, convolves an internal block normally just as a traditional 2D filter, but does not work across the block boundaries (Fig. 5.6b). By interleaving these two types of filters, we can approximate the effect of a 4D filter while not sacrificing the advantages stated above (Fig. 5.6c). The corresponding pooling structures are also implemented in the same way. To use the pre-trained models, the parameters from the original spatial filters are copied to the new spatial filters, while the angular filters are inserted between them and trained from scratch.

5.5 Experimental results

The various architectures in Sec. 5.4 are trained end-to-end using back-propagation. To ease the training of 4D light-fields, we initialize the weights with pre-trained 2D image models. The optimization is done with Stochastic Gradient Descent (SGD) using the Caffe toolbox [116]. The inputs are patches of spatial resolution 128×128 and angular resolution 7×7. To bring the spatial resolution to the normal size of 256×256 for VGG, we add a deconvolution layer at the beginning. We use a basic learning rate of 10^{-4}, while the layers that are modified or newly added use 10 times the basic learning rate. Below, we present a detailed performance comparison between different scenarios.

5.5.1 Comparison of different CNN architectures

We first compare the prediction accuracies for different architectures introduced in the previous section. Each method is tested 5 times on different randomly divided training and test sets to compute the performance average and variance. In Table 5.1, the first column (2D) is the result of the MINC VGG-16 model fine-tuned on our dataset, using only a single (central view) image. The remaining columns summarize the results of the other 4D architectures. Note that these 4D methods use more data as input than a 2D image; we will make a comparison where the methods take in an equal number of pixels in Sec. 5.5.3, and the results are still similar.

As predicted, averaging results from each view (2D avg) is only slightly better than using a 2D image alone. Next, the viewpool method actually performs slightly worse than using a 2D input; this indicates that the method is not suitable for light-fields, where the viewpoint changes are usually very small. The stack method and the EPI method achieve somewhat better performance, improving upon 2D inputs by 2-3%. The angular filter method achieves significant improvement over other methods; compared to using 2D input, it obtains about 7% gain. This shows the advantages of using light-fields rather than 2D images, as well as the importance of choosing the appropriate representation. The 4D filter method achieves approximately the same performance as the angular filter method. However, the angular filter method consumes much less memory, so it will be used as the primary comparison method in the following. The performance of each material class for the angular filter method is detailed in Table 5.2.

To further test the angular filter method, we compare performances by varying three parameters: the filter location, the filter size, and the number of output channels of the angular filter. First, we apply the angular filter at different layers of the VGG-16 network, and compare their performance. The classification accuracies when the filter is applied on layer 1 and layer 2 are 76.6% and 73.7%, respectively. Compared with applying it on the input directly (77.8%), we can see that the performance is better when we combine

Architecture	2D	2D avg	viewpool	stack	EPI	angular	4D
Accuracy (%)	$70.2_{\pm 1.0}$	$70.5_{\pm 0.9}$	$70.0_{\pm 1.0}$	$72.8_{\pm 1.1}$	$72.3_{\pm 1.0}$	$\mathbf{77.0}_{\pm \mathbf{1.1}}$	$\mathbf{77.0}_{\pm \mathbf{1.1}}$

Table 5.1: *Classification accuracy (average and variance) for different architectures. The 2D average method is only slightly better than using a single 2D image; the viewpool method actually performs slightly worse. The stack method and the EPI method both achieve better results. Finally, the angular filter method and the 4D filter method obtain the highest accuracy.*

Fabric: 65.5%	Foliage: 92.5%	Fur: 77.9%	Glass: 65.2%
Leather: 91.1%	Metal: 73.5%	Paper: 60.4%	Plastic: 50.0%
Sky: 98.2%	Stone: 87.1%	Water: 92.0%	Wood: 72.6%

Table 5.2: *Patch accuracy by category for the angular filter method.*

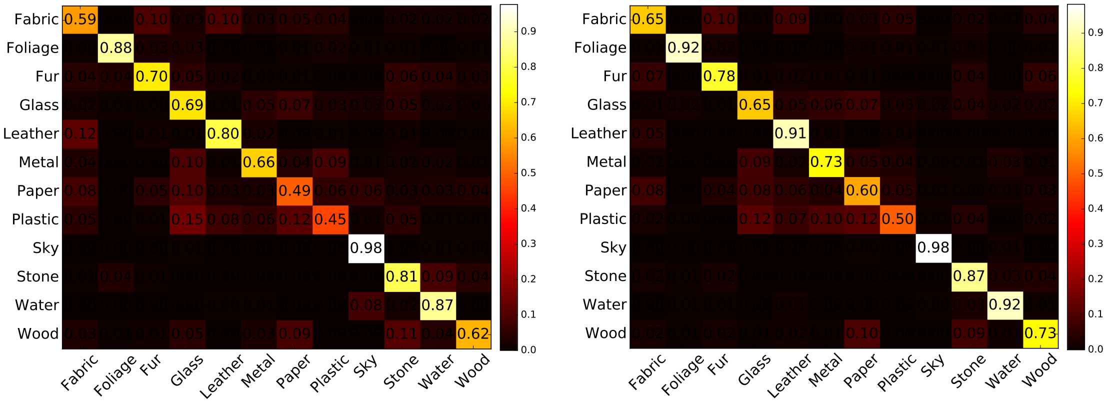
different views earlier. This also agrees with our findings on the viewpool method and 4D method. Next, we decompose the 7×7 angular filter into smaller filters. The accuracies for three consecutive 3×3 filters and a 5×5 filter followed by a 3×3 filter are 74.8% and 73.6%, respectively. It can be seen that making the filters smaller does not help improve the performance. One reason might be that in contrast to the spatial domain, where the object location is not important, in the angular domain the location actually matters (e.g. the upper-left pixel has a different meaning from the lower-right pixel), so a larger filter can better capture this information. Finally, we vary the number of output channels of the angular filter. Since the filter is directly applied on the light-field input, this can be considered as a "compression" of the input light-field. The fewer channels we output, the more compression we achieve using these filters. We test the number from 3 (all views compressed into one view) to 147 (no compression is made), and show the results in Table 5.3. It can be seen that the performance has a peak at 64 channels. We hypothesize that with fewer channels, the output might not be descriptive enough to capture variations in our data, but a much larger number of channels leads to overfitting due to the resulting increase in number of parameters.

5.5.2 Comparison between 2D and light-field results

The confusion matrices for both 2D and light-field methods (using the angular filter method) are shown in Fig. 5.7, and a graphical comparison is shown in Fig. 5.8a. Relative to 2D images, using light-fields achieves the highest performance boost on leather, paper

Number of channels	3	16	32	64	128	147
Accuracy	71.6%	74.8%	76.7%	***77.8%***	73.6%	72.8%

Table 5.3: *Number of output channels of the angular filter architecture. As we can see, using more channels increases the performance up to some point (64 channels), then the performance begins to drop, probably due to overfitting. This may also be related to light-field compression, where we do not need the entire 49×3 input channels and can represent them in fewer channels for certain purposes.*



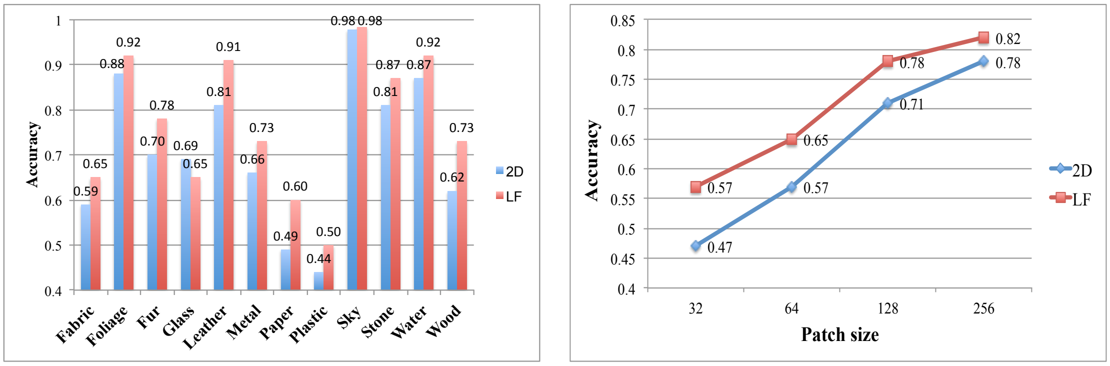
(a) 2D

(b) light-field

Figure 5.7: *Confusion matrix comparison between 2D and light-field results.*

and wood, with absolute gains of over 10%. This is probably because the appearances of these materials are determined by complex effects such as subsurface scattering or interreflections, and multiple views help in disambiguating these effects. Among all the 12 materials, only the performance for glass drops. This is probably because the appearance of glass is often dependent on the scene rather than on the material itself. Figure 5.9 shows some examples that are misclassified using 2D inputs but predicted correctly using light-fields, and vice versa. We observe that light-fields perform the best when the object information is missing or vague, necessitating reliance only on local texture or reflectance. On the other hand, the 2D method often generates reasonable results if the object category in the patch is clear.

Next, we change the patch size for both methods, and test their accuracies to see the effect of patch size on performance gain. We tried patch sizes 32, 64, 128 and 256 (Fig. 5.8b). It is observed that as we shrink the patch size from 128, the absolute gain steadily increases, from 7% to 10%. If we look at the relative gain, it is growing even



(a) accuracy by category

(b) accuracy vs. patch sizes

Figure 5.8: *Prediction accuracy comparison for using 2D and light-field inputs. (a) We first show the accuracies for each category. It can be seen that using light-fields achieves the highest performance boost on leather, paper and wood, obtaining absolute gains of over 10%. On the other hand, only the performance of glass drops. (b) Next, we vary the input patch size and test the performance again. It can be seen that as the patch size becomes smaller, the gain steadily increases.*

more rapidly, from about 10% at size 128 to 20% at size 32. At size 256 the absolute gain becomes smaller. A possibility is that at this scale, the object in the patch usually becomes apparent, and this information begins to dominate over the reflectance information. Therefore, the benefits of light-fields are most pronounced when using small patches. As the patch becomes smaller and smaller, it becomes harder and harder to recognize the object, so only local texture and reflectance information is available. Also note that although increasing the patch size will lead to better accuracy, it will also reduce the output resolution for full scene classification, so it is a tradeoff and not always better. Finally, while we have shown a significant increase in accuracy from 2D to 4D material recognition, once the dataset is published, our approach can still be improved by future advances that better exploit the full structure of 4D light-field data.

5.5.3 Comparison between spatial/angular resolution

Since light-field images contain more views, which results in an effectively larger number of pixels than 2D images, we also perform an experiment where the two inputs have the same number of pixels. Specifically, we extract the light-field image with a spatial resolution of 128×128 and an angular resolution of 4×4, and downsample the image in the spatial resolution by a factor of 4. This results in a light-field image of the same size as an original 2D image. The classification results for using original 2D input

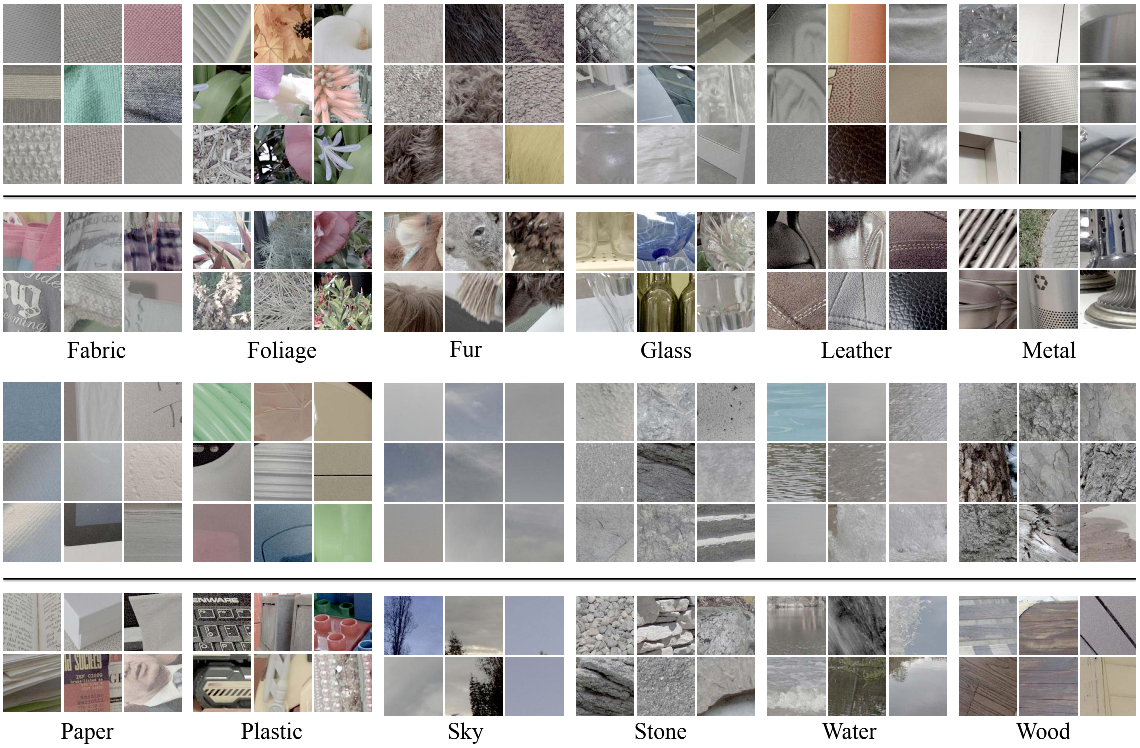


Figure 5.9: *Prediction result discrepancy between 2D and light-field inputs. The first* 3×3 *grids show example patches that are predicted correctly using LF inputs, but misclassified using 2D inputs. The second grids show the opposite situation, where 2D models output the correct class but LF models fail. We found that the LF model performs the best when the object information is missing or vague, so we can only rely on the local texture, viewpoint change or reflectance information.*

and this downsampled light-field are 70.7% and 75.2% respectively. Comparing with the original light-field results (77.8%), we observe that reducing the spatial resolution lowers the prediction accuracy, but it still outperforms 2D inputs by a significant amount.

5.5.4 Results on other datasets

Finally, to demonstrate the generality of our model, we test it on other datasets. Since no light-field material datasets are available, we test on the synthesized BTF database [97]. The database captures a large number of different viewing and lighting directions on 84 instances evenly classified into 7 materials. From the database we can render arbitrary views by interpolation on the real captured data. We thus render light-field images and evaluate our model on these rendered images.

First, directly applying our model on the BTF database already achieves 65.2% classification accuracy (for the materials that overlap). Next, since the BTF database contains different material categories from our dataset, we use our models to extract the 4096-dimensional output of the penultimate fully connected layer. This is the vector that is used to generate the final class probability in the network, and acts as a feature descriptor of the original input. We then use this feature descriptor to train an SVM. We pick two-thirds of the BTF dataset as training set and the rest as test set. The results for using 2D and light-field inputs are 59.8% and 63.7% respectively. Note that light-field inputs achieve about 4% better performance than using 2D inputs. Considering that the rendered images may not look similar to the real images taken with a Lytro camera, this is a somewhat surprising result. Next, we fine-tune our models on the training set, and test the performance on the test set again. The results for using 2D and light-field inputs are 67.7% and 73.0% respectively. Again using light-fields achieves more than 5% performance boost. These results demonstrate the generality of our models.

5.5.5 Full scene material segmentation

Finally, we convert our patch model to a fully convolutional model and test it on an entire image to perform material segmentation. We do not directly train a fully convolutional network (FCN) since we find it very unstable and the training loss seldom converges. Instead, we first train our model on image patches as described previously, convert it to a fully convolutional model, and then fine-tune it on entire images. To train on a full image, we add another material class to include all other materials that do not fall into any of the 12 classes in our dataset. We repeat this process for both our models of patch size 256 and 128 to get two corresponding FCN models, and combine their results by averaging their output probability maps. Finally, as the probability map is low-resolution due to the network stride, we use edge-aware upsampling [117] to upsample the probability map to the same size as the original image. The per pixel accuracy for FCN prediction before and after the guided filter is 77.0% and 79.9%, respectively. The corresponding accuracies for 2D models are 70.1% and 73.7%, after we apply the same procedure. Note that our method still retains 6-7% boost compared with 2D models. Example results for both methods are shown in Fig. 5.10.

5.6 Conclusion

We introduce a new light-field dataset in this chapter. Our dataset is the first one acquired with the Lytro Illum camera, and contains 1200 images, which is much larger than all previous datasets. Since light-fields can capture different views, they implicitly contain the reflectance information, which should be helpful when classifying materials. In view of this, we exploit the recent success in deep learning approaches, and train a CNN on this dataset to perform material recognition. To utilize the pre-trained 2D models, we implement a number of different architectures to adapt them to light-fields, and propose a "decomposed" 4D filter. These architectures provide insights to light-field researchers interested in adopting CNNs, and may also be generalized to other tasks involving light-fields in the future. Our experimental results demonstrate that we can benefit from using 4D light-field images, obtaining an absolute gain of about 7% in classification accuracy compared with using a single view alone. Finally, although we utilize this dataset for material recognition, it can also spur research towards other applications that combine learning techniques and light-field imagery.

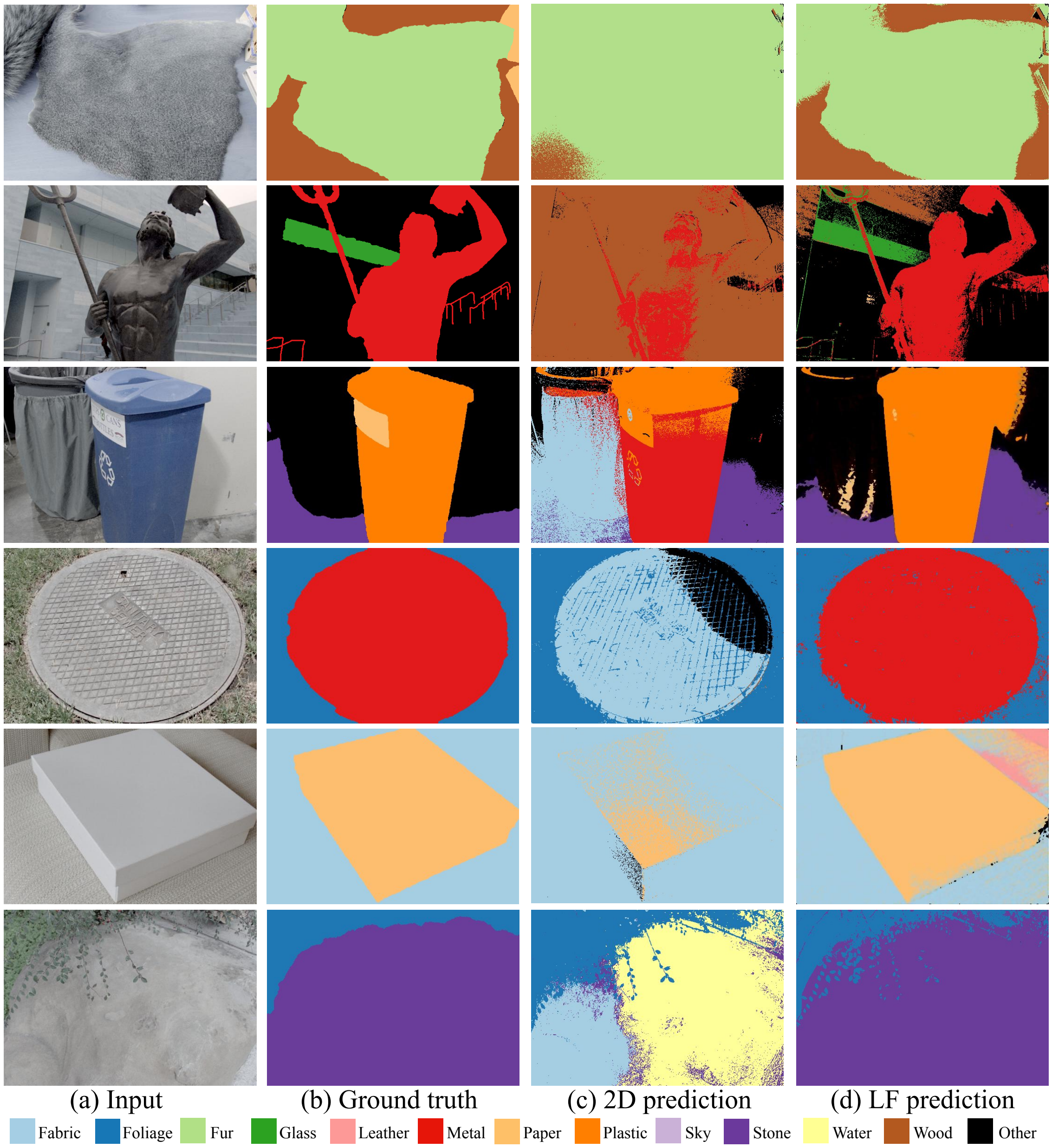

(a) Input (b) Ground truth (c) 2D prediction (d) LF prediction

Figure 5.10: *Full scene material classification examples. Bottom: legend for material colors. Compared with using 2D inputs, we can see that our light-field method produces more accurate prediction results.*

Chapter 6

Conclusion

In this thesis, we demonstrate that by going beyond the photo-consistency assumption, we are able to obtain better depth maps, estimate spatially-varying reflectances, and determine material types, therefore reconstructing three important properties of the scene. These recovered properties can then be used to improve the results for other applications, such as semantic segmentation, object detection, or scene classification, etc.

In Ch. 3, we propose a depth estimation algorithm which handles occlusion explicitly, so we are able to get sharp boundaries around occlusions. Moreover, we can also predict occlusion boundaries, to further regularize our depth map or be used for other applications.

Next, in Ch. 4 we look at a different problem which also fails photo-consistency – glossy objects. For these kinds of objects, their intensities will vary once the viewpoint is changed, so photo-consistency cannot be easily applied. To estimate the shapes for these objects, we derive a spatially varying BRDF-invariant equation. Using this equation, shapes can be recovered, and reflectances can also be estimated afterwards.

Since we have successfully demonstrated the ability to extract reflectance from light-field images, in Ch. 5 we adopt another learning-based approach to use this information for material recognition. We design several new neural network architectures to utilize this extra information in light-field images, and train the networks end-to-end to directly predict the material classification results. Our light-field material classifier performs much better than 2D classifiers, resulting in 7% absolute accuracy gain.

We believe these works, by relaxing the previous assumptions, solve important problems in light-field imaging, and can be very useful for further researches in this area. Below is a short summary of each work in this thesis.

6.1 Summary

Occlusions Most previous depth estimation approaches assume photo-consistency, and thus obtain smooth transition boundaries around occlusions. For our occlusion-aware depth estimation, we show that although not all camera views will follow photo-consistency, approximately half the views will still be un-occluded, so photo-consistency can still be applied on these views. Based on this observation, we propose an occlusion model, which can be used to break the camera views into two groups, one that follows photo-consistency and one that does not, so depth can be estimated only using the group that follows photo-consistency. As demonstrated by the experiments, this dramatically improved the estimated depths around occlusion boundaries. In addition to improved depth maps, occlusion boundaries can then also be predicted by combining the depth cue, the correspondence cue and the refocus cue, which can then be used for other applications such as segmentation.

Glossy objects For glossy objects, we build upon a traditional optical flow framework and develop a systematic approach for estimating shape and reflectance simultaneously instead of in an alternating fashion. We show that although in general the system of equations is ambiguous due to the new introduced unknowns, we can break the ambiguity by assuming a half-angle BRDF model, which is commonly adopted. Using this assumption, we are able to derive a spatially varying BRDF-invariant equation, which only depends on object shape and is independent of the BRDFs. Since shape and reflectance are now decoupled, we are able to solve them without using an alternating approach. However, solving this equation requires initial conditions, and may not be realistic for many scenarios. Instead, by applying a locally polynomial shape prior, we can solve this equation more easily and robustly, leading to results less sensitive to noise. The reflectance can then also be estimated directly once shape is recovered. This also enables us to deal with spatially-varying BRDF materials, which was not achieved by previous work.

Materials Finally, we adopt a learning-based approach to extract reflectance and use it for classifying materials. We collect a new light-field dataset and design several new network architectures to utilize the extra information in the light-field images. Our dataset is the first mid-size light-field dataset, with ground truth material labels, which is two orders of magnitude larger than previous datasets. We also propose a novel decomposed 4D filter, which successfully extracts the hidden reflectance information in light-fields to improve the material classification results. Experimental results demonstrate that our framework outperforms using 2D images alone. The collected dataset and architectures can also be used for other applications and research tasks as well.

6.2 Future work

Given our current work, there are several potential directions we can pursue further. Below we discuss about some possibilities.

Depth for reflective/transparent objects We have developed a framework for estimating depths for glossy objects. However, for pure reflective or mirror-like objects, estimating their shape is a different problem. Again, for these kinds of objects, photo-consistency certainly does not hold, and new equations must be derived to handle this case and find correspondences between different views [118, 119]. Furthermore, estimating shapes for transparent or semi-transparent objects would require new assumptions and careful treatment [120]. These problems are beyond the scope of this thesis, and we leave them to future work.

Shape ambiguity space In Chapter 4, we show that without any prior knowledge of the shape, the shape is ambiguous in the sense that we have more unknowns than the equations we have. It would be an interesting direction to see what different shapes can result in the same set of images when viewed from different perspectives, and find the relationship between this family of shapes, similar to the bas-relief ambiguity [121].

Illumination estimation The image intrinsic decomposition problem estimates shape, reflectance and lighting all at once from the given images. So far most existing algorithms can only estimate two of them [77], or have heavy prior assumptions on the given scene [82]. Our approach in Chapter 4 also currently assumes lighting is given, and tries to solve shape and reflectance. A potential future work is to also estimate lighting of the scene jointly, without resorting to impractical assumptions.

View synthesis Given the inherent trade-off between the spatial and angular resolution of light-field cameras, and the recent emerging popularity of virtual reality (VR), a very promising direction of light-field research is synthesizing novel views given sparse input views [122, 11]. A possible future work would be to extend the view synthesis to unstructured light-fields, where the cameras are organized in an unstructured way instead of a regular grid.

Hybrid imaging system Combining cameras of two different types to complement each other has also been proposed before [123, 124, 125]. In particular, an interesting line of works is to combine light-field cameras with DSLRs to increase either the spatial resolution [6] or the temporal resolution [126] of light-field cameras. Increasing both the resolutions is a potential next step to pursue further in this field.

Light-field videos With current Lytro consumer cameras, it is hard to capture light-field videos due to the huge required bandwidth. In fact, the Lytro ILLUM camera can only take at most three images per second. However, recent work has demonstrated the possibility to capture consumer light-field videos using a hybrid imaging system [126]. In the future, as the technology improves, it might become easier for consumers to obtain light-field videos without too much effort.

With the availability of light field videos, we would like to extend previous visual understanding works to videos. For example, recovering temporally consistent depth or material types, or other applications such as estimating saliency [91] or matting [127] for dynamic objects and scenes.

Appendix A

Derivation Details of the SVBRDF-Invariant Equation

A.1 Derivation of $\nabla_{\mathbf{v}}\rho$

Suppose $\rho = (\rho_d(\mathbf{x}, \mathbf{n}, \mathbf{s}) + \rho_s(\mathbf{x}, \hat{\mathbf{n}}^\top \hat{\mathbf{h}})) \cdot (\hat{\mathbf{n}}^\top \hat{\mathbf{s}})$, where $\hat{\mathbf{n}}^\top \hat{\mathbf{s}}$ is the cosine falloff term. Since $\hat{\mathbf{n}}^\top \hat{\mathbf{s}}$ is independent of $\mathbf{v}$, it just carries over the entire derivation and will be omitted in what follows. By the chain rule we have

$$\begin{aligned}\nabla_{\mathbf{v}}\rho &= \frac{\partial \rho_s}{\partial(\hat{\mathbf{n}}^\top \hat{\mathbf{h}})} \frac{\partial(\hat{\mathbf{n}}^\top \hat{\mathbf{h}})}{\partial \mathbf{v}} = \rho_s' \frac{\partial(\hat{\mathbf{n}}^\top \hat{\mathbf{h}})}{\partial \mathbf{v}} = \rho_s' \frac{\partial(\hat{\mathbf{n}}^\top \hat{\mathbf{h}})}{\partial \hat{\mathbf{h}}} \frac{\partial \hat{\mathbf{h}}}{\partial \mathbf{v}} \\ &= \rho_s' \hat{\mathbf{n}}^\top \frac{\partial \hat{\mathbf{h}}}{\partial \mathbf{v}} = \rho_s' \hat{\mathbf{n}}^\top \frac{\partial \hat{\mathbf{h}}}{\partial \mathbf{h}} \frac{\partial \mathbf{h}}{\partial \hat{\mathbf{v}}} \frac{\partial \hat{\mathbf{v}}}{\partial \mathbf{v}}\end{aligned} \tag{A.1}$$

Recall that for a vector $\mathbf{w}$, $\partial \hat{\mathbf{w}} / \partial \mathbf{w} = (\mathbf{I} - \hat{\mathbf{w}}\hat{\mathbf{w}}^\top)/\|\mathbf{w}\|$. Then

$$\frac{\partial \hat{\mathbf{h}}}{\partial \mathbf{h}} = \frac{\mathbf{I} - \hat{\mathbf{h}}\hat{\mathbf{h}}^\top}{\|\hat{\mathbf{s}} + \hat{\mathbf{v}}\|}, \quad \frac{\partial \hat{\mathbf{v}}}{\partial \mathbf{v}} = \frac{\mathbf{I} - \hat{\mathbf{v}}\hat{\mathbf{v}}^\top}{\|\mathbf{v}\|} \tag{A.2}$$

And

$$\frac{\partial \mathbf{h}}{\partial \hat{\mathbf{v}}} = \frac{\partial(\hat{\mathbf{s}}+\hat{\mathbf{v}})}{\partial \hat{\mathbf{v}}} = \mathbf{I} \tag{A.3}$$

So (A.1) can be simplified as

$$\nabla_{\mathbf{v}}\rho = \rho_s' \hat{\mathbf{n}}^\top \frac{\mathbf{I} - \hat{\mathbf{h}}\hat{\mathbf{h}}^\top}{\|\hat{\mathbf{s}} + \hat{\mathbf{v}}\|} \cdot \mathbf{I} \cdot \frac{\mathbf{I} - \hat{\mathbf{v}}\hat{\mathbf{v}}^\top}{\|\mathbf{v}\|} \tag{A.4}$$

Let $\mathbf{H} \equiv (\mathbf{I} - \hat{\mathbf{h}}\hat{\mathbf{h}}^\top)(\mathbf{I} - \hat{\mathbf{v}}\hat{\mathbf{v}}^\top)$, and note that

$$\begin{aligned} \|\mathbf{v}\| &= \|(0, 0, -f)^\top - (x, y, z)^\top\| \\ &= \sqrt{x^2 + y^2 + (z + f)^2} \\ &= (1 + \beta z)\sqrt{u^2 + v^2 + f^2} \end{aligned} \tag{A.5}$$

then (A.4) becomes

$$\begin{aligned} \nabla_{\mathbf{v}}\rho &= \rho_s' \hat{\mathbf{n}}^\top \frac{\mathbf{H}}{\|\hat{\mathbf{s}} + \hat{\mathbf{v}}\|\|\mathbf{v}\|} \\ &= \rho_s' \hat{\mathbf{n}}^\top \frac{\mathbf{H}}{\|\hat{\mathbf{s}} + \hat{\mathbf{v}}\|(1 + \beta z)\sqrt{u^2 + v^2 + f^2}} \end{aligned} \tag{A.6}$$

which is the equation we used in (4.17). The following procedure is described in the main text. Finally, after expanding (4.19), the κ's in (4.20) are

$$\begin{aligned} \kappa_1 &= (\gamma_2 + \gamma_1 I_u - I_u)H_{12} - (\gamma_3 + \gamma_1 I_v - I_v)H_{11} \\ \kappa_2 &= \beta(\gamma_2 + \gamma_1 I_u)H_{12} - \beta(\gamma_3 + \gamma_1 I_v)H_{11} \\ \kappa_3 &= (\gamma_2 + \gamma_1 I_u - I_u)H_{22} - (\gamma_3 + \gamma_1 I_v - I_v)H_{21} \\ \kappa_4 &= \beta(\gamma_2 + \gamma_1 I_u)H_{22} - \beta(\gamma_3 + \gamma_1 I_v)H_{21} \\ \kappa_5 &= -(\gamma_2 + \gamma_1 I_u - I_u)H_{32} + (\gamma_3 + \gamma_1 I_v - I_v)H_{31} \\ \kappa_6 &= -\beta(\gamma_2 + \gamma_1 I_u)H_{32} + \beta(\gamma_3 + \gamma_1 I_v)H_{31} \end{aligned} \tag{A.7}$$

The mathematical solution to the PDE (4.20) is a parametric curve defined by

$$\begin{aligned} z(s) &= -\frac{\kappa_5}{\kappa_6} + c_1 e^{-\kappa_6 s} \\ x(s) &= \kappa_1 s + \kappa_2(-\frac{c_1}{\kappa_6}e^{-\kappa_6 s} - \frac{\kappa_5}{\kappa_6}s) + c_2 \\ y(s) &= \kappa_3 s + \kappa_4(-\frac{c_1}{\kappa_6}e^{-\kappa_6 s} - \frac{\kappa_5}{\kappa_6}s) + c_3 \end{aligned} \tag{A.8}$$

where c_1, c_2, c_3 are constants, and require some initial condition to be uniquely identified. Note that κ's are different for each pixel, which makes the problem even harder. Therefore, directly obtaining shape is not possible, and we refer to a polynomial shape prior, as introduced in the main text.

For cameras focused at some finite distance F, the new κ values become

$$\begin{aligned}
\kappa_1 &= (\gamma_2 + (\gamma_1 - \frac{\beta F}{1+\beta F})I_u)H_{12} - (\gamma_3 + (\gamma_1 - \frac{\beta F}{1+\beta F})I_v)H_{11} \\
\kappa_2 &= \beta(\gamma_2 + (\gamma_1 + \frac{1}{1+\beta F})I_u)H_{12} - \beta(\gamma_3 + (\gamma_1 + \frac{1}{1+\beta F})I_v)H_{11} \\
\kappa_3 &= (\gamma_2 + (\gamma_1 - \frac{\beta F}{1+\beta F})I_u)H_{22} - (\gamma_3 + (\gamma_1 - \frac{\beta F}{1+\beta F})I_v)H_{21} \\
\kappa_4 &= \beta(\gamma_2 + (\gamma_1 + \frac{1}{1+\beta F})I_u)H_{22} - \beta(\gamma_3 + (\gamma_1 + \frac{1}{1+\beta F})I_v)H_{21} \\
\kappa_5 &= -(\gamma_2 + (\gamma_1 - \frac{\beta F}{1+\beta F})I_u)H_{32} + (\gamma_3 + (\gamma_1 - \frac{\beta F}{1+\beta F})I_v)H_{31} \\
\kappa_6 &= -\beta(\gamma_2 + (\gamma_1 + \frac{1}{1+\beta F})I_u)H_{32} + \beta(\gamma_3 + (\gamma_1 + \frac{1}{1+\beta F})I_v)H_{31}
\end{aligned} \tag{A.9}$$

A.2 Derivation of n_x and n_y

Since $u = x/(1+\beta z)$ by (4.1), we can multiply both sides in (4.22) by $(1+\beta z)^2$ and get

$$\begin{aligned}
z(1+\beta z)^2 = a_1x^2 + a_2y^2 + a_3xy + a_4x(1+\beta z) \\
+ a_5y(1+\beta z) + a_6(1+\beta z)^2
\end{aligned} \tag{A.10}$$

Taking derivatives of both sides, the above equation becomes

$$\begin{aligned}
(1+\beta z)^2\delta z + 2\beta z(1+\beta z)\delta z = 2a_1x\delta x + 2a_2y\delta y \\
+a_3x\delta y + a_3y\delta x + a_4(1+\beta z)\delta x + a_4\beta x\delta z \\
+a_5(1+\beta z)\delta y + a_5\beta y\delta z + 2\beta a_6(1+\beta z)\delta z
\end{aligned} \tag{A.11}$$

After some rearrangement, we can write the normal n_x as,

$$\begin{aligned}
n_x &= \frac{\partial z}{\partial x} \\
&= \frac{2a_1x + a_3y + a_4(1+\beta z)}{(1+\beta z)^2 + 2\beta z(1+\beta z) - a_4\beta x - a_5\beta y - 2\beta a_6(1+\beta z)} \\
&= \frac{2a_1u + a_3v + a_4}{1 + 3\beta z - a_4\beta u - a_5\beta v - 2\beta a_6} \\
&= \frac{n_u}{1 + 3\beta z - a_4\beta u - a_5\beta v - 2\beta a_6}
\end{aligned} \tag{A.12}$$

Similarly, n_y can be written as

$$n_y = \frac{\partial z}{\partial y} = \frac{n_v}{1 + 3\beta z - a_4\beta u - a_5\beta v - 2\beta a_6} \tag{A.13}$$

A.3 The formula of M

We provide the entire formula of the matrix M in (4.27) in this section. Starting from (4.26), which we redisplay here for convenience

$$\begin{aligned}(3\beta\kappa_6 u^4 + 2\kappa_2 u^3)a_1^2 + 2(3\beta\kappa_6 u^2 v^2 + \kappa_4 u^2 v + \kappa_2 u v^2)a_1 a_2 \\ +(\kappa_4 u^3 + 3\kappa_2 u^2 v + 6\beta\kappa_6 u^3 v)a_1 a_3 + \cdots + \\ (3\beta\kappa_6 v^4 + 2\kappa_4 v^3)a_2^2 + (\kappa_2 v^3 + 3\kappa_4 u v^2 + 6\beta\kappa_6 u v^3)a_2 a_3 \\ + \cdots + \beta\kappa_6 a_6^2 + (2\kappa_1 u + \kappa_6 u^2 + 3\beta\kappa_5 u^2)a_1 + \cdots + \kappa_5\end{aligned} \tag{A.14}$$

We can then factorize the above equation into the following matrix form for easier optimization

$$\begin{bmatrix} \mathbf{a}^\top & 1 \end{bmatrix} M \begin{bmatrix} \mathbf{a} \\ 1 \end{bmatrix} = 0 \tag{A.15}$$

Since M is symmetric, we provide only half the matrix components here,

$$\begin{aligned}
M_{11} &= 3\beta\kappa_6 u^4 + 2\kappa_2 u^3,\\
M_{12} &= 3\beta\kappa_6 u^2 v^2 + \kappa_4 u^2 v + \kappa_2 u v^2,\\
M_{13} &= \kappa_4 u^3/2 + 3\kappa_2 u^2 v/2 + 3\beta\kappa_6 u^3 v,\\
M_{14} &= 3\beta\kappa_6 u^3 + 3\kappa_2 u^2/2,\\
M_{15} &= \kappa_4 u^2/2 + \kappa_2 uv + 3\beta\kappa_6 u^2 v,\\
M_{16} &= 2\beta\kappa_6 u^2 + \kappa_2 u,\\
M_{17} &= \kappa_1 u + \kappa_6 u^2/2 + 3\beta\kappa_5 u^2/2,\\
M_{22} &= 3\beta\kappa_6 v^4 + 2\kappa_4 v^3,\\
M_{23} &= \kappa_2 v^3/2 + 3\kappa_4 u v^2/2 + 3\beta\kappa_6 u v^3,\\
M_{24} &= \kappa_2 v^2/2 + \kappa_4 uv + 3\beta\kappa_6 u v^2,\\
M_{25} &= 3\beta\kappa_6 v^3 + 3\kappa_4 v^2/2,\\
M_{26} &= 2\beta\kappa_6 v^2 + \kappa_4 v,\\
M_{27} &= \kappa_3 v + \kappa_6 v^2/2 + 3\beta\kappa_5 v^2/2,\\
M_{33} &= 3\beta\kappa_6 u^2 v^2 + \kappa_4 u^2 v + \kappa_2 u v^2,\\
M_{34} &= \kappa_4 u^2/2 + \kappa_2 uv + 3\beta\kappa_6 u^2 v,\\
M_{35} &= \kappa_2 v^2/2 + \kappa_4 uv + 3\beta\kappa_6 u v^2,\\
M_{36} &= \kappa_4 u/2 + \kappa_2 v/2 + 2\beta\kappa_6 uv,\\
M_{37} &= (\kappa_3 u + \kappa_1 v + \kappa_6 uv + 3\beta\kappa_5 uv)/2,\\
M_{44} &= 3\beta\kappa_6 u^2 + \kappa_2 u,\\
M_{45} &= \kappa_4 u/2 + \kappa_2 v/2 + 3\beta\kappa_6 uv,\\
M_{46} &= \kappa_2/2 + 2\beta\kappa_6 u,\\
M_{47} &= (\kappa_1 + \kappa_6 u + 3\beta\kappa_5 u)/2,\\
M_{55} &= 3\beta\kappa_6 v^2 + \kappa_4 v,\\
M_{56} &= \kappa_4/2 + 2\beta\kappa_6 v,\\
M_{57} &= (\kappa_3 + \kappa_6 v + 3\beta\kappa_5 v)/2,\\
M_{66} &= \beta\kappa_6,\\
M_{67} &= (\kappa_6 + \beta\kappa_5)/2,\\
M_{77} &= \kappa_5
\end{aligned} \tag{A.16}$$

Bibliography

[1] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of SIGGRAPH*, 1996.

[2] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," *Computer Science Technical Report CSTR*, vol. 2, no. 11, 2005.

[3] "Lytro decoding toolbox." `http://forums.lytro.com`.

[4] T. E. Bishop, S. Zanetti, and P. Favaro, "Light field superresolution," in *Proceedings of the IEEE International Conference on Computational Photography (ICCP)*, pp. 1–9, 2009.

[5] D. Cho, M. Lee, S. Kim, and Y.-W. Tai, "Modeling the calibration pipeline of the lytro camera for high quality light-field image reconstruction," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3280–3287, 2013.

[6] V. Boominathan, K. Mitra, and A. Veeraraghavan, "Improving resolution and depth-of-field of light field cameras using a hybrid imaging system," in *Proceedings of the IEEE International Conference on Computational Photography (ICCP)*, pp. 1–10, 2014.

[7] A. Levin and F. Durand, "Linear view synthesis using a dimensionality gap light field prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1831–1838, 2010.

[8] L. Shi, H. Hassanieh, A. Davis, D. Katabi, and F. Durand, "Light field reconstruction using sparsity in the continuous fourier domain," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 1, pp. 12:1–12:13, 2014.

[9] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar, "Compressive light field photography using overcomplete dictionaries and optimized projections," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, pp. 46:1–46:12, 2013.

[10] Z. Zhang, Y. Liu, and Q. Dai, "Light field from micro-baseline image pair," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3800–3809, 2015.

[11] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, 2016.

[12] K. Mitra and A. Veeraraghavan, "Light field denoising, light field superresolution and stereo camera based refocussing using a GMM light field patch prior," pp. 22–28, 2012.

[13] S. Wanner and B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 36, no. 3, pp. 606–619, 2014.

[14] Y. Yoon, H. G. Jeon, D. Yoo, J. Y. Lee, and I. S. Kweon, "Learning a deep convolutional network for light-field image super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, pp. 57–65, 2015.

[15] S. Wanner and B. Goldluecke, "Globally consistent depth labeling of 4D light fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[16] I. Tosic and K. Berkner, "Light field scale-depth space transform for dense depth estimation," pp. 435–442, 2014.

[17] S. Zhang, H. Sheng, C. Li, J. Zhang, and Z. Xiong, "Robust depth estimation for light field via spinning parallelogram operator," *Computer Vision and Image Understanding*, vol. 145, pp. 148–159, 2016.

[18] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[19] S. Heber and T. Pock, "Shape from light field meets robust pca," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 751–767, 2014.

[20] H.-G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y.-W. Tai, and I. S. Kweon, "Accurate depth map estimation from a lenslet light field camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[21] "Lytro redefines photography with light field cameras. Press release, Jun 2011." http://www.lytro.com.

[22] C. Perwass and L. Wietzke, "Single lens 3D-camera with extended depth-of-field," in *Proceedings of IS&T/SPIE Electronic Imaging*, 2012.

[23] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proceedings of SIGGRAPH*, 1996.

[24] E. H. Adelson and J. Y. A. Wang, "Single lens stereo with a plenoptic camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 14, no. 2, pp. 99–106, 1992.

[25] T.-C. Wang, A. A. Efros, and R. Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras.," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[26] T.-C. Wang, A. A. Efros, and R. Ramamoorthi, "Depth estimation with occlusion modeling using light-field cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 38, no. 11, pp. 2170–2181, 2016.

[27] C. Chen, H. Lin, Z. Yu, S. B. Kang, and J. Yu, "Light field stereo matching using bilateral statistics of surface cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1518–1525, 2014.

[28] Z. Yu, X. Guo, H. Ling, A. Lumsdaine, and J. Yu, "Line assisted light field triangulation and stereo matching," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[29] V. Kolmogorov and R. Zabih, "Multi-camera scene reconstruction via graph cuts," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2002.

[30] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon, "Global stereo reconstruction under second-order smoothness priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 31, no. 12, pp. 2115–2128, 2009.

[31] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing binary mrfs via extended roof duality," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[32] M. Bleyer, C. Rother, and P. Kohli, “Surface stereo with soft segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[33] Y. Wei and L. Quan, “Asymmetrical occlusion handling using graph cut for multi-view stereo,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[34] S. McCloskey, “Masking light fields to remove partial occlusion,” in *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, pp. 2053–2058, 2014.

[35] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. H. Gross, “Scene reconstruction from high spatio-angular resolution light fields.,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 73, 2013.

[36] M. W. Tao, P. P. Srinivasan, J. Malik, S. Rusinkiewicz, and R. Ramamoorthi, “Depth from shading, defocus, and correspondence using light-field angular coherence,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[37] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 26, no. 9, pp. 1124–1137, 2004.

[38] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 23, no. 11, pp. 1222–1239, 2001.

[39] V. Kolmogorov and R. Zabin, “What energy functions can be minimized via graph cuts?,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 26, no. 2, pp. 147–159, 2004.

[40] S. Wanner, S. Meister, and B. Goldlücke, “Datasets and benchmarks for densely sampled 4D light fields,” in *Annual Workshop on Vision, Modeling and Visualization*, 2013.

[41] “3D modeling, animation, and rendering software.” `http://www.autodesk.com/products/3ds-max`.

[42] B. Curless and M. Levoy, “A volumetric method for building complex models from range images,” in *Proceedings of SIGGRAPH*, 1996.

[43] V. Krishnamurthy and M. Levoy, “Fitting smooth surfaces to dense polygon meshes,” in *Proceedings of SIGGRAPH*, 1996.

[44] G. Turk and M. Levoy, “Zippered polygon meshes from range images,” in *Proceedings of SIGGRAPH*, 1994.

[45] “Free 3ds models.” http://www.free-3ds-models.com.

[46] C.-K. Liang and R. Ramamoorthi, “A light transport framework for lenslet light field cameras,” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 2, p. 16, 2015.

[47] B. K. Horn and B. G. Schunck, “Determining optical flow,” in *Technical Symposium East*, 1981.

[48] B. D. Lucas, T. Kanade, *et al.*, “An iterative image registration technique with an application to stereo vision.,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 81, pp. 674–679, 1981.

[49] R. J. Woodham, “Photometric method for determining surface orientation from multiple images,” *Optical engineering*, vol. 19, no. 1, pp. 191139–191139, 1980.

[50] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, “A comparison and evaluation of multi-view stereo reconstruction algorithms,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[51] Y. Furukawa and J. Ponce, “Accurate, dense, and robust multiview stereopsis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 32, no. 8, pp. 1362–1376, 2010.

[52] M. Chandraker, “On shape and material recovery from motion,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[53] M. Chandraker, “What camera motion reveals about shape with unknown BRDF,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[54] M. Chandraker, “The information available to a moving observer on shape with unknown, isotropic BRDFs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2015.

[55] M. W. Tao, J.-C. Su, T.-C. Wang, J. Malik, and R. Ramamoorthi, “Depth estimation and specular removal for glossy surfaces using point and line consistency with light-field cameras,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 38, no. 6, pp. 1155–1169, 2016.

[56] T.-C. Wang, M. Chandraker, A. A. Efros, and R. Ramamoorthi, "SVBRDF-invariant shape and reflectance estimation from light-field cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[57] T.-C. Wang, M. Chandraker, A. A. Efros, and R. Ramamoorthi, "SVBRDF-invariant shape and reflectance estimation from light-field cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[58] W. Matusik, H. Pfister, M. Brand, and L. McMillan, "A data-driven reflectance model," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 759–76, 2003.

[59] M. W. Tao, T.-C. Wang, J. Malik, and R. Ramamoorthi, "Depth estimation for glossy surfaces with light-field cameras," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2014.

[60] S. Tominaga, "Surface identification using the dichromatic reflection model," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, no. 7, pp. 658–670, 1991.

[61] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Manhattan-world stereo," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[62] D. Gallup, J.-M. Frahm, and M. Pollefeys, "Piecewise planar and non-planar stereo for urban scene reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[63] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Schematic surface reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[64] T. E. Zickler, P. N. Belhumeur, and D. J. Kriegman, "Helmholtz stereopsis: Exploiting reciprocity for surface reconstruction," *International Journal of Computer Vision (IJCV)*, vol. 49, no. 2-3, pp. 215–227, 2002.

[65] T. Bonfort and P. Sturm, "Voxel carving for specular surfaces," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2003.

[66] R. Yang, M. Pollefeys, and G. Welch, "Dealing with textureless regions and specular highlights-a progressive space carving scheme using a novel photo-consistency measure," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2003.

[67] A. Treuille, A. Hertzmann, and S. M. Seitz, “Example-based stereo with general brdfs,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2004.

[68] H. Jin, S. Soatto, and A. J. Yezzi, “Multi-view stereo reconstruction of dense shape and complex appearance,” *International Journal of Computer Vision (IJCV)*, vol. 63, no. 3, pp. 175–189, 2005.

[69] T. Yu, N. Ahuja, and W.-C. Chen, “SDG cut: 3D reconstruction of non-lambertian objects using graph cuts on surface distance grid,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[70] A. Ngan, F. Durand, and W. Matusik, “Experimental analysis of BRDF models.,” *Rendering Techniques*, 2005.

[71] K. Nishino and S. Lombardi, “Directional statistics-based reflectance model for isotropic bidirectional reflectance distribution functions,” *Journal of the Optical Society of America A*, vol. 28, no. 1, pp. 8–18, 2011.

[72] F. Romeiro, Y. Vasilyev, and T. Zickler, “Passive reflectometry,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

[73] F. Romeiro and T. Zickler, “Blind reflectometry,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[74] M. Chandraker and R. Ramamoorthi, “What an image reveals about material reflectance,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

[75] J. Lawrence, A. Ben-Artzi, C. DeCoro, W. Matusik, H. Pfister, R. Ramamoorthi, and S. Rusinkiewicz, “Inverse shade trees for non-parametric material representation and editing,” *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 735–745, 2006.

[76] N. Alldrin, T. Zickler, and D. Kriegman, “Photometric stereo with non-parametric and spatially-varying reflectance,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[77] G. Oxholm and K. Nishino, “Shape and reflectance from natural illumination,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[78] J.-D. Durou, M. Falcone, and M. Sagona, "Numerical methods for shape-from-shading: A new survey with benchmarks," *Computer Vision and Image Understanding*, vol. 109, no. 1, pp. 22–43, 2008.

[79] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, "Shape-from-shading: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 21, no. 8, pp. 690–706, 1999.

[80] M. K. Johnson and E. H. Adelson, "Shape estimation in natural illumination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[81] J. T. Barron and J. Malik, "Color constancy, intrinsic images, and shape estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[82] J. T. Barron and J. Malik, "Shape, albedo, and illumination from a single image of an unknown object," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[83] A. Ecker and A. D. Jepson, "Polynomial shape from shading," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[84] Y. Xiong, A. Chakrabarti, R. Basri, S. J. Gortler, D. W. Jacobs, and T. Zickler, "From shading to local shape," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 37, no. 1, pp. 67–79, 2015.

[85] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis, *Geometrical considerations and nomenclature for reflectance*, vol. 160. US Department of Commerce, National Bureau of Standards Washington, DC, USA, 1977.

[86] K. J. Dana, B. Van Ginneken, S. K. Nayar, and J. J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM Transactions on Graphics (TOG)*, vol. 18, no. 1, pp. 1–34, 1999.

[87] H. Zhang, K. Dana, and K. Nishino, "Reflectance hashing for material recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[88] T.-C. Wang, J.-Y. Zhu, E. Hiroaki, M. Chandraker, A. A. Efros, and R. Ramamoorthi, "A 4D light-field dataset and CNN architectures for material recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[89] A. Adams, M. Levoy, V. Vaish, B. Wilburn, and N. Joshi, "Stanford light field archive." http://lightfield.stanford.edu/.

[90] A. Jarabo, B. Masia, A. Bousseau, F. Pellacini, and D. Gutierrez, "How do people edit light fields?," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 146–1, 2014.

[91] N. Li, J. Ye, Y. Ji, H. Ling, and J. Yu, "Saliency detection on light field," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[92] R. Raghavendra, K. B. Raja, and C. Busch, "Exploring the usefulness of light field cameras for biometrics: An empirical study on face and iris recognition," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 922–936, 2016.

[93] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh, "On the significance of real-world conditions for material classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2004.

[94] B. Caputo, E. Hayman, and P. Mallikarjuna, "Class-specific material categorisation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[95] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[96] W. Li and M. Fritz, "Recognizing materials from virtual examples," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[97] M. Weinmann, J. Gall, and R. Klein, "Material classification based on training data synthesized using a BTF database," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[98] L. Sharan, R. Rosenholtz, and E. Adelson, "Material perception: What can you see in a brief glance?," *Journal of Vision*, vol. 9, no. 8, pp. 784–784, 2009.

[99] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Opensurfaces: A richly annotated catalog of surface appearance," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 111, 2013.

[100] S. Bell, P. Upchurch, N. Snavely, and K. Bala, “Material recognition in the wild with the materials in context database,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[101] O. G. Cula and K. J. Dana, “3D texture recognition using bidirectional feature histograms,” *International Journal of Computer Vision (IJCV)*, vol. 59, no. 1, pp. 33–60, 2004.

[102] S. Lombardi and K. Nishino, “Single image multimaterial estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[103] C. Liu and J. Gu, “Discriminative illumination: Per-pixel classification of raw materials based on optimal projections of spectral BRDF,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 36, no. 1, pp. 86–98, 2014.

[104] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz, “Exploring features in a bayesian framework for material recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[105] D. Hu, L. Bo, and X. Ren, “Toward robust material recognition for everyday objects,” in *Proceedings of the British Machine Vision Conference (BMVC)*, 2011.

[106] G. Schwartz and K. Nishino, “Visual material traits: Recognizing per-pixel material context,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, 2013.

[107] X. Qi, R. Xiao, C.-G. Li, Y. Qiao, J. Guo, and X. Tang, “Pairwise rotation invariant co-occurrence local binary pattern,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 36, no. 11, pp. 2199–2213, 2014.

[108] M. Cimpoi, S. Maji, and A. Vedaldi, “Deep filter banks for texture recognition and segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[109] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[110] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[111] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[112] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning hierarchical features for scene labeling,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 35, no. 8, pp. 1915–1929, 2013.

[113] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and transferring mid-level image representations using convolutional neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[114] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[115] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multi-view convolutional neural networks for 3d shape recognition,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[116] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the ACM International Conference on Multimedia*, 2014.

[117] K. He, J. Sun, and X. Tang, “Guided image filtering,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[118] G. D. Canas, Y. Vasilyev, Y. Adato, T. Zickler, S. Gortler, and O. Ben-Shahar, “A linear formulation of shape from specular flow,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 191–198, 2009.

[119] Y. Vasilyev, T. Zickler, S. Gortler, and O. Ben-Shahar, “Shape from specular flow: Is one flow enough?,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2561–2568, 2011.

[120] Q. Shan, S. Agarwal, and B. Curless, “Refractive height fields from single and multiple images,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 286–293, 2012.

[121] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille, “The bas-relief ambiguity,” *International Journal of Computer Vision (IJCV)*, vol. 35, no. 1, pp. 33–44, 1999.

[122] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deepstereo: Learning to predict new views from the worldâĂŹs imagery," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5515–5524, 2016.

[123] H. S. Sawhney, Y. Guo, K. Hanna, R. Kumar, S. Adkins, and S. Zhou, "Hybrid stereo camera: an IBR approach for synthesis of very high resolution stereoscopic image sequences," in *Proceedings of SIGGRAPH*, pp. 451–460, 2001.

[124] P. Bhat, C. L. Zitnick, N. Snavely, A. Agarwala, M. Agrawala, M. Cohen, B. Curless, and S. B. Kang, "Using photographs to enhance videos of a static scene," in *Eurographics Symposium on Rendering (EGSR)*, pp. 327–338, 2007.

[125] T.-C. Wang, M. Srikanth, and R. Ramamoorthi, "Depth from semi-calibrated stereo and defocus," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3717–3726, 2016.

[126] T.-C. Wang, J.-Y. Zhu, N. K. Kalantari, A. A. Efros, and R. Ramamoorthi, "Light field video capture using a learning-based hybrid imaging system," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, 2017.

[127] D. Cho, S. Kim, and Y.-W. Tai, "Consistent matting for light field images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 90–104, 2014.