Unified Multi-Cue Depth Estimation from Light-Field Images: Correspondence, Defocus, Shading, and Specularity

Michael Tao
Ravi Ramamoorthi, Ed.
Jitendra Malik, Ed.
Alexei (Alyosha) Efros, Ed.

Electrical Engineering and Computer Sciences
University of California at Berkeley

July 21, 2015

Acknowledgement

Unified Multi-Cue Depth Estimation from Light-Field Images: Correspondence, Defocus, Shading, and Specularity

by

Michael Tao

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ravi Ramamoorthi, Chair
Professor Jitendra Malik
Associate Professor Alexei Efros
Associate Professor Sara McMains

Summer 2015

Unified Multi-Cue Depth Estimation from Light-Field Images: Correspondence, Defocus, Shading, and Specularity

Copyright 2015

by

Michael Tao

Abstract

Unified Multi-Cue Depth Estimation from Light-Field Images:
Correspondence, Defocus, Shading, and Specularity

by

Michael Tao

Doctor of Philosophy in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Ravi Ramamoorthi, Chair

Light-field cameras have recently become available to the consumer market. An array of micro-lenses captures enough information that one can refocus images after acquisition, as well as shift one's viewpoint within the sub-apertures of the main lens, effectively obtaining multiple views. Thus, depth cues from defocus, correspondence, specularity, and shading are available simultaneously in a single capture. Previously, defocus could be achieved only through multiple image exposures focused at different depths; correspondence and specularity cues needed multiple exposures at different viewpoints or multiple cameras; and shading required very well controlled scenes and low-noise data. Moreover, all four cues could not easily be obtained together.

In this thesis, we will present a novel framework that decodes the light-field images from a consumer Lytro camera and uses the decoded image to compute dense depth estimation by obtaining the four depth cues: defocus, correspondence, specularity, and shading. By using both defocus and correspondence cues, depth estimation is more robust with consumer-grade noisy data than previous works. Shading cues from light-field data enable us to better regularize depth and estimate shape. By using specularity, we formulate a new depth measure that is robust against specularity, making our depth measure suitable for glossy scenes. By combining the cues into a high quality depth map, the results are suitable for a variety of complex computer vision applications.

To My Family and Friends

For their continuous support and encouragement. It doesn't stop here.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisors, Ravi Ramamoorthi and Jitendra Malik, for their valuable guidance and support throughout my Ph.D. career. They have spent countless hours coming up with ideas and refining my projects. I especially want to thank Ravi for giving me the advice that extend much more than just my academic career.

I thank my co-authors and those who helped in all my publications: Sunil Hadap for helping me with initial ideas on light-fields depth estimation and shape from shading; Szymon Rusinkiewicz for advices on formulating shape from light-fields and quantitative evaluation of our depth estimation algorithms; Pratul Srinivasan for helping me with several projects for countless hours; Ting-Chun Wang for helping me with the implementation of the specularity paper; Sean Arietta for setting up the 3D printer and computers; Donald Dansereau for advices on how to decode the Lytro cameras; Jong-Chyi Su for his help on our specularity journal paper's theory; and Weilun Sun for helping with and gathering 3D data.

I would also like to thank Brian Barsky, Sylvain Paris, Aravind Krishnaswamy, Jeff Chien, and Pushmeet Kohli for building the foundations in my undergraduate and earlier Ph.D. career; Jiamin Bai, Jon Barron, Amy Lam, Dikpal Reddy, and Eno Toeppe for helping me build project ideas; and, of course, The Daily Californian for project inspirations.

Finally, Go Bears!

Chapter 1

Introduction

The dissertation introduces a new pipeline that uses light-field cameras to not only capture color information from the scene, but the 3D structure of the scene. Many researchers and practitioners have used conventional cameras to achieve such tasks. Conventional cameras take a snapshot of a scene. The snapshot includes the visible light spectrum that passes through the lens. Light is represented by directional rays that come from the scene and is recorded on the film sensor of the camera. The film of the cameras only records the summation of the rays, causing ambiguity in distinguishing among the rays. Light-field cameras contain extra lenses that help distinguish the rays entering the lens. The extra lenses, called micro-lenses, are placed between the main-lens and the sensor. The micro-lens disperses the directional rays entering the sensor, allowing us to retrieve direction and magnitude of the rays. By retrieving the rays' direction and magnitude, we are able to estimate depth, which we define as the distance between a point of the scene and the camera. The final output of our system is a depth map, which maps the depth of each point in the scene observed by the sensor.

In this dissertation, we explain how we exploit the data from light-field cameras to estimate depth. The goal of the dissertation is to demonstrate a point-and-shoot depth acquisition pipeline suitable for multiple scenarios, including real-world scenes using different camera parameters. We first describe how we decode the Lytro Illum camera image file from a Lytro proprietary image file into a standard format, suitable for our depth analysis (Chapter 2). To estimate depth, we then demonstrate how we can use the decoded light-field image to extract the following four cues from the data: defocus, correspondence, shading, and specularity. We use both defocus and correspondence as a baseline robust depth estimation for light-field cameras. We use shading to assist us in normal and shape estimation (explained in Chapter 3). Finally, to increase depth estimation robustness in scenes with glossy objects, we reduce the effects from specularity

by introducing a new depth metric that is robust against specular regions (explained in Chapter 4).

The larger implication of the dissertation is creating an easy-to-use point-and-shoot depth acquisition device, where a novice user can capture depth information from a scene. With this depth estimation, there are countless applications in both computer vision and computer graphics that enable the user to have more control of the captured data. The control includes changing depth-of-field, focus plane, and small perspective views. Prior to this work, many of these applications required specialized active depth acquisition devices or setups that are difficult for an average user.

1.1 Motivation: Depth Cues and Light-fields

Background: Depth Cues

We will first explain two different depth cues: monocular and stereo. Monocular depth cues are cues that one viewpoint can observe (a single camera setup), while stereo depth cues are cues that multiple different viewpoints can observe (two human eyes).

Monocular Cues. By using one camera from a single perspective, depth cues, such as object size [72], shading, and defocus blur [52], can be obtained from the scene. In biological sciences and psychology, many studies have shown monocular cues assist both human and animals understand depth from a scene [30]. However, using a single image is difficult to estimate depth [72]. The difficulties arise because most monocular cues require training or only provide information about relative depth [49]. Therefore, stereo or multiple images from different perspectives help with better depth estimation of the scene.

Stereo Cues. Stereo cues have been extensively studied to improve depth estimation [64, 82]. Popular stereo techniques include using two cameras to triangulate where objects are in the scene [16]. To compute the triangulation, many techniques such as computing displacement changes between images from different perspectives are effective [29, 48]. Because of the ability to triangulate, stereo cues compliment monocular cues in determining absolute depth.

Combining Monocular and Stereo Cues. As described by Sousa et al. [72], combining the two cues provides more accurate depth estimations. Combining the two cues has been the rudimentary basis of how humans perceive depth [86].
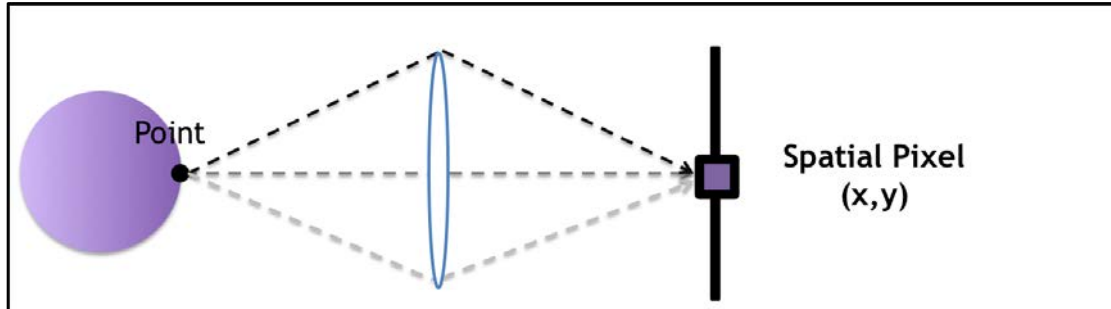
Figure 1.1: *Conventional Cameras and Recorded Rays. We have a conventional camera taking a photo of a purple sphere scene. If the main lens is focused on the point indicated in the scene, rays from the point enter into the main lens and converge at the sensor with coordinates (x, y). Because the sensor only records the summation of the magnitudes from the three rays, distinguishing among the three rays is ambiguous.*

Why Light-Field Cameras

Recently, light-field cameras have become readily available for consumers. Light-field cameras hold great promise for passive and general depth estimation and 3D reconstruction in computer vision. Light-field cameras enable us to distinguish rays coming from the main lens. Because of their ability to distinguish each ray's direction and magnitude, light-field cameras provide *both* monocular and stereo depth cues in an easy-to-capture method. In this work, we focus on two depth cues: defocus (monocular) and correspondence (stereo) that we can extract from the light-field camera. In addition, because of the extra ray information, we can extract both shading and specularity cues robustly.

Conventional Cameras

With conventional cameras, combining both monocular and stereo cues is difficult because, to obtain the necessary information, multiple captures or cameras are needed. In Fig. 1.1, we have a conventional camera that has a lens and a sensor. We consider a point in the scene and the light rays starting from that point. To simplify the visualization, we use three rays. All three rays enter the main lens and converge to a point on the sensor, which we call a pixel, with coordinates (x, y). The pixel just sums up the three rays.

Because of the summation, obtaining such depth cues is limited to monocular cues such as object size and defocus blur. Therefore, using a single conventional camera limits the ability to extract stereo depth cues. Using multiple exposures or cameras is needed to capture stereo information. However, for a novice user, capturing mul-
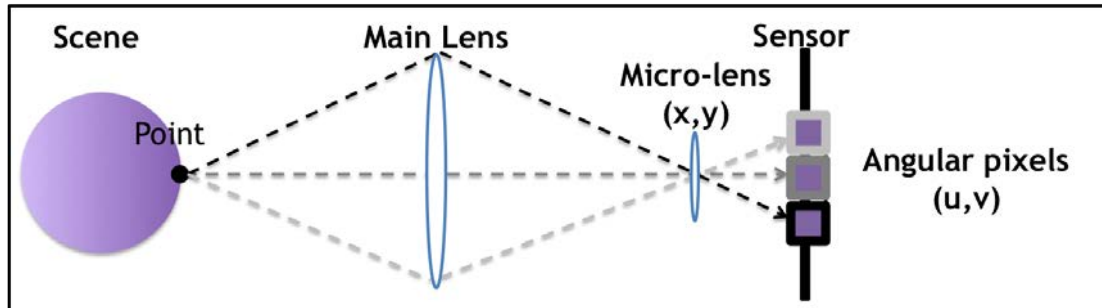
Figure 1.2: *Light-field cameras and Recorded Rays. Instead of the rays converging on the sensor, the rays continues their path from the micro-lens to the sensor. As a result, each sensor pixel records each of the three rays, thereby, preserving the direction and magnitude of the light rays.*

tiple exposures or using a multi-camera set up requires additional steps and is difficult. Therefore, a simpler one-shot point-and-shoot approach is needed.

Light-Field Advantages

Light-field cameras enable the simpler one-shot point-and-shoot approach. Light-field cameras contain additional micro-lenses between the main lens and the sensor. Instead of the rays converging at the sensor, the rays continue their path from the micro-lens to the sensor. The micro-lens acts as a diffuser, allowing, in this case, the three rays to disperse onto three different pixel coordinates of the sensor. As a result, a different sensor pixel records each of the three rays, thereby preserving the direction and magnitude of the light rays, as shown in Fig. 1.2.

Because we are able to obtain both direction and magnitude of each of the light rays, the light-field camera is able to extract stereo cues is one exposure within one camera. With the stereo cues, the light-field camera provides four important depth cues: multiple perspectives, refocusing, shading, and specularity.

Multiple Perspectives   By extracting the direction and magnitude of each of the light rays, each ray therefore represents a new viewpoint from the camera. In Figure 1.3, we can see that the viewpoint changes are recorded in three different parts of the sensor.

The new perspective viewpoints then start from sampled points on the main lens aperture and points towards the focus point in the scene. Because of the shift in perspectives, we can then use stereo depth cues to estimate the distance between the
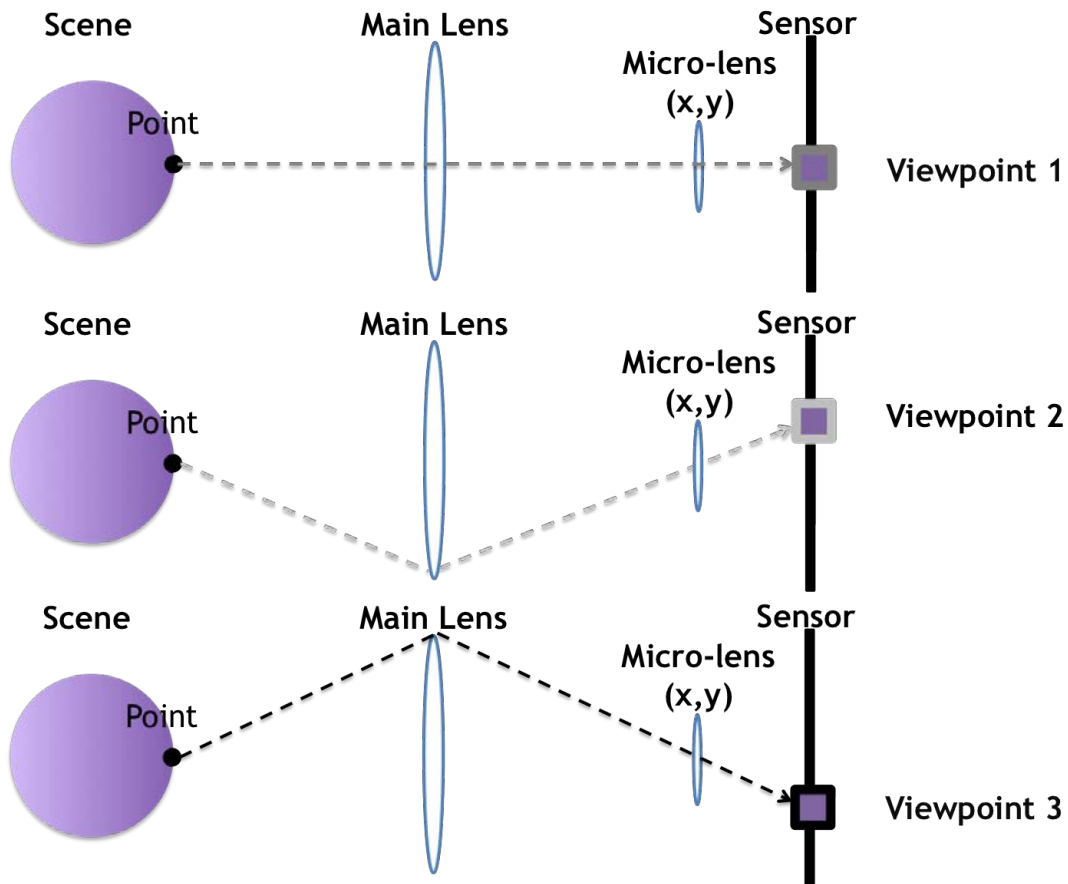
Figure 1.3: *Multiple Perspectives from a Light-Field Camera. Because each ray is recorded by one sensor pixel, by observing the different sensor pixels, we can change viewpoints as illustrated above. In this case, we can extract three different viewpoints.*
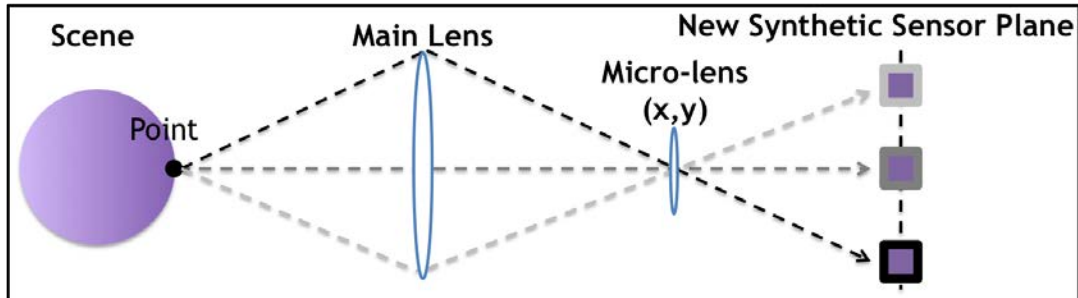
Figure 1.4: *Light-Field Cameras and Defocus. By rearranging the recorded pixels, we can synthetically change the focus plane even after the picture is taken. This enables us to use defocus cues, or observe how refocusing changes the recorded scene, for depth estimation.*

point on the scene and the camera. We explain how exactly we shift viewpoints with light-field cameras throughout the description of our algorithm in Chapter 3.

Defocus   Another depth cue that we are able to extract from the light-field camera is defocus. By being able to re-arrange the recorded pixels of the sensor, we are able to simulate a new synthetic sensor plane as shown in Fig. 1.4. The rearranging then allows us to create a new synthetic focus plane, enabling us to change the focus settings of the camera even after the picture is taken.

By being able to create a synthetic sensor plane, we are able to change the focus plane of the captured image. Pixel rearrangement is described in Chapter 3 of the thesis.

Shading   The third depth cue is shading. In a completely diffuse surface with no viewpoint angular dependence, because all three of the rays originate from the same point of the scene, the recorded values from the sensor should be the same. Therefore, the three recorded pixels should also register the same shading values as shown in Fig. 1.5.

Because of the redundancy of shading information, we are able to robustly estimate the shading values of each recorded scene point on the sensor. This is especially important because sensors may record incorrect values due to imperfections such as noise. Therefore, we can estimate the shading values from such consumer cameras.
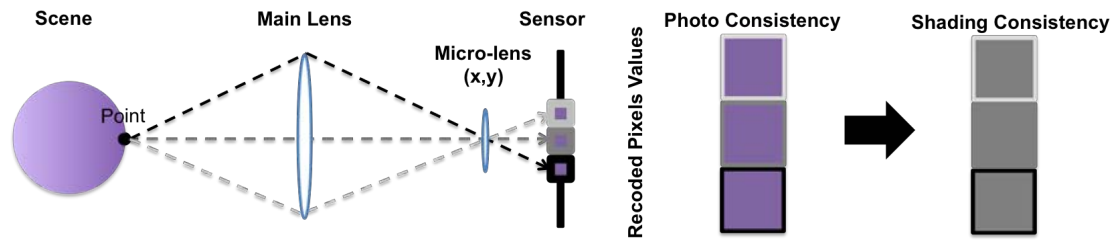
Figure 1.5: *Light-Field Cameras and Shading. Since all three rays come from the same point in the scene, all three rays should record the same pixel value, purple, which we call photo consistency. This also translates to shading consistency, since the point should have the same shading value. This redundancy of information becomes especially important for robustly estimating the shading values, which is described in Chapter 3.*

Specularity Finally, the fourth depth cue is specularity. Because we are able to extract multiple viewpoints that are directed to the same point of the scene, we can distinguish between diffuse and glossy regions. For diffuse regions, such as chalk or plastic surfaces, changing viewpoints do not affect the registered pixel values. However, for glossy regions, such as polished surfaces or mirrors, changing viewpoints affect the registered pixel values as shown in Fig. 1.6.

We exploit this property for glossy surfaces and formulate a new depth metric that is robust against these specular regions. Moreover, because we can analyze the change in registered color values from viewpoint changes, we are also able to estimate the light source color, which we later describe in Chapter 4.

Summary and Challenges

The light-field camera enables us to capture enough information to extract multiple perspectives and refocusing capabilities– all in one capture. However, capturing such data has its downsides. Extracting the image data from the Lytro camera is not trivial. The Lytro camera uses a proprietary image format that encodes and compresses the image data. In order to perform the depth analysis, we need to extract the image information from the camera by reverse engineering the image format. Because each sensor pixel captures one ray instead of a summation of rays, the sensor now effectively captures less spatial resolution (which is now determined by the number of micro-lenses). Moreover, calibration of where the micro-lenses are located becomes a challenge. Many times the micro-lenses across the image are not placed in a uniform fashion due to imperfections
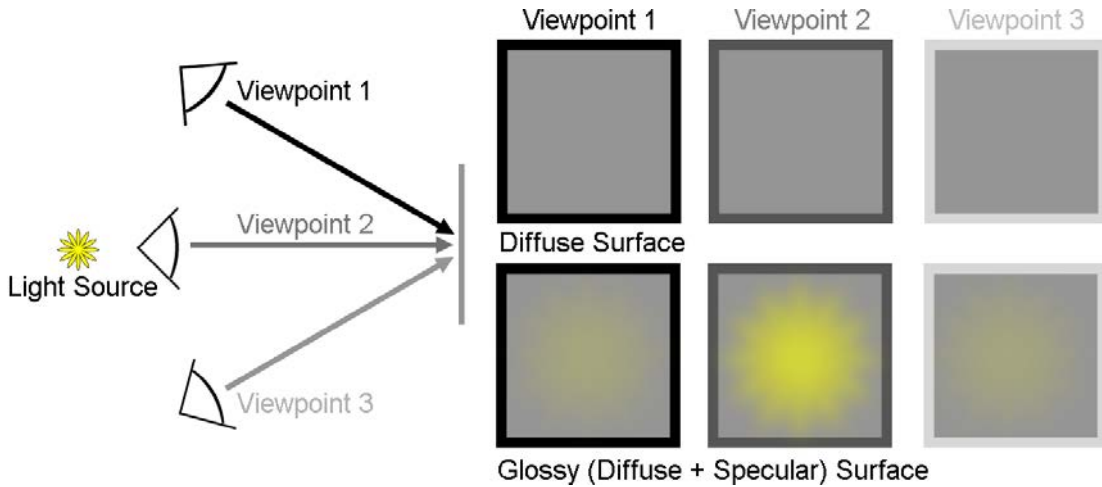
Figure 1.6: *Light-Field Cameras and Specularity. In a diffuse surface, changing view-points that are directed to a same point on the scene should not experience changes in the recorded pixel values. However, for a glossy surface, changing viewpoints will affect the registered pixel values. In Chapter 4, we discuss how we use this property to estimate the light source color and depth in such glossy surfaces.*

of manufacturing. Obtaining the center of each micro-lens is important as it is necessary for us to be able to accurately determine the rays' directions. Therefore, even with the factory calibration, image artifacts may result from refocusing, which we address in our depth acquisition algorithm through using both defocus and correspondence. The perspective change of the light-field camera is also limited by the aperture of the main-lens. Usually, small devices contain small aperture lenses; therefore, perspective changes are limited. Although we are able to change viewpoints, the change is limited to the aperture size of the lens, which is why we use both defocus and correspondence to achieve higher quality depth. Finally, noise becomes a significant issue due to the reduced amount of light absorbed by the sensor. Therefore, addressing robustness of our high quality depth is pertinent throughout the thesis.

1.2   Dissertation Overview

In this dissertation, we take into account both benefits and disadvantages of using a light-field camera to produce high quality depth estimation results. First, we show how we extract the image data from the proprietary image format encoded by Lytro. Then, we focus on obtaining robust and high quality depth estimation by using the four cues that we extract from the image data: defocus, correspondence, shading, and specularity.
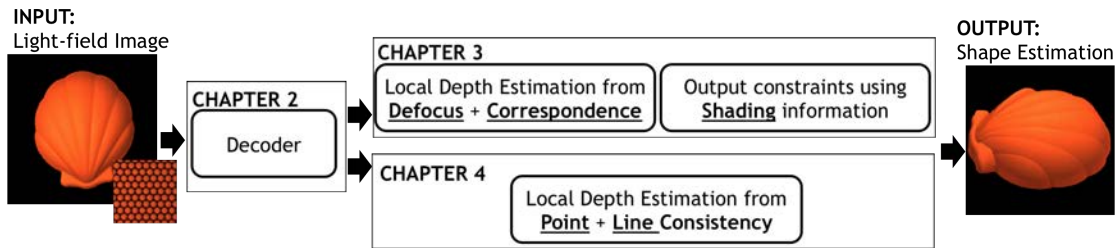
Figure 1.7: *Dissertation Roadmap. We first describe how we obtain the image data from the Lytro Illum camera (Chapter 2). With the image data, we can extract four cues: defocus, correspondence, shading, and specularity. Chapter 3 describes how we use defocus and correspondence for estimating depth for scenes with textures. We then incorporate the depth cues with shading to improve shape estimation. Chapter 4 describes how we reduce the effects of specular regions by introducing a new depth estimation measure, which we call line-consistency.*

We first use defocus and correspondence as our initial depth estimation and show how the two cues can provide us robust planar depth estimation. Then we show how shading estimation provides us better information about the normals and shape of the scene. Finally, we describe how we minimize specularity effects that are present in scenes with glossy surfaces.

The illustration of the dissertation roadmap is shown in Fig. 1.7. The input of the system is an image captured by a light-field camera. The output of the system is a depth estimation of the scene.

Dissertation Chapter Description

Chapter 2. Decoding the Lytro Illum Camera

With the Lytro Illum camera, Lytro introduced a new Lytro Image File type (.LFR). The file contains a non-standard encoding of the image data. Before we can process and estimate depth, we need to extract the image data from the Lytro cameras. In this chapter, we will discuss decoding the .LFR file. We show how we locate the image data and decode the data. We also discuss how we calibrate the color and the centers of the micro-lenses, which are needed for our depth estimation analysis. The chapter involves reverse engineering of the encoded proprietary file and calibrating both color and micro-lens positions.

Chapter 3. Shape Estimation from Shading, Defocus, and Correspondence Using Light-

Field Angular Coherence
Light-field cameras capture many nearby views simultaneously using a single image with a micro-lens array, thereby providing a wealth of cues for depth recovery: defocus, correspondence, and shading. In particular, apart from conventional image shading, one can refocus images after acquisition, and shift one's viewpoint within the sub-apertures of the main lens, effectively obtaining multiple views. We present a principled algorithm for dense depth estimation that combines defocus and correspondence metrics, showing how both can be expressed as simple operators in the epipolar image. We then extend our analysis to the additional cue of shading, using it to refine fine details in the shape. By exploiting an all-in-focus image, in which pixels are expected to exhibit **angular coherence**, we define an optimization framework that integrates photo consistency, depth consistency, and shading consistency. We show that combining all three sources of information: defocus, correspondence, and shading, outperforms state-of-the-art light-field depth estimation algorithms in multiple scenarios.

Chapter 4. Depth Estimation and Specular Removal for Glossy Surfaces Using Point and Line Consistency with Light-Field Cameras
We have a demonstrated practical algorithm for depth recovery from a passive single-shot capture. However, current light-field depth estimation methods are designed for Lambertian objects and fail or degrade for glossy or specular surfaces because photo-consistency depth measurement is a poor metric for such surfaces. In this chapter, we present a novel theory of the relationship between light-field data and reflectance from the dichromatic model. We present a physically-based and practical method to separate specular regions and estimate the light source color. As opposed to most previous works, our algorithm supports multiple lights of different colors. Our novel algorithm robustly removes the specular component for glossy objects. In addition, our approach enables depth estimation to support both specular and diffuse scenes. We show that our method outperforms current state-of-the-art specular removal and depth estimation algorithms in multiple real world scenarios using the consumer Lytro and Lytro Illum light field cameras.

Chapter 5. Conclusion
We then discuss the possible applications for our depth estimation and future work.

Chapter 2

Decoding the Lytro Illum Camera

With the Lytro Illum camera, Lytro introduced a new Lytro Image File type (.LFR). The file contains a non-standard encoding of the image data. Before we can process and estimate depth, we need to extract the image data from the Lytro cameras. In this chapter, we will discuss decoding the .LFR file. We show how we locate the image data and decode the data. We also discuss how we calibrate the color and the centers of the micro-lenses, which are needed for our depth estimation analysis. The chapter describes reverse engineering of the encoded proprietary file and calibrating both color and micro-lens positions.

2.1   Introduction

The first Lytro camera was introduced in 2012 [33]. The .LFR file format introduced with the Lytro camera is .LFP. The file format is simple and the data components all have fixed lengths [31]. Therefore, extracting the image information was trivial. However, with the updated version of the camera, both the Lytro Illum camera and an updated file format .LFR were introduced in 2014 [32]. The new file format introduces new encoding, color handling, and calibration.

The file format introduces a flexible framework that allows a variable amount of metadata and image data to be stored in one file. The flexibilities include the following: variable amount of metadata that can be stored, variable image data that can be stored, and encoded image data to reduce storage size. The .LFR file format is proprietary; therefore, obtaining the image data requires reverse engineering of its encoding. The desktop software included with the camera does not allow us to access the raw light-field image data, which is needed for our shape analysis.
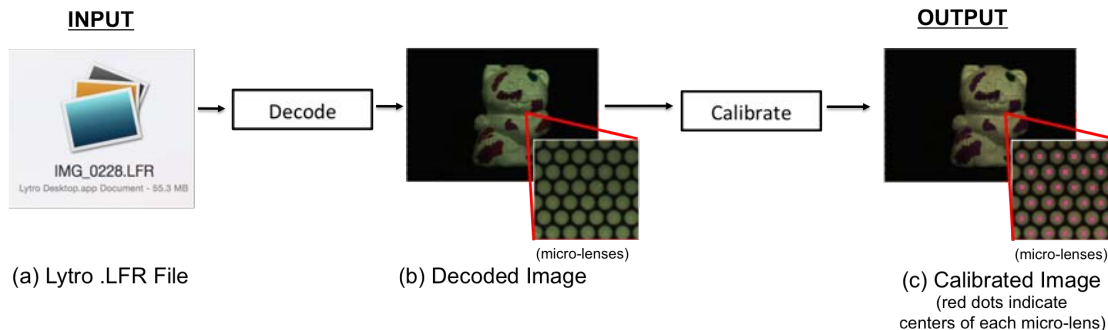
Figure 2.1: Decoding and Calibration Pipeline. *The input of our system is the .LFR file. Because the .LFR file format is proprietary, the image data is encoded. We show the reverse engineering performed to extract the image data from the file and convert the data into a standard image format. After decoding the image, we calibrate the image color and find the center of each micro-lens.*

Because of the file storage flexibility, decoding the .LFR file format is not as trivial as its predecessor. In this work, we show two steps to the decoding of the .LFR file: 1) find the image data block and 2) decode the image data.

The decoder has been posted publicly as one of the first open source Illum decoders[1]. As a disclaimer, this work is not affiliated with or endorsed by Lytro.

2.2 Decoding

As shown in Fig. 3.1, our input is the Lytro .LFR image file and the output is the decoded light-field image that is represented by 5368 pixels by 7728 pixels by 12-bit data. We will first describe how we find the image data. Then we explain how we decode the image data to extract the light-field image. Finally, we briefly describe how we interpret the pixel value information to convert the data into a standard RGB image format.

Finding the Image Data

In Fig. 2.2, we show the .LFR file structure. The blue areas indicate variable amount of metadata. In this section, we ignore most of the metadata, which contains all the information about camera settings, color calibration, camera serial numbers, and other

---

[1]`http://cseweb.ucsd.edu/~ravir/illum_full.zip`

.LFR File (Represented in Bytes)
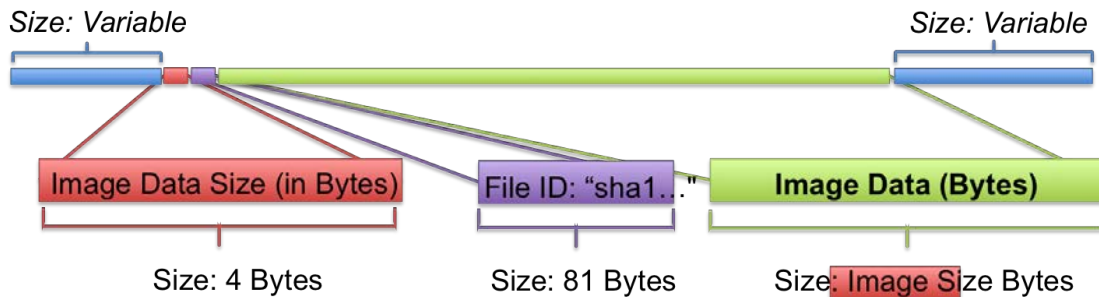
Figure 2.2: .LFR File Structure. *The .LFR file is read in Bytes. Because there is a variable amount of meta-data before and after the image data, we need to first locate the image data. To locate the image data, we first locate the file ID, which is designated by "sha1" in ASCII. The 4 bytes immediately before the file ID store the image data size, which is important for us to identify where the image data ends. Immediately after the file ID, the image data is stored, which we will use to decode the image of interest.*

sensor data. Our goal is to find the start and end of the image data, which will be used for decoding.

We open the .LFR file as a "char" (8 bit) file in text format. To find the image block, there are two important data blocks we need to locate: the image data size and the file ID. The image data size tells us the size in Bytes of the image data, which is needed to find the beginning and end data block of the image file size. The file ID is sandwiched between the two; because of its unique string properties, the file ID is easier and more consistent to locate. Our first step is to find the File ID by searching for the string "sha1." "sha1" is a hash identification number used in all the Lytro files. The file ID is exactly 81 Bytes which helps with identifying the starting location of the image data.

By identifying the location of the File ID, we extract the image data size, which is in Byte units. The 4 Bytes prior to the File ID contain the image data size. Given the size, we then are able to extract the image data, that immediately follows after the 81 Bytes of the file ID data block.
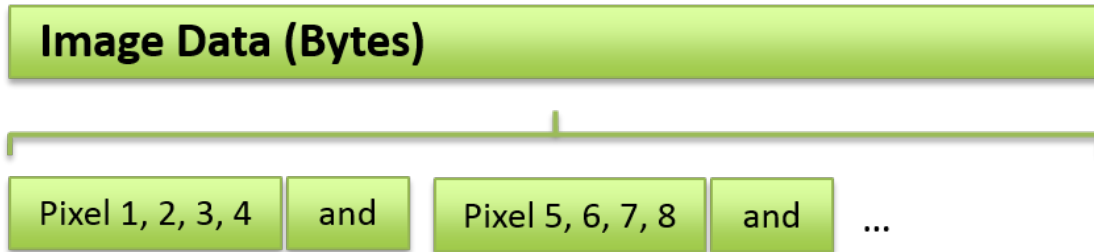
Figure 2.3: Image Data Structure. *The image data groups four consecutive pixels in five Bytes to reduce storage space. The first four Bytes represent the pixel values and the last Byte represents the logical decoder that is needed to extract the values of the pixels.*

Decoding the Image Data

Now that we have located the image data block, we need to decode the image data. The image data is encoded to reduce file size. The output of each pixel value is 12 bits. However, the encoded image data for four pixels is grouped in 5 Bytes (40 bits); in other words, each pixel value is encoded on average in 10 bits, reducing 2 bits of storage per pixel. To pack the pixels in 10 bits on average, part of the 5 Bytes encodes special logic bits. This significantly reduces data footprint especially for larger images.

The image data groups four pixels values together and a logical Byte in groups of five Bytes, as shown in Fig. 2.3. The first four Bytes encode the encoded pixel values in 8 bits. The fifth Byte is a logical Byte that is needed to decode the pixel values into a standard 12 bit values.

The outline algorithm is shown in Alg. 1 and illustrated in Fig. 2.4. The input of the system takes 5 Bytes of the image data block at a time. We do this for each of the 5 Byte groups within the image data block.

Algorithm 1 Decoding 5-Pixel Groups

1: **procedure** DECODE($B1, B2, B3, B4, B5$)
2: $B1, B2, B3, B4, B5 = \text{uint16}(B1, B2, B3, B4, B5)$
3: $P1 = (B1 << 2) + (((B5)\&00000011) >> 0)$
4: $P2 = (B2 << 2) + (((B5)\&00001100) >> 2)$
5: $P3 = (B3 << 2) + (((B5)\&00110000) >> 4)$
6: $P4 = (B4 << 2) + (((B5)\&11000000) >> 6)$
7: **end procedure**
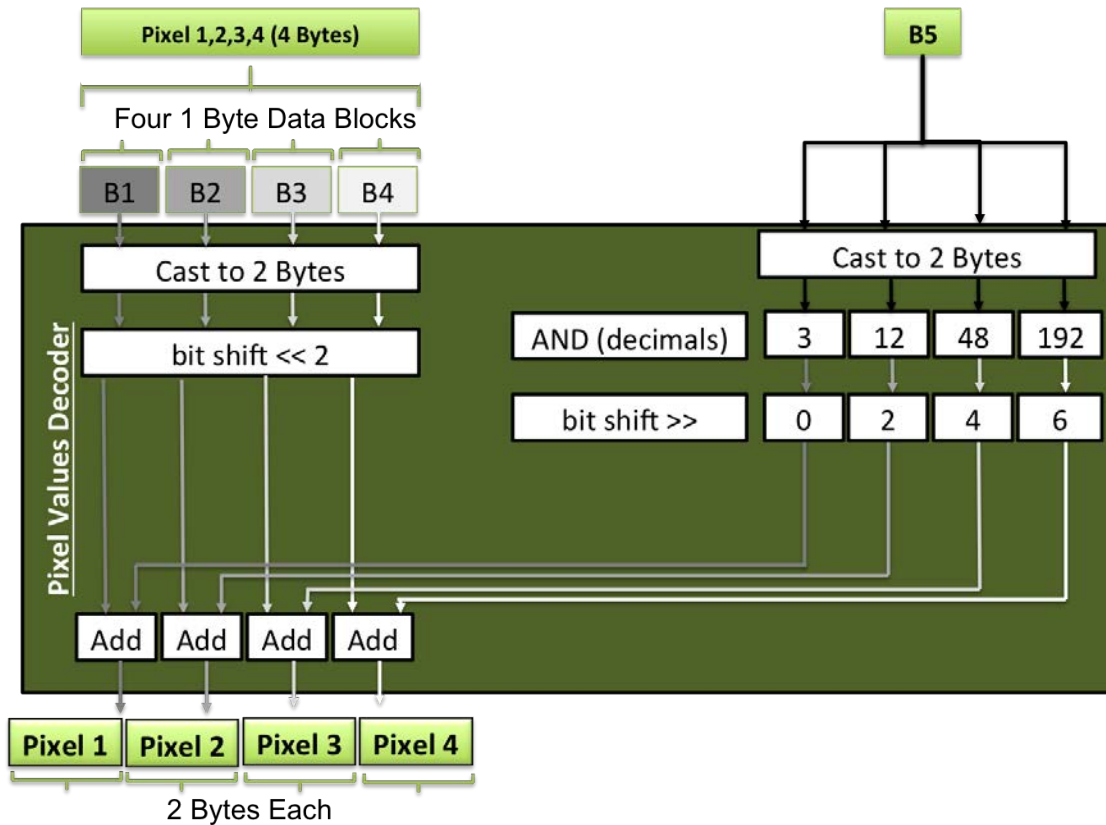
Figure 2.4: Image Data Decoder. *The image decoder consists of three steps: 1) bit shifting, 2) decode logical byte, and 3) adding. The input of the decoder is the five 1 Byte (8 bits) blocks and the output of each pixel value is 12 bits.*

The decoder consists of three steps: 1) bit shifting, 2) decode logical Byte, and 3) adding. For bit shifting, each pixel Byte block is shifted to the left by two. For the logical Byte, we logical AND with the different binary values for each pixel and shift to the right, which is different for each pixel value. Then, we add the shifted pixel value block and the new logical block together.

Interpreting the Decoded Image Data

After decoding all the 5 Byte blocks in the image data, we have the pixel value for each pixel on the sensor. To make an image file out of the file, we just remap the image data

into a (5368×7728) matrix for the Lytro Illum camera. We will now discuss two parts: extracting color and calibrating the center of micro-lenses.

Color   In Fig. 2.5, we show the two steps of converting grayscale to color. The first step involves demosaicing and the second step is color correction.

Since these pixel values are just grayscale values from the sensor overlaid with a Bayer color filter array, to obtain full color information, we need to use a demosaicing algorithm. A Bayer color filter array is used because, since a sensor pixel cannot distinguish between different color wavelengths, a Bayer color filter array places a filter on top of each sensor pixel, yielding either red, green, or blue pixels to pass through. The Lytro Illum uses a green-red-blue-green sensor pattern (top-left, top-right, bottom-left, and bottom-right, respectively). In order to obtain RGB information for each sensor pixel, interpolation among the colors is needed. This interpolation is called demosaic. After demosaicing, we can then save the RGB image file out as a standard image file such as PNG or TIFF.

For color correction, there are two pertinent items of color correction data: white balance gain and a color correction matrix. To get these values, by parsing through the metadata before and after the image data, we look for two headers in ASCII: "whiteBalanceGain" and "ccm." The white balance gain gives four values in float: a multiplier for each color filter, green-red-blue-green. After demosaicing, we then apply the color correction matrix provided by the metadata "ccm," which is represented by a 3x3 color matrix. For each pixel, we then compute the final RGB values as shown in Algorithm 2.

| Before Demosaicing | After Demosaicing | With Color Correction |
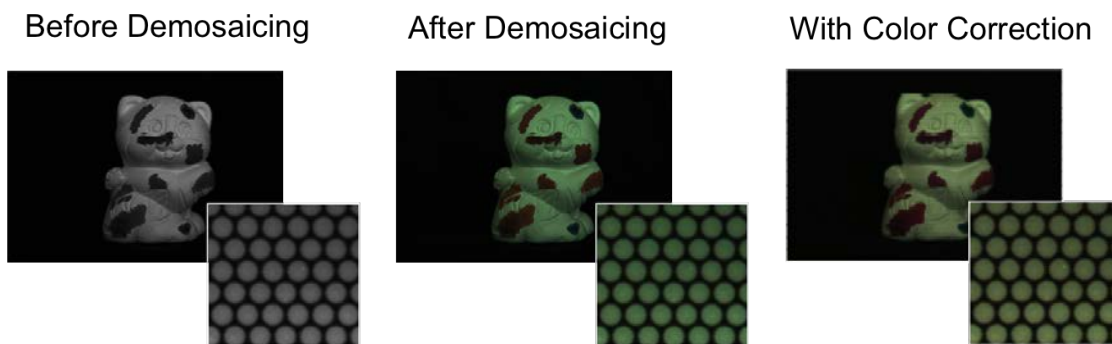
Figure 2.5: Grayscale to Color. *Before demosaicing, the image sensor only stores grayscale values. After demosaicing, we can obtain full color information. Color correction is then applied through white balancing and a color correction matrix computed by the camera.*

Algorithm 2 Color Correction Matrix

1: **procedure** COLORCORRECTIONMATRIX(R, G, B, CCM)
2:     $R = CCM(1,1) * R + CCM(2,1) * G + CCM(3,1) * B$
3:     $G = CCM(1,2) * R + CCM(2,2) * G + CCM(3,2) * B$
4:     $B = CCM(1,3) * R + CCM(2,3) * G + CCM(3,3) * B$
5: **end procedure**

Note: The color correction methods may differ among Lytro Illum cameras due to changes in sensor manufacturing process or supplier.

Micro-lens Calibration   To be able to register the direction of the rays for each micro-lens, we need to find the center. Finding the centers is what we define as micro-lens calibration. In the Lytro Illum, the micro-lens array is relatively uniform and consistent throughout among cameras. We can simply overlay an evenly spaced grid across the light-field image to locate the centers of the micro-lenses. Since the micro-lenses are hexagons, the micro-lenses are arranged with a different starting point for even and odd row-number pixels. In our user-defined parameters, the first odd-row pixel is $(12, 0.1412)$ and the first even-row pixel is $(4.4892, 12)$ and the spacing between the micro-lenses are 13.5108 for x and 11.8588 for y. In Fig. 2.6, we show that this simple approach exhibits very small errors even at the corners of the image.
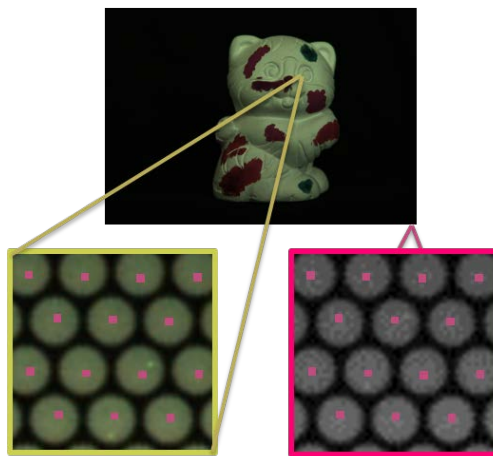
Figure 2.6: Micro-Lens Calibration. *We calibrate the micro-lenses by finding the centers of each micro-lens. To do so, we use an evenly-spaced grid. Because of Lytro Illum's uniformity in micro-lenses, the evenly-spaced grid exhibits low drifting errors even at the corners.*

Remapping and Notation After decoding the image and calibrating the centers of each micro-lens, we have a 2D image, which we denote as A and we have centers of each of the micro-lenses. Currently, the light-field image is represented by an image, $A(x, y)$, and for each micro-lens, we have a center of $c(x, y)$ where (x, y) denotes the indices of each micro-lens; c_x gives you the column component of A and c_y gives you the row component.

To simplify the notation, we remap $A(x, y)$ to $L(x, y, u, v)$ as follows,

$$L(x, y, u, v) = A(c_x(x, y) + u, c_y(x, y) + v). \tag{2.1}$$

In $L(x, y, u, v)$, throughout this dissertation, we denote (x, y) as the spatial coordinates of the light-field and (u, v) as the angular coordinates of each (x, y).

2.3 Applications and Limitations

By calibrating the center pixels of each micro-lens, we can then modify depth-of-field and viewpoint changes. In this section, we demonstrate how we perform the depth-of-field and viewpoint changes. We then discuss the limitations of our approach. In this section, we will denote L as the input light-field image, (x, y) are the spatial coordinates, and (u, v) are the angular coordinates. The central viewpoint (micro-lens center) is located at $(u, v) = (0, 0)$.

Modify Depth of Field To change the depth-of-field of the image, we just integrate all the angular pixels for each (x, y) coordinate as follows,

$$I(x, y) = \sum_{u', v'} M(u', v')L(x, y, u', v'). \tag{2.2}$$

M stores the aperture mask values, ranging from 0 to 1. We then normalize by sum of M values. In Fig. 2.7, we first show that we first only integrate the central viewpoint pixel and then we decrease the depth-of-field by integrating 7×7 angular pixels with $M(u', v') = 1$, centered around the center viewpoint.

Figure 2.7: Refocusing. *On the left, we only use the central viewpoint to generate the output image. By integrating a small 7×7 angular pixel neighborhood, we show that we decrease the depth-of-field significantly.*

Because we can change the depth-of-field of the light-field image, we are able to obtain refocusing cues, which are described in the next chapter. In our implementation, the maximum number of angular pixels we integrate is 7×7. If we increase this number, we start seeing micro-lens artifacts where some pixels become dimmer or the image exhibits patterned artifacts. This is because micro-lenses themselves have vignetting and lens imperfections.

Viewpoint Changes To change viewpoints, we use the following equation to choose a viewpoint,

$$I(x, y) = L(x, y, u, v). \tag{2.3}$$

When $(u, v) = (0, 0)$, we have the central pinhole image. In Fig. 2.8, we show that we can extract different viewpoints from the light-field image. Because we can change viewpoints, we are able to obtain stereo cues, which we will discuss in the next chapter.

Viewpoint 1	Viewpoint 2	Viewpoint Change Visualization
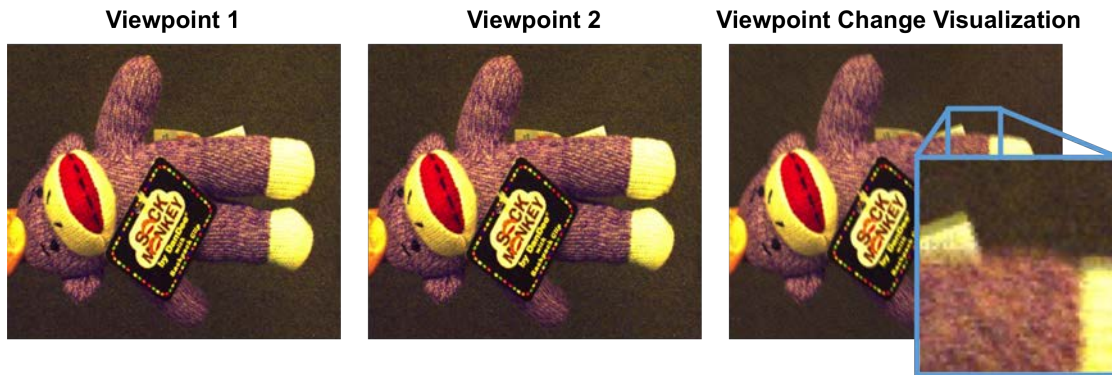
Figure 2.8: Viewpoint Changes. *By choosing a certain angular pixel (u, v), we are able to change viewpoints. On the left, we have a left-bottom corner viewpoint and in the middle, we have a right-top corner viewpoint. On the right, we overlay the two images to visualize the viewpoint changes.*

Limitations   Although we are able to extract the images from the camera, the Lytro software does more complex color correction, denoising, and super resolution for better image quality. As noted before, the color correction methods may differ among Lytro Illum cameras due to changes in sensor manufacturing process or supplier. The multipliers or methods of multiplying the colors to obtain the final result may vary depending on the camera used. In Fig. 2.9, we show the slight color, noise, and detail differences between our output image and Lytro's image.
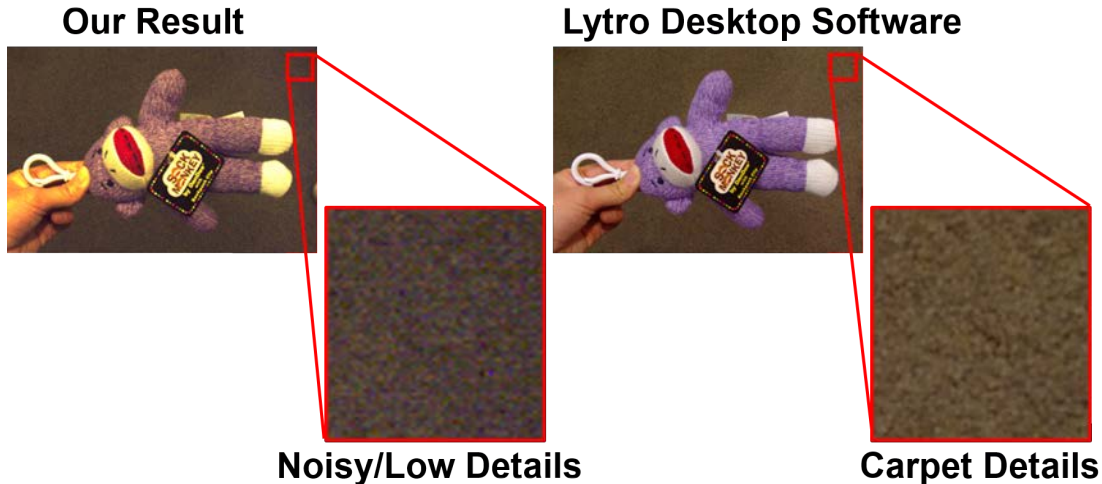
Figure 2.9: Decoder Limitations. *Because Lytro uses complex color correction, denoising, and super resolution techniques, we can see that in this image of the monkey, the carpet exhibits higher noise, lower details, and slight color shift compared to the Lytro's final output image. The exact color is not important for our depth estimation.*

2.4   Conclusion

In this chapter, we have discussed how to decode the .LFR file. We successfully converted the .LFR information into a standard image format that is appropriate for image depth analysis. We also show how we can change depth-of-field and viewpoints with the calibrated light-field images. The decoder has been posted publicly as one of the first open source Illum decoders[2].

---

[2]http://cseweb.ucsd.edu/~ravir/illum_full.zip

Chapter 3

Shape Estimation from Shading, Defocus, and Correspondence Using Angular Coherence

Light-field cameras capture many nearby views simultaneously using a single image with a micro-lens array, thereby providing a wealth of cues for depth recovery: defocus, correspondence, and shading. In particular, apart from conventional image shading, one can refocus images after acquisition, and shift one's viewpoint within the sub-apertures of the main lens, effectively obtaining multiple views. We present a principled algorithm for dense depth estimation that combines defocus and correspondence metrics, showing how both can be expressed as simple operators in the epipolar image. We then extend our analysis to the additional cue of shading, using it to refine fine details in the shape. By exploiting an all-in-focus image, in which pixels are expected to exhibit **angular coherence**, we define an optimization framework that integrates photo consistency, depth consistency, and shading consistency. We show that combining all three sources of information: defocus, correspondence, and shading, outperforms state-of-the-art light-field depth estimation algorithms in multiple scenarios.

3.1 Introduction

Light-fields can be used to refocus images [55]. Light-field cameras hold great promise for passive and general depth estimation and 3D reconstruction in computer vision. As noted by Adelson and Wang [1], a single exposure provides multiple viewpoints (sub-apertures on the lens). However, a light-field contains more information about depth; since we can refocus, change our viewpoint locally, *and* capture scene color information,

defocus, *correspondence*, and *shading* cues are present in a single exposure. Our main contribution is integrating all three cues as shown in Fig. 3.1.

We make the common assumption of Lambertian surfaces under general (distant) direct lighting. We differ from previous works because we exploit the full angular data captured by the light-field to utilize defocus, correspondence, and shading cues. Our algorithm is able to use images captured with the Lytro cameras. We compare our results both qualitatively and quantitatively against the Lytro Illum software and other state of the art methods (Figs. 3.10, 3.11, 3.12, and 3.14), demonstrating that our results give accurate representations of the shapes captured.

We describe a first approach to extract defocus and correspondence cues using contrast detection from the light-field data as shown in Fig. 3.2. We exploit the epipolar image (EPI) extracted from the light-field data, which we will describe later in this Chapter [10, 15]. The illustrations in the chapter use a 2D slice of the EPI labeled as (x, u), where x is the spatial dimension (image scan-line) and u is the angular dimension (location on the lens aperture); in practice, we use the full 4D light-field EPI. We shear to perform refocusing as proposed by Ng et al. [55]. For each shear value, we compute the *defocus cue response* by considering the spatial x (*horizontal*) variance, after integrating over the angular u (vertical) dimension. The defocus response is computed through the Laplacian operator, where high response means the point is in focus. The defocus measure is equivalent to finding the highest contrast given a refocus setting. In contrast, we compute the *correspondence cue response* by considering the angular u (*vertical*) variance. The correspondence measure is equivalent to finding the lowest variance across the angular pixels, achieving photo-consistency. Using contrast techniques for defocus and correspondence cue measurements is suitable in scenes with high textures and edges (Fig. 3.4). However, in scenes with low textures that rely on shading estimation, using such contrast techniques is more prone to instabilities in both depth and confidence measurements due to calibration problems, micro-lens vignetting, and high frequencies introduced from the shearing techniques (described in Chapter 3.3 and Fig. 3.7)

We overcome the shortcomings by improving our cue measures. Specifically, we use *angular coherence* to significantly improve robustness. When refocused to the correct depth, the angular pixels corresponding to a single spatial pixel represent viewpoints that converge on one point on the scene, exhibiting angular coherence. Angular coherence means the captured data would have **photo consistency**, **depth consistency**, and **shading consistency**, shown in Fig. 3.5. We extend the consistency observations from Seitz and Dyer [69] by finding the relationship between refocusing and achieving angular coherence. The extracted central pinhole image from the light-field
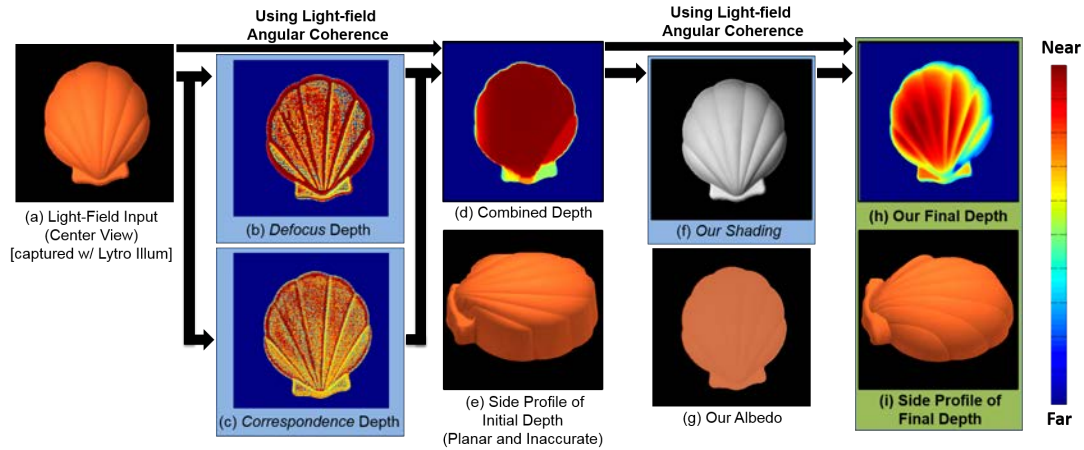
Figure 3.1: Light-field Depth Estimation Using Shading, Defocus, and Correspondence Cues. *In this work, we present a novel algorithm that estimates shading to improve depth recovery using light-field angular coherence. Here we have an input of a real scene with a shell surface and a camera tilted slightly toward the right of the image (a). We obtain improved defocus (b) and correspondence (c) depth cues for depth estimation (d,e). However, because local depth estimation is only accurate at edges or textured regions, depth estimation of the shell appears regularized and planar. We use the depth estimation to estimate shading, which is S, as shown in (f), the component in $I = AS$, where I is the observed image and A is the albedo (g). With the depth and shading estimations, we can refine our depth to better represent the surface of the shell (h,i). Throughout this chapter, we use the scale on the right to represent depth.*

data helps us enforce the three properties of angular coherence.

To utilize the shading cue, we first estimate the shading component of the image by extending a standard Retinex image decomposition framework introduced by Zhao et al. [94] enforces albedo and shading constraints. By using the full light-field 4D EPI and angular coherence, our method is robust against noisy and imperfect data (Fig. 3.8). The robustness allows us to accurately estimate lighting (Chapter 3.6) and estimate normals (Chapter 3.6). The angular coherence and combination of defocus, correspondence, and shading cues provide robust constraints to estimate the shading

normal constraints, previously not possible with low-density depth estimation.

In this chapter, our main contributions are

1. Analysis of defocus and correspondence.
We extract defocus and correspondence from a light-field image and show why using both cues is important.
2. Depth cues with angular coherence.
We show the relationship between refocusing a light-field image and angular coherence to formulate improved defocus and correspondence measures and shading estimation constraints.
3. Shading estimation constraints.
We formulate a new shading constraint, that uses angular coherence and a confidence map to exploit light-field data.
4. Depth refinement with the three cues.
We design a novel framework that uses shading, defocus, and correspondence cues to refine shape estimation.
5. Quantitative and Qualitative Dataset.
We quantitatively and qualitatively assess our algorithm with both synthetic and real-world images (Figs. 3.10, 3.11, 3.12, and 3.14).

3.2 Previous Work

Shape from Defocus and Correspondence

Depth from Defocus. Depth from defocus has been achieved either through using multiple image exposures or a complicated apparatus to capture the data in one exposure [68, 73, 85]. Defocus measures the optimal contrast within a patch, where occlusions may easily affect the outcome of the measure, but the patch-based variance measurements improve stability over these occlusion regions. However, out-of-focus regions, such as certain high frequency regions and bright lights, may yield higher contrast. The size of the analyzed patch determines the largest sensible defocus size. In many images, the defocus blur can exceed the patch size, causing ambiguities in defocus measurements. Our work not only can detect occlusion boundaries, we can provide dense stereo.

Depth from Correspondences. Extensive work has been done in estimating depth using stereo correspondence, as the cue alleviates some of the limitations of defocus [53, 67]. Large stereo displacements cause correspondence errors because of limited patch search space. Matching ambiguity also occurs at repeating patterns and

noisy regions. Occlusions can cause non-existent correspondence. Optical flow can also
be used for stereo to alleviate occlusion problems since the search space is both hor-
izontal and vertical [29, 48], but the larger search space dimension may lead to more
matching ambiguities and less accurate results. Multi-view stereo [45, 57] also alle-
viates the occlusion issues, but requires large baselines and multiple views to produce
good results.

Combining Defocus and Correspondence. Combining both depth from de-
focus and correspondence has been shown to provide benefits of both image search
reduction, yielding faster computation, and more accurate results [41, 74]. However,
complicated algorithms and camera modifications or multiple image exposures are re-
quired. In our work, using light-field data allows us to reduce the image acquisition
requirements. Vaish et al. [81] also propose using both stereo and defocus to com-
pute a disparity map designed to reconstruct occluders, specifically for camera arrays.
This chapter shows how we can exploit light-field data to not only estimate occlusion
boundaries but also estimate depth by exploiting the two cues in a simple and principled
algorithm.

Shape from Shading and Photometric Stereo

Shape from shading has been well studied with multiple techniques. Extracting geom-
etry from a single capture [28, 93] was shown to be heavily under constrained. Many
works assumed known light source environments to reduce the under constrained prob-
lem [19, 20, 26, 93]; some use partial differential equations, which require near ideal
cases with ideal capture, geometry, and lighting [11, 42, 94]. In general, these ap-
proaches are especially prone to noise and require very controlled settings. Recently,
Johnson and Adelson [37] described a framework to estimate shape under natural illu-
mination. However, the work requires a known reflectance map, which is hard to obtain.
In our work, we focus on both general scenes and unknown lighting, without requiring
geometry or lighting priors. To relax lighting constraints, assumptions about the geome-
try can be made such as faces [9, 75] or other data-driven techniques [5]. The method by
Barron and Malik [4, 6] works for real-world scenes and recovers shape, illumination,
reflectance, and shading from an image. However, many constraints, such as shape pri-
ors, are needed for both geometry and illumination. In our framework, we do not need
any priors and have fewer constraints.

A second set of works focuses on using photometric stereo [8, 17, 20, 26,
88, 87]. These works are not passive and require the use of multiple lights and captures.
In contrast, shape from shading and our technique just require a single capture.

Shape from Depth Cameras and Sensors

Work has been done using Kinect data [22]. Barron and Malik [5] introduce SIRFS that reconstructs depth, shading, and normals. However, the approach requires multiple shape and illumination priors. Moreover, the user is required to assume the number of light sources and objects in the scene. Chen and Koltun [13] introduce a more general approach to perform intrinsic image decomposition. However, the method does not optimize depth and, given sparse input depth with poor normal estimations at smooth surfaces, their shading estimation is poor and unsuitable for refining depth. Other works [54, 89] introduce an efficient method to optimize depth using shading information. The limitations of these approaches are that they require very dense and accurate depth estimation, achieved by active depth cameras. Even in non-textured surfaces, these active systems provide meaningful depth estimations. With passive light-field depth estimation, the local depth output has no or low-confidence data in these regions.

Shape from Modified Cameras

To achieve high quality depth and reduce algorithmic complexity, modifying conventional camera systems such as adding a mask to the aperture has been effective [43, 46]. The methods require a single or multiple masks to achieve depth estimation. The general limitation of these methods is that they require modification of the lens system of the camera, and masks reduce incoming light to the sensor.

Shape from Light-Fields and Multi-View Stereo

Hasinoff and Kutulakos [25] explain how focus and aperture provide shape cues and Van Doorn et al. [18] explain how light-fields provide useful shading information. To estimate these depth cues from light-field images, Perwass and Wietzke [61] propose correspondence techniques, while others [1, 44] have proposed using contrast measurements. Kim et al. and Wanner et al. [39, 83] propose using global label consistency and slope analysis to estimate depth. Their local estimation of depth uses only a 2D EPI to compute local depth estimates, while ours uses the full 4D EPI. Because the confidence and depth measure rely on ratios of tensor structure components, their result is vulnerable to noise and fails at very dark and bright image features.

Since light-fields and multi-view stereo are passive systems, these algorithms struggle with the accuracy of depth in low-textured regions [39, 65, 80, 79, 83] because they rely on local contrast, requiring texture and edges. With traditional regularizers [35] and light-field regularizers, such as one proposed by Wanner et al. [23], depth labeling is planar in these low-textured regions. In this chapter, we show how the angu-

lar coherence of light-field data can produce better 1) depth estimation and confidence levels, and 2) regularization.
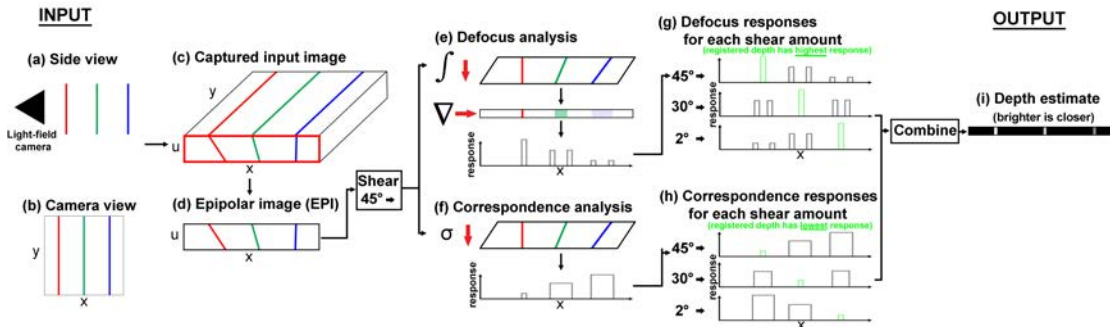
Figure 3.2: *Defocus and Correspondence Framework. This setup shows three different poles at different depths with a side view of (a) and camera view of (b). The light-field camera captures an image (c) with its epipolar image (EPI). By processing each row's EPI (d), we shear the EPI to perform refocusing. Our contribution lies in computing both defocus analysis (e), which integrates along angle u (vertically) and computes the spatial x (horizontal) gradient, and correspondence (f), which computes the angular u (vertical) variance. The response to each shear value is shown in (g) and (h). By combining the two cues through regularization, the algorithm produces high quality depth estimation (i). In Sections 3.4 and 3.5, we refine the defocus and correspondence measure and incorporate shading information to our regularization to produce better shape and normal estimation results by using angular coherence. With angular coherence, our defocus and correspondence measures are more robust in scenes with less texture and fewer edges.*

3.3 Defocus and Correspondence

By using both defocus and correspondence depth cues for local depth estimation, the algorithm benefits from the advantages of each cue, shown in Fig. 3.3. Defocus cues are better with occlusions, repeating patterns, and noise; but correspondence is more robust in bright/darker features of the image and has more defined depth edges.

We first present a simple contrast-based approach to compute the response of both cues. The contrast-based approach gives us a better insight on how to extract the cues. However, because of the limitations of the contrast-based methods in scenes with low texture and edges, in Chapter 3.4, we refine the measures to mitigate the limitations. Throughout this chapter, we refer to the contrast-based depth measure as the contrast based method in our comparison figures.

	Implementation	Occlusions	Repeating Patterns	Bright/Dark Features	Noise
Defocus	+ no calibration needed - aperture-size dependent - patch size dependent	+ easily affected - more stable	+ contrast detection distinguishes	- contrast detection ambiguous	+ 2D blur kernel provides better support with noise
Correspondence	+ not dependent on DOF - noise from using pinhole - correspondence problem	+ less affected - unstable if affected	- correspondence ambiguity	+ correspondence not affected as much	- matching prone to noise - pinhole image noise

Figure 3.3: Defocus and Correspondence Strengths and Weaknesses. *Each cue has its benefits and limitations. Most previous work use one cue or another, as it is hard to acquire and combine both in the same framework. In this chapter, we exploit the strengths of both cues. Additionally, we provide further refinement, using the shading cue.*

For easier conceptual understanding, we use the 2D EPI in this section, considering a scan-line in the image, and angular variation u, i.e. an (x-u) EPI where x represents the spatial domain and u represents the angular domain as shown in Fig. 3.2. This approach provides insight on how to utilize both cues. Ng et al. [55] explain how shearing the EPI can achieve refocusing. For a 2D EPI, we remap the EPI input as follows,

$$L_\alpha(x, u) = L(x + u(1 - \frac{1}{\alpha}), u), \tag{3.1}$$

where L denotes the input EPI and L_α denotes the EPI sheared by a value of α. We extend the refocusing to 4D in Eq. 3.7.

Defocus

Light-field cameras capture enough angular resolution to perform refocusing, allowing us to exploit the defocus cue for depth estimation. We find the optimal α with the highest contrast at each pixel as shown in Fig. 3.2e. The first step is to take the sheared EPI and integrate across the angular u dimension (vertical columns),

$$\bar{L}_\alpha(x) = \frac{1}{N_u} \sum_{u'} L_\alpha(x, u') \tag{3.2}$$

where N_u denotes the number of angular pixels (u). $\bar{L}_\alpha(x)$ is simply the refocused image for the shear value alpha. Finally, we compute the defocus response by using a measure:

$$D_\alpha(x) = \frac{1}{|W_D|} \sum_{x' \in W_D} |\Delta_x \bar{L}_\alpha(x')| \tag{3.3}$$

where W_D is the window size around the current pixel (to improve robustness) and Δ_x is the horizontal (spatial) Laplacian operator, using the full patch. For each pixel in the image, we now have a measured defocus contrast response for each α. (We improve upon this measure in Eq. 3.16, where the actual algorithm compares the contrast to the central pinhole image instead. The improved measure gives us better depth and confidence measures in low textured regions.)

Correspondence

Light-field cameras capture enough angular information to render multiple pinhole images from different perspectives in one exposure. Consider an EPI as shown in Fig. 3.2d. For a given shear α (Fig. 3.2f), we consider the angular (vertical) variance for a given spatial pixel.

$$\sigma_\alpha(x)^2 = \frac{1}{N_u} \sum_{u'} (L_\alpha(x, u') - \bar{L}_\alpha(x))^2 \tag{3.4}$$

For each pixel in x, instead of just computing the pixel variance, we need to compute the patch difference. We average the variances in a small patch for greater robustness,

$$\mathbf{C}_\alpha(x) = \frac{1}{|W_C|} \sum_{x' \in W_C} \sigma_\alpha(x') \tag{3.5}$$

where W_C is the window size around the current pixel to improve robustness. For each pixel in the image, we now have a measured correspondence response for each α. (We improve upon the correspondence measure in Eq. 3.17, where the actual algorithm compares the central pinhole image color value instead of the mean.)

Depth and Confidence Estimation

We seek to maximize spatial (horizontal) contrast for defocus and minimize angular (vertical) variances for correspondence across shears. We find the α value that maximizes the defocus measure and the α value that minimizes the correspondence measure.

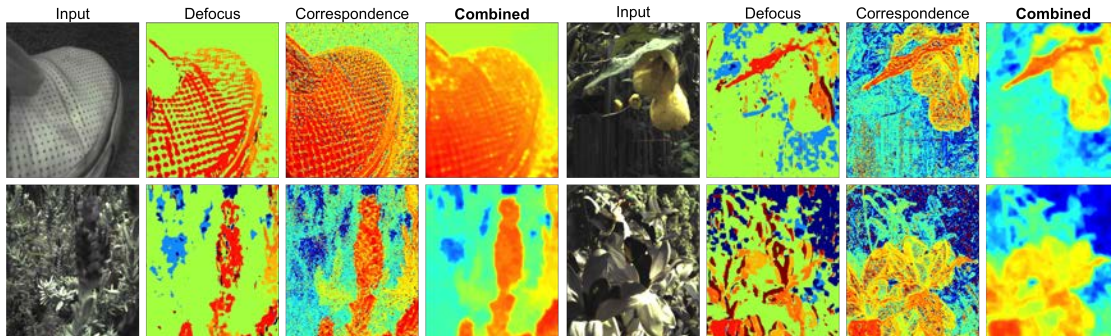| Input | Defocus | Correspondence | **Combined** | Input | Defocus | Correspondence | **Combined** |

Figure 3.4: Contrast-Based Defocus and Correspondence Results. *Defocus consistently shows better results at noisy regions and repeating patterns, while correspondence provides sharper results. By combining both cues, our method provides more consistent results in real world examples. The two low light images on the top show how our algorithm is able to estimate depth even at high ISO settings. The flowers (bottom left and right) show how we recover complicated shapes and scenes. By combining both cues, our algorithm still produces reasonable results. However, we can see that the contrast-based defocus and correspondence measures perform poorly in scenes where textures and edges are absent (Figs. 3.10, 3.11, 3.12, and 3.14). Therefore, we develop more robust cue measurements with angular coherence in Chapter 3.5.*

$$\alpha_D^\star(x) = \operatorname*{argmax}_{\alpha} \ D_\alpha(x)$$
$$\alpha_C^\star(x) = \operatorname*{argmin}_{\alpha} \ C_\alpha(x)$$

(3.6)

Defocus and correspondence cues might not agree on the optimal shear; we address this using our confidence measure and global step. To measure the confidence of $\alpha_D^\star(x)$ and $\alpha_C^\star(x)$, we found Attainable Maximum Likelihood (AML), explained in Hu and Mordohai [34], to be the most effective.

Regularization and Results

To combine the two responses and propagate the local depth estimation, we used the same optimization scheme, which is later described in Chapter 3.5. In Fig. 3.4, we show four depth estimation results using the contrast based depth cue measurement. We captured the images in three different scenarios (indoors and outdoors, low and high ISO, and different focal lengths). Throughout the examples, defocus is

less affected by noise and repeating patterns compared to correspondence while correspondence provides more edge information compared to defocus. Our combined results perform consistently and are robust in high texture situations.

Discussion

By using both defocus and correspondence, we are able to improve robustness of the system as shown in Fig. 3.4. However, using these measures is not ideal for scenes where the object is mainly textureless. Some of the factors include:

Shearing to refocus may introduce high frequencies. This can be due to miscalibration of micro-lenses, vignetting (lens imperfect where edges of the image are darker than the center), and other lens imperfections. In Fig. 3.7, we can see this effect on the top with the dinosaur example.

Noise and spots create low-confidence measures. This is especially prominent in smooth regions, which is not ideal for our depth results. In Fig. 3.7, we can see that angular variance measures fail.

Using these measures without shading constraints are not suitable for measuring normals as they introduce errors in smooth regions, as seen in Figs. 3.11 and 3.12. The depth and confidence of the contrast-based measures result in inconsistent regularization. Therefore, we use angular coherence to improve robustness in such scenes.

3.4   4D Angular Coherence and Refocusing

Angular coherence plays a large role in our algorithm to establish formulations for both the improved defocus-correspondence depth estimation and shading constraints. Our goal is to solve for 1) depth map, α^*, and 2) shading in $P = AS$, where P is the central pinhole image of the light-field input L, A is the albedo, and S is shading. In order to address the limitations of light-field cameras, we exploit the angular resolution of the data.

Here, we explain why a light-field camera's central pinhole image provides us with an important cue to obtain angular coherence. To shear the full 4D light-field image, the EPI remapping is as follows,

$$L_\alpha(x, y, u, v) = L(x^f(\alpha), y^f(\alpha), u, v)$$
$$x^f(\alpha) = x + u(1 - \frac{1}{\alpha})$$
$$y^f(\alpha) = y + v(1 - \frac{1}{\alpha})$$
(3.7)

where L is the input light-field image, L_α is the refocused image, (x, y) are the spatial coordinates, and (u, v) are the angular coordinates. The central viewpoint is located at $(u, v) = (0, 0)$.

Given the depth $\alpha^*(x, y)$ for each spatial pixel (x, y), we calculate L_{α^*} by refocusing each spatial pixel to its respective depth. All angular rays converge to the same point on the scene when refocused at α^*, as shown in Fig. 3.5. We can write this observation as
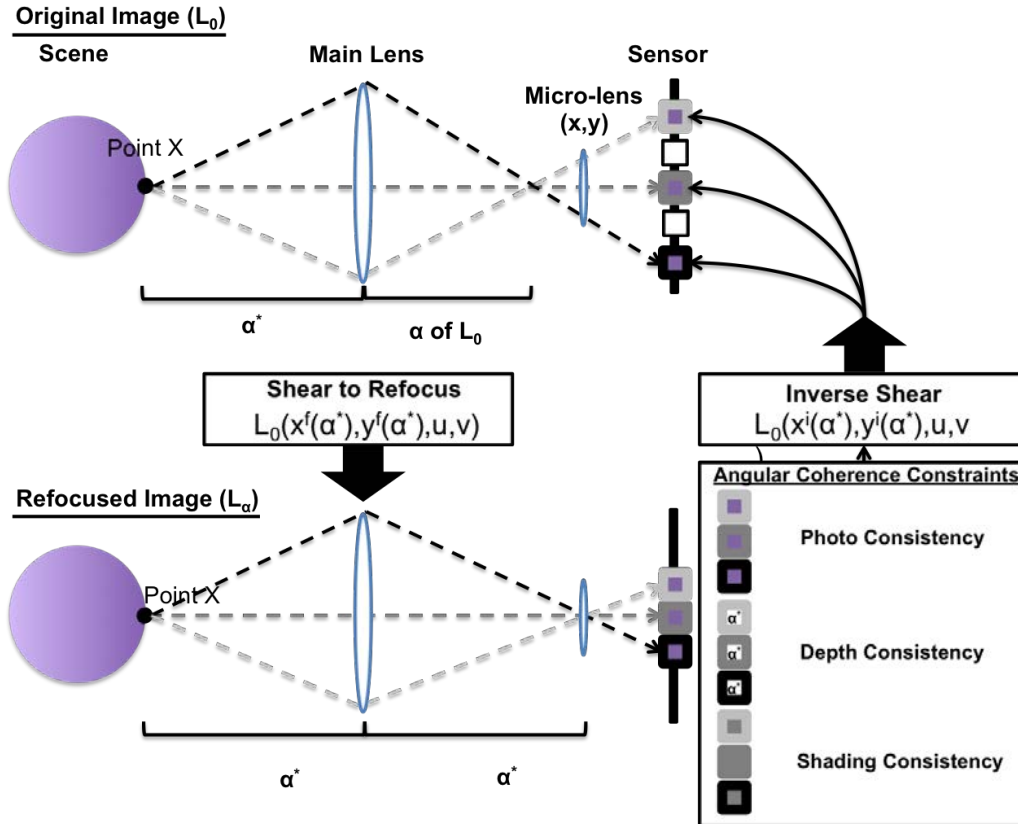
Figure 3.5: Angular Coherence and Refocusing. *In a scene where the main lens is focused to point X with a distance α^* from the camera, the micro-lenses enable the sensor to capture different viewpoints represented as angular pixels as shown on the bottom. As noted by Seitz and Dyer [69], the angular pixels exhibit angular coherence, which gives us photo, depth, and shading consistency. In our chapter, we extend this analysis by finding a relationship between angular coherence and refocusing, as described in Chapter 3.4. In captured data, pixels are not guaranteed to focus at α (shown on the top). Therefore, we cannot enforce angular coherence on the initial captured light-field image. We need to shear the initial light-field image using Eq. 3.7, use the angular coherence constraints, and remap the constraints back to the original coordinates using Eq. 3.13.*

$$L_{\alpha^*}(x, y, u, v) = L(x^f(\alpha^*(x, y)), y^f(\alpha^*(x, y)), u, v). \tag{3.8}$$

We call this equation the *angular coherence* equation. Effectively, $L_{\alpha*}$ represents the remapped light-field data of an all-in-focus image. However, utilizing this relationship is difficult because α^* is unknown. From Eq. 3.7, the center pinhole image P, where the angular coordinates are at $(u, v) = (0, 0)$, exhibits a unique property: the sheared $x^f(\alpha), y^f(\alpha)$ are independent of (u, v). At every α,

$$L_\alpha(x, y, 0, 0) = P(x, y). \tag{3.9}$$

The central angular coordinate always images the same point in the scene, regardless of the focus. This property of refocusing allows us to exploit *photo consistency*, *depth consistency*, and *shading consistency*, shown in Fig. 3.5. The motivation is to use these properties to formulate depth estimation and shading constraints.

Photo consistency. In L_{α^*}, since all angular rays converge to the same point in the scene at each spatial pixel, the angular pixel colors converge to $P(x, y)$. In high noise scenarios, we use a simple median filter to de-noise $P(x, y)$. Therefore, we represent the photo consistency measure as

$$L_{\alpha^*}(x, y, u, v) = P(x, y). \tag{3.10}$$

Depth consistency. Additionally, the angular pixel values should also have the same depth values, which is represented by

$$\bar{\alpha}^*(x, y, u, v) = \alpha^*(x, y) \tag{3.11}$$

where $\bar{\alpha}^*$ is just an up-sampled α^* with all angular pixels, (u, v), sharing the same depth for each (x, y). (Although depths vary with the viewpoint, (u, v), we can assume the variation of depths between angular pixels is minimal since the aperture is small and our objects are comparatively far away.)

Shading consistency. Following from the photo consistency of angular pixels for each spatial pixel in L_{α^*}, shading consistency also applies, since shading is viewpoint independent for Lambertian surfaces. Therefore, when solving for shading across all views, *shading consistency* gives us

$$S(x^f(\alpha^*(x,y)), y^f(\alpha^*(x,y)), u, v) = S(x, y, 0, 0). \tag{3.12}$$

Inverse Mapping

For all three consistencies, the observations only apply to the coordinates in L_{α^*}. To map these observations back to the space of L, we need to use the coordinate relationship between L_{α^*} and L, as shown in Fig. 3.5 on the bottom.

$$L(x^i(\alpha^*), y^i(\alpha^*), u, v) = L_{\alpha^*}(x, y, u, v)$$

$$x^i(\alpha) = x - u(1 - \frac{1}{\alpha}) \quad y^i(\alpha) = y - v(1 - \frac{1}{\alpha}) \tag{3.13}$$

We use this property to map depth and shading consistency to L.

3.5 Algorithm

Our algorithm is shown in Algorithm 3 and Fig. 3.6. We discuss local estimation using angular coherence (3.5) and regularization (3.5), corresponding to lines 2 and 3 of the algorithm. Section 3.6 describes shading and lighting estimation and the final optimization.

Depth Cues using Angular Coherence [Line 2]

We start with local depth estimation, where we seek to find the depth α^* for each spatial pixel. We improve the robustness of the defocus and correspondence cues. We use *photo*

Algorithm 3
Depth from Shading, Defocus, and Correspondence

1: **procedure** DEPTH(L)
2:      Z, Z_{conf} = LocalEstimation(L)
3:      Z^* = OptimizeDepth(Z, Z_{conf})
4:      S = EstimateShading(L)
5:      l = EstimateLighting(Z^*, S)
6:      Z^* = OptimizeDepth($Z^*, Z_{\text{conf}}, l, S$)
7:      **return** Z^*
8: **end procedure**

---

consistency (Eq. 3.10) to formulate an improved metric for defocus and correspondence. From angular coherence (Eq. 4.11), we want to find α^* such that

$$\alpha^*(x, y) = \operatorname*{argmin}_{\alpha} |L(x^f(\alpha), y^f(\alpha), u, v) - P(x, y)|. \tag{3.14}$$

The equation enforces all angular pixels of a spatial pixel to equal the center view pixel color, because regardless of α the center pixel color P does not change. We will now reformulate defocus and correspondence to increase robustness of the two measures.

Defocus. Instead of using a spatial contrast measure to find the optimal depth as shown in Eq. 3.5, we use Eq. 3.14 for our defocus measure. The first step is to take the EPI and average across the angular (u, v) pixels (instead of just the 2D EPI in Eq. 3.2),

$$\bar{L}_\alpha(x, y) = \frac{1}{N_{(u,v)}} \sum_{(u', v')} L_\alpha(x, y, u', v'), \tag{3.15}$$

(a) Light-Field Input L_0 (Center View)

Local Depth Estimation

(b) Confidence (Z_{conf}) [line 2]

(c) Local Depth (Z) [line 2]

Regularize (w/o Shading)

Estimate Shading

Estimate Lighting

(d) Initial Depth (Z')   (e) Our Shading (S)   (f) Our Lighting (l)

Regularize (w/ Shading)

(g) Our Final Depth (Z*)
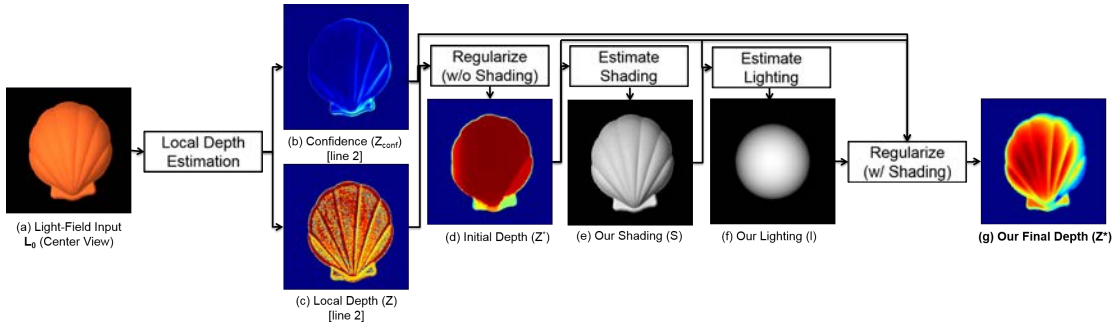
Figure 3.6: Pipeline. *The pipeline of our algorithm contains multiple steps to estimate the depth of our input light-field image (a). The first is to locally estimate the depth (line 2), which provides us both confidence (b) and local depth estimation (c). We use these two to regularize depth without shading cues (d) (line 3). The depth is planar, which motivates us to use shading information to refine our depth. We first estimate shading (e) (line 4), which is used to estimate lighting (f) (line 5). We then use the lighting, shading, initial depth, and confidence to regularize into our final depth (g) (line 6).*

where $N_{(u,v)}$ denotes the number of angular pixels (u, v). Finally, we compute the defocus response by using a measure:

$$D_\alpha(x, y) = \frac{1}{|W_D|} \sum_{(x',y') \in W_D} |\bar{L}_\alpha(x', y') - P(x', y')| \tag{3.16}$$

where W_D is the window size (to improve robustness). For each pixel in the image, we compare a small neighborhood patch of the refocused image and its respective patch at the same spatial location of the center pinhole image.

Even with refocusing artifacts or high frequency out-of-focus blurs, the measure produces low values for non-optimal α. In Fig. 3.7, we can see that the new measure responses are more robust than simply using the contrast, as in Eq. 3.3 in Chapter 3.3.
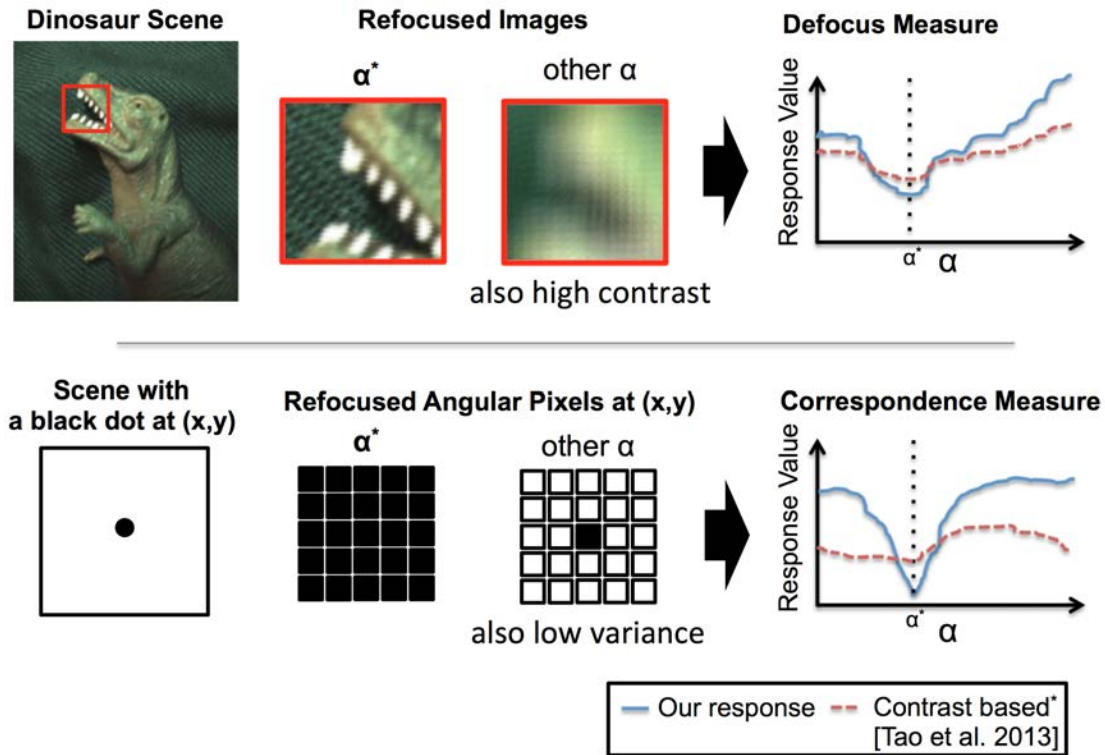
Figure 3.7: Depth estimation using angular coherence. *On the top, we have a scene with a dinosaur. Even refocused to an non-optimal depth, not equal to α^*, high contrast still exists. By using a contrast based defocus measure, the optimal response is hard to distinguish. On the bottom, we have a scene with a black dot in the center. When refocused at a non-optimal depth, the angular pixels may exhibit the same color as the neighboring pixels. Both the optimal and non-optimal α measures would have low variance. However, by using angular coherence to compute the measures, we can see that, in both cases, the resulting measure better differentiates α^* from the rest, giving us better depth estimation and confidence (also in Fig. 3.10). Note: For defocus measurement, we inverted the contrast-based defocus response for clearer visualization.*

Correspondence. By applying the same concept as Eqn. 3.14, we can also

formulate a new correspondence measure. To measure photo consistency, instead of measuring the variance of the angular pixels as shown in Eq. 3.5, we measure the difference between the refocused angular pixels at α and their respective center pixel. This is represented by

$$C_\alpha(x, y) = \frac{1}{N_{(u', v')}} \sum_{(u', v')} |L_\alpha(x, y, u', v') - P(x, y)|. \tag{3.17}$$

The previous correspondence measure only considers the variance in L_α directly, while we also compare to the intended value. The advantage is the measurement is more robust against small angular pixel variations such as noise. See Fig. 3.7 bottom, where at an incorrect depth, the angular pixels are similar to neighboring pixels. Measuring the variance will give an incorrect response as opposed to our approach of comparing against the center view.

Regularization w/ Confidence Measure [Line 3]

Similar to Chapter 3.3, $\alpha_D^\star(x)$ and $\alpha_C^\star(x)$ are the minimum of the defocus and correspondence responses. To measure the confidence of $\alpha_D^\star(x)$ and $\alpha_C^\star(x)$, we use Attainable Maximum Likelihood (AML).

Up to this point, we have obtained a new local depth estimation. Now the goal is to propagate the local estimation to regions with low confidence. To combine the two responses, for each spatial pixel, we use a simple average of the defocus and correspondence responses weighted by their respective confidences. To find the optimal depth value for each spatial pixel, we use the depth location of the minimum of the combined response curve, which we will label as Z. We used the same AML measure for the new combined response to compute the overall confidence level for local depth estimation, which we then label as Z_{conf}. Z and α are in the same scale; therefore, all equations above can be used with Z.

In our optimization scheme, given Z, the local depth estimation, and its confidence, Z_{conf}, we want to find a new Z^* that minimizes

$$E(Z^*) = \sum_{(x,y)} \lambda_d E_d(x, y) + \lambda_v E_v(x, y) \tag{3.18}$$

where Z^* is the optimized depth, E_d is our data constraint, and E_v is our smoothness

constraint. In our final optimization, we also use E_s, our shading constraint (line 6). In our implementation, we used $\lambda_d = 1$ and $\lambda_v = 4$.

Data constraint (E_d). To weight our data constraint, we want to optimize depth to retain the local depth values with high confidence. Note that since we use light-field data, we have a confidence metric from defocus and correspondence. Therefore, we can establish the data term as follows,

$$E_d(x, y) = Z_{\text{conf}}(x, y) \cdot ||Z^*(x, y) - Z(x, y)||^2. \tag{3.19}$$

Smoothness constraint (E_v) The smoothness term is the following:

$$E_v(x, y) = \sum_{i=1,2,3} ||(Z^* \otimes F_i)(x, y)||^2. \tag{3.20}$$

In our implementation, we use three smoothness kernels,

$$F_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} F_2 = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} F_3 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \tag{3.21}$$

where F_1 is the second derivative and F_2 and F_3 are horizontal and vertical first derivatives respectively.

3.6   Finding Shading Constraints

The problem with just using the data and smoothness terms is that the smoothness terms do not accurately represent the shape (Fig. 3.6d). Since smoothness propagates data with high local confidence, depth regularization becomes planar and incorrect (See Fig. 3.1). Shading information provides important shape cues where our local depth estimation does not. Before we can add a shading constraint to the regularizer, we need to estimate shading and lighting.

Shading w/ Angular Coherence [Line 4]

The goal of the shading estimation is to robustly estimate shading with light-field data. We use the standard decomposition, $P = AS$, where P is the central pinhole image,

A is the albedo, and S is the shading. However to improve robustness, we extend the Retinex image decomposition framework [94] to use the full light-field data $L = AS$ by introducing a new angular coherence term. The new angular coherence term increases robustness against noise as shown in Fig. 3.8.

Our optimization solves for $S(x, y, u, v)$. In this section, to simplify our notation, we use I to denote L, following the standard intrinsic image notation. We use the log space $\log I = \log(A \cdot S)$. We also use $a = i - s$ where the lower case (i, a, s) terms are the log of (I, A, S) RGB values. We solve for s by using the following error metric,

$$E(s) = \sum_{t=(x,y,u,v)} E_{ls}(t) + E_{la}(t) + E_{ns}(t) + E_{na}(t) + E_{ac}(t). \qquad (3.22)$$

We use a least squares solver to optimize for $s(x, y, u, v)$. To map to $s(x, y)$ (the shading decomposition of P), we take the central viewpoint, $s(x, y, 0, 0)$. We use the shading component of P for lighting and depth refinement.

Depth propagation. Since the shading constraints depend on normals of the entire (x, y, u, v) space, we need to propagate depth and constraints from $Z^*(x, y)$ to $Z^*(x, y, u, v)$. By looking at Fig. 3.5, we need to map $Z^*(x, y)$ to $\bar{Z}^*(x, y, u, v)$ by using Eq. 3.12. To map $\bar{Z}^*(x, y, u, v)$ back to the inverse coordinates, we use

$$Z^*(x^i(\alpha^*), y^i(\alpha^*), u, v) = \bar{Z}^*(x, y, u, v). \qquad (3.23)$$

Local shading and albedo constraint (E_{ls}, E_{la}). To smooth local shading, we look at the 4-neighborhood normals. If the normals are similar, we enforce smoothness.

$$\begin{aligned}
E_{ls}(t) &= w_{ls}(t) \cdot ||(s \otimes F_1)(t)||^2 \\
E_{la}(t) &= w_{la}(t) \cdot ||((i - s) \otimes F_1)(t)||^2
\end{aligned} \qquad (3.24)$$

where w_{ls} is the average of the dot product between normal of p and w_{la} is the average of the dot product between the pairwise center pixel's and its neighbors' RGB chromaticities. F_1 is the second derivative kernel from Eqn. 3.21.

Nonlocal shading and albedo constraint (E_{ns}, E_{na}). To smooth nonlocal shading, we search for the global closest normals and enforce smoothness. For the pixels with similar normals, we enforce similarity.

$$
\begin{aligned}
E_{ns}(t) &= \sum_{p,q\in\aleph_{ns}} w_{ns}(p,q)\cdot ||s(p)-s(q)||^2 \\
E_{na}(t) &= \sum_{p,q\in\aleph_{na}} w_{na}(p,q)\cdot ||(i-s)(p)-(i-s)(q)||^2
\end{aligned}
\tag{3.25}
$$

where p and q represent two unique (x,y,u,v) coordinates within $\aleph_{ns}$ and $\aleph_{na}$, the top 10 pixels with nearest normal and chromaticity respectively. w_{ns} and w_{na} are the dot product between each pairwise normals and chromaticities.

Angular coherence constraint (E_{ac}). So far, we are operating largely similar to shape from shading systems in a single (non light-field) image. We only constrain spatial pixels for the same angular viewpoint. Just like our depth propagation, we can enforce *shading consistency*. We do this by the constraints represented by Eq. 3.12, as shown in Fig. 3.5. For each pair of the set of (x,y,u,v) coordinates, we impose the shading constraint as follows,

$$
E_{ac}(t) = \sum_{p,q\in\aleph_{ac}} ||s(p)-s(q)||^2
\tag{3.26}
$$

where p,q are the coordinate pairs (x,y,u,v) in $\aleph_{ac}$, all the pixels within the shading constraint. The term plays a large role in keeping our shading estimation robust against typical artifacts and noise associated with light-field cameras. Without the term, the shading estimation becomes noisy and creates errors for depth estimation (Figs. 3.8, 3.9).

Lighting Estimation [Line 5]

With shading, S, we use spherical harmonics to estimate general lighting as proposed by Ramamoorthi and Hanrahan [63] and Basri and Jacobs [7] as follows,

$$
P = A(x,y)\sum_{k=0}^{8} l_k H_k(Z^*(x,y)).
\tag{3.27}
$$

(a) Shading Image (S)
(Center View)

(b) Without
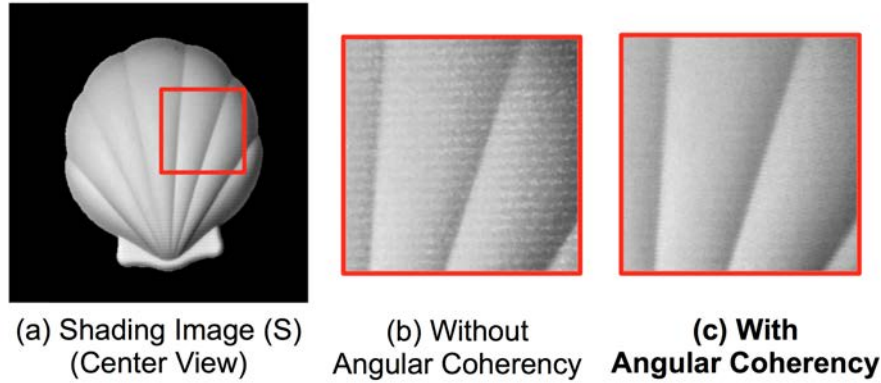Angular Coherency

**(c) With
Angular Coherency**

Figure 3.8: Angular Coherence and Robust Shading. *From the shading image we generate (a), without angular coherency causes noise and unwanted artifacts (b). With angular coherence, the noise reduces. Quantitatively, we can see these effects in Fig. 3.9.*

where P is the central pinhole image, A is the albedo, l are the spherical harmonic coefficients of the lighting, and H_k are the spherical harmonics basis functions that take a unit surface normal (n_x, n_y, n_z) derived from $Z^*(x, y)$.

We have computed S. A is estimated as $P = AS$. Therefore, l is the only unknown and can be estimated from these equations using a linear least squares solver.

Regularization w/ Shading Constraints [Line 6]

With both shading S and lighting l, we can regularize with the shading cue. The new error metric is

$$E(Z^*) = \sum_{(x,y)} \lambda_d E_d(x, y) + \lambda_v E_v(x, y) + \lambda_s E_s(x, y) \tag{3.28}$$

where E_d and E_v are the same as Eq. 3.27 and E_s is our shading constraint. We use $\lambda_s = 2$ in our implementation. We use a non-linear least squares approach with a 8 nearest-neighbors Jacobian pattern to solve for the minimization.

Shading constraint (E_s). To constrain the depth with shading, we want Z^* to satisfy $\sum_{k=0}^{8} l_k H_k(Z^*(x, y)) = S$. Hence, the error term is
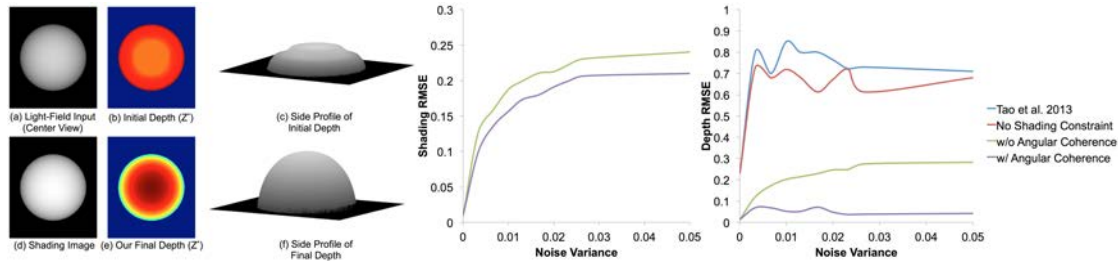
Figure 3.9: Qualitative and quantitative synthetic measurement. *We have a simple diffuse ball lit by a distant point light-source (a). With just regularization without shading information, our depth estimation does not represent the shape (b,c). With our shading image (d), our depth estimation recovers the ball's surface (e,f). We added Gaussian noise with a variable variance. Without the shading constraint, the RMSE (root mean squared error) against ground truth shading and depth are high. Angular coherence results lower RMSE for both shading and depth.*

$$E_s(x, y) = w_s(x, y) \cdot || \sum_{k=0}^{8} l_k H_k(Z^*(x, y)) - S ||^2 \qquad (3.29)$$

where $w_s(x, y) = (1 - Z_{\text{conf}}(x, y))$ to enforce the shading constraint where our local depth estimation is not confident.

3.7 Results and Validation

We validated our algorithm (depth regularized without shading constraints, shading estimation, and depth regularized with shading constraints) using a synthetic light-field image (Fig. 3.9), and we compare our depth results to the state-of-the-art methods by the Lytro Illum Software and Wanner et al. [83] (Figs. 3.10, 3.11, 3.12, 3.14). We quantitively evaluated both uniform and non-uniform albedo examples on real images. To capture all the natural images in the chapter, we reverse engineered the Lytro Illum decoder (see Chapter 2) and used varying camera parameters to capture scenes under different lighting conditions.

Quantitative Analysis

Synthetic: Noise

To validate the depth and shading results of our algorithm, we compare our results to the ground truth depth and shading for a synthetic light-field image of a Lambertian white

sphere illuminated by a distant point light source. We added Gaussian noise (zero mean with variance from 0 to 0.03) to the input image. In Fig 3.9, we see that using shading information helps us better estimate the shape of the sphere. With angular coherence constraints on our shading, both depth and shading RMSE are reduced, especially with increased noise.

Lytro Illum Images

Approach. In Figures 3.10, 3.11, and 3.12, we first 3D scanned all four figurines (cupcake, flat cat, standing dog, and standing cat), using the NextEngine 3D scanner. We used the three-bracket mode with 40 points per square inch. For each of the algorithms, we used an iterative closest point (ICP) approach to map the depth maps to the ground truth scan [14]. We compute the point-to-point error for each point of the ground-truth scan points, as well as the root-mean-squared error (RMSE). The parameters we used for the ICP are point-to-point minimization metric, Euclidean distance tolerance of 0.01, Radian distance tolerance of 0.009, and maximum of 100 iterations. We will release both the dataset and code to generate the RMSE and visualizations.

Figure 3.10: Uniform Albedo Comparisons. *We compare qualitative and quantitative measures with two different examples against Lytro Illum Software, Wanner and Goldluecke [83], contrast-based defocus and correspondence from Chapter 3.3, and angular coherence based defocus and correspondence from Chapter 3.5. On the top, we have an example of a cupcake, where our algorithm is able to estimate the contours of the cupcake decorations. On the bottom, we have an image of a flat cat figurine. We can see that our algorithm is able to recover the curvature of the body and face. For comparison against ground truth, we use the NextEngine 3D scanner to obtain the ground truth and align each of the resulting depth maps using the iterative closest point (ICP) algorithm. The color diagram shows the Euclidean distance of each ground truth point to the closest point after the ICP transformation for each algorithm. We can see that we align closely with the ground truth with the lowest RMSE.*
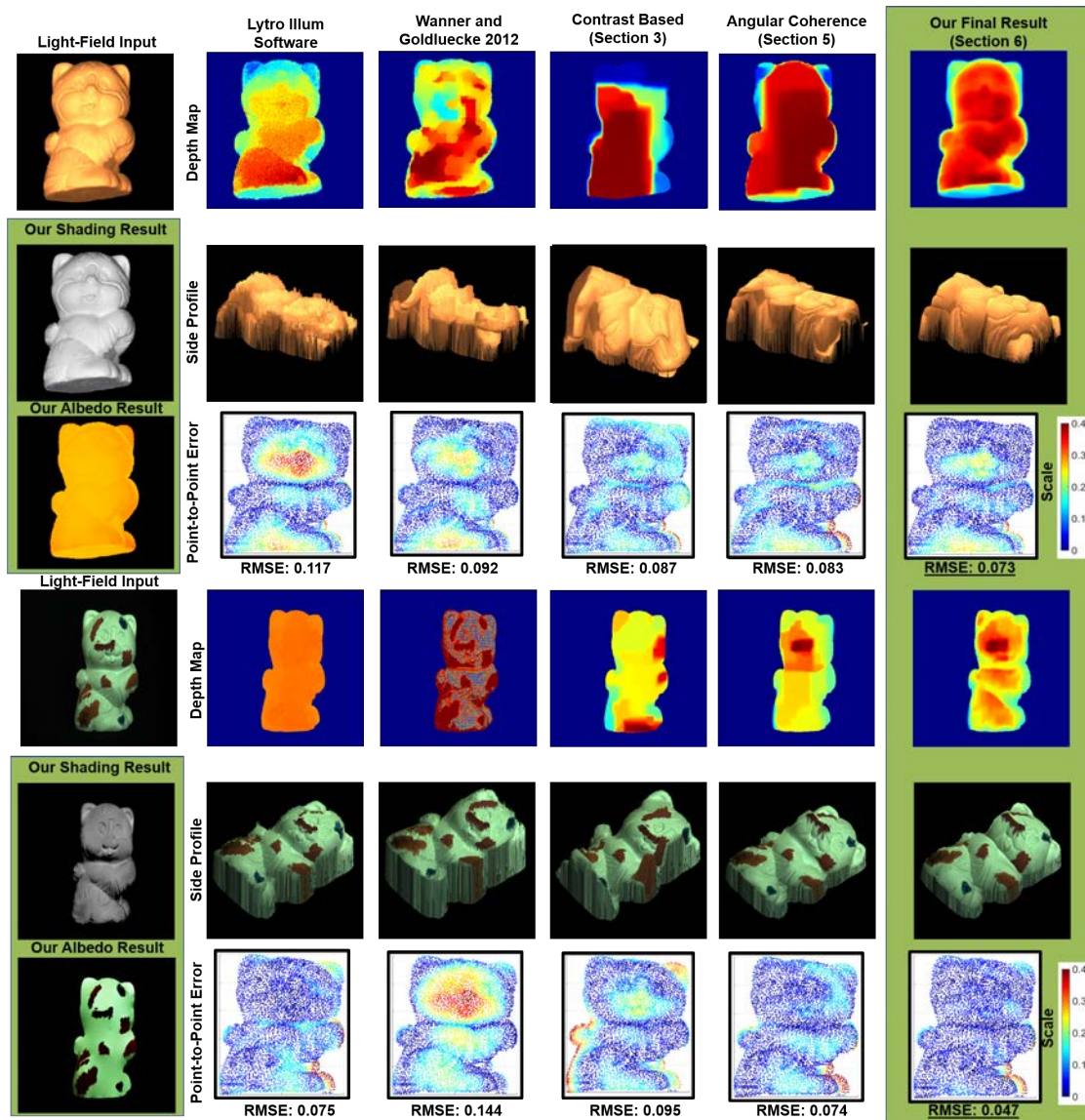
Figure 3.11: Varying Albedo Comparisons: Cat. *In this figure, we took two pictures of the same figurine of the standing cat. Starting from the uniform albedo results, our algorithm is able to recover the contours of the cat, with nice side curvature. Our point-to-point errors also show low errors across the cat. On the bottom, we painted the cat with different colors. Our algorithm was able to recover a reasonable shading estimation from the image. We can also see that our depth estimation can still resolve the contours of the cat with low RMSE.*
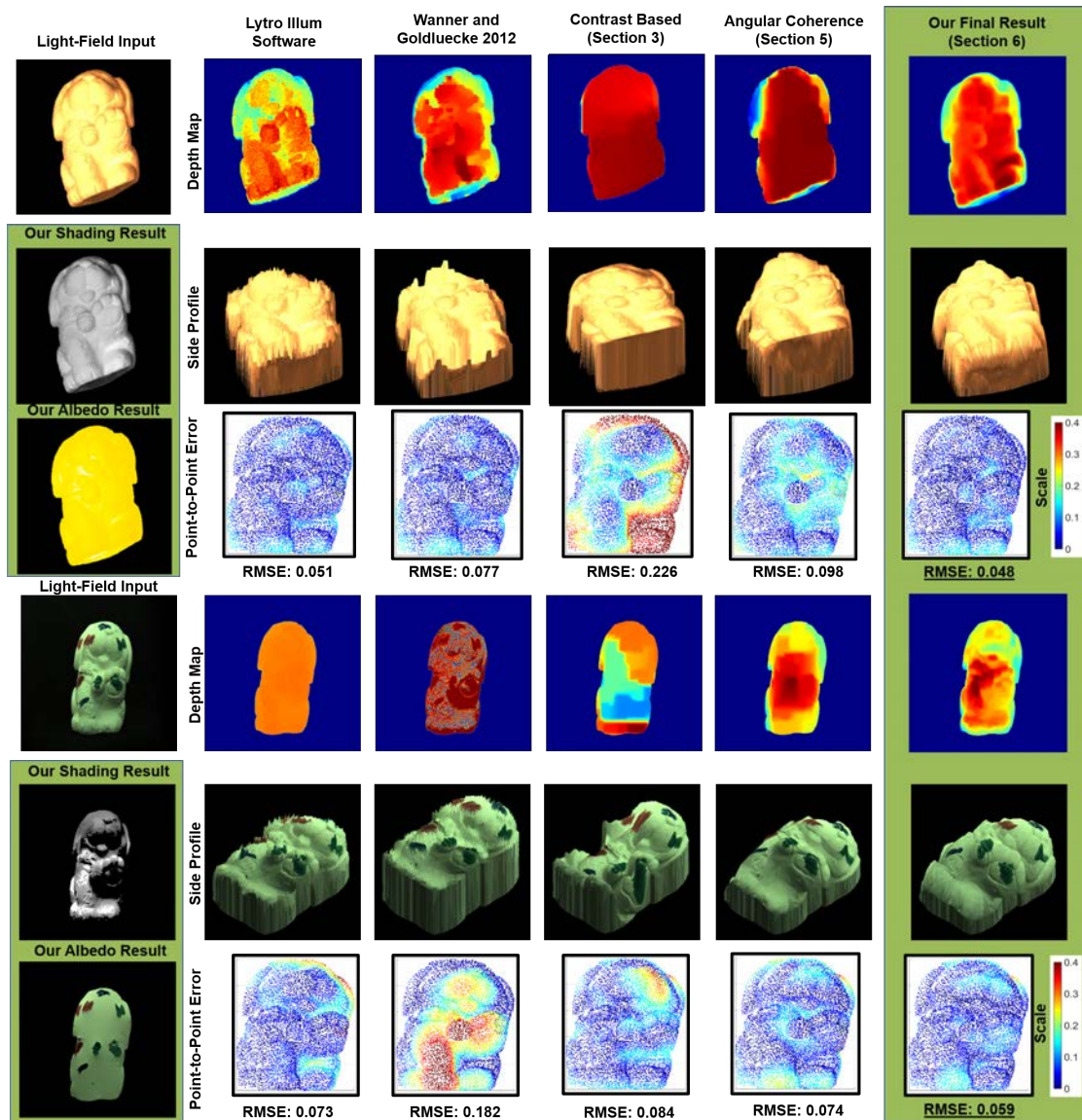
Figure 3.12: Varying Albedo Comparisons: Dog. *In this figure, we took two pictures of the standing dog, both without or with varying albedo. Starting from the uniform albedo results, our algorithm is able to recover the contours of the dog. Our point-to-point errors also show low errors across the dog. On the bottom, we painted the dog with different colors. Our algorithm was able to recover a reasonable shading estimation from the image. We can also see that our depth estimation can still resolve the contours of the dog with low RMSE. With the contrast based method, we can see instabilities of regularization when no texture is present, resulting in high RMSE; however, RMSE reduces significantly when albedo changes are present. With angular coherence, we see over-smoothing after regularization due to lack of texture, causing larger errors than Lytro and Wanner and Goldluecke; but the depth result also significantly improves with albedo changes.*

Analysis. The plot below each example shows the point-to-point error of each point of the ground truth scan and RMSE.

For the uniform albedo results in Fig. 3.10, on the top, we have an input image of the cupcake with decorations. We can see that our shape estimation captures the curvature of the cupcake. Our defocus and correspondence using angular coherence from Chapter 3.5 gives a flatter result, but an improvement over using the contrast based defocus and correspondence from Chapter 3.3, which shows noisier results; Wanner and Goldluecke [83] also shows high errors in smooth regions; and the Lytro Illum software shows noisier results that are not suitable for estimating normals. Quantitatively, these observations are consistent with the point-to-point error, where our method shows low errors for the cupcake with a low RMSE. We observe the same on the bottom rows with the cat example. Although all examples show difficulties resolving the shape of the nose, our shape estimation still performs better with a low RMSE.

For non-uniform albedo results in Figs. 3.11 and 3.12, we have two different captured images: one for uniform albedo and one with varying albedo, painted on the figurines. In these figures, we gain a better understanding how albedo impacts the results. With the cat example, we can see that our algorithm is robust, even with the painted colors. Because of our shading estimation, we are still able to retain the curvature of the cat. Although some errors are introduced in the shading result, our RMSE is still lower than the other methods' RMSE. Note that contrast-based methods run into regularization errors, due to low confidence regions. In Fig. 3.12, we observe the same behavior and we are able to extract a reasonable shading and albedo result to reduce the impact of the albedo changes.

Qualitative Analysis

3D Printing. In Fig. 3.13, we qualitatively assess our shape estimation by comparing the three printed objects (standing cat, flat cat, and cupcake) against the original figurines. We first converted our depth estimation to a 3D point cloud and then used MeshLab to compute the normals for the set of points. We then created the mesh using Poisson Surface Reconstruction [38], and used the Makerbot Replicator Z18 3D printer to print the figurines. We scale our prints such that they can fit in a 50mm cube. We can see with the three prints, given the limited spatial resolution (430x539) of the Lytro Illum Camera and 0.2mm printing precision, we are able to print low resolution 3D prints of the original figurines. Therefore, the surfaces still look smooth. However, with higher spatial resolution cameras, more points can result in higher quality prints. For normals computation, we used 10 neighbors with 0 smooth iterations. For the surface reconstruction, we used the Poisson method with an octree depth of 6, solver divide of 6, 1

Figure 3.13: 3D Printing. *We 3D printed the standing cat, flat cat, and cupcake examples. The printed examples showcase the capability of using passive camera systems with a single exposure to capture shape for 3D printing. Note that we scaled the printed figurines to fit in a 50mm cube with 0.2mm precision.*

sample per node, and 1 surface offset distance, which all are defined in [38].

Natural Images. In Fig. 3.14, we show that our algorithm works with natural images across different camera settings. On the top, we have an orange plastic shell, illuminated by indoor lighting. The Illum software produces noisy results. Wanner and Goldlucke's regularization propagates errors in regions where local estimation fails. In the contrast-based results, we see stronger fluctuations in the defocus and correspondence confidence measure, causing depth blockiness in some areas. Even without shading constraints, we produce a less noisy result. Our depth estimation recovers the shell shape, including the ridges and curvature. In the middle, we have an example of a dinosaur toy with varying albedo. The dinosaur teeth, claws, and neck ridges are salient in our results, while other algorithms have trouble recovering these shapes. Using shading gives a significant benefit in recovering the object shapes. On the bottom, we have an outdoor image of leaves. Our algorithm captures the shapes of the leaves while other algorithms produce noise and spikes.
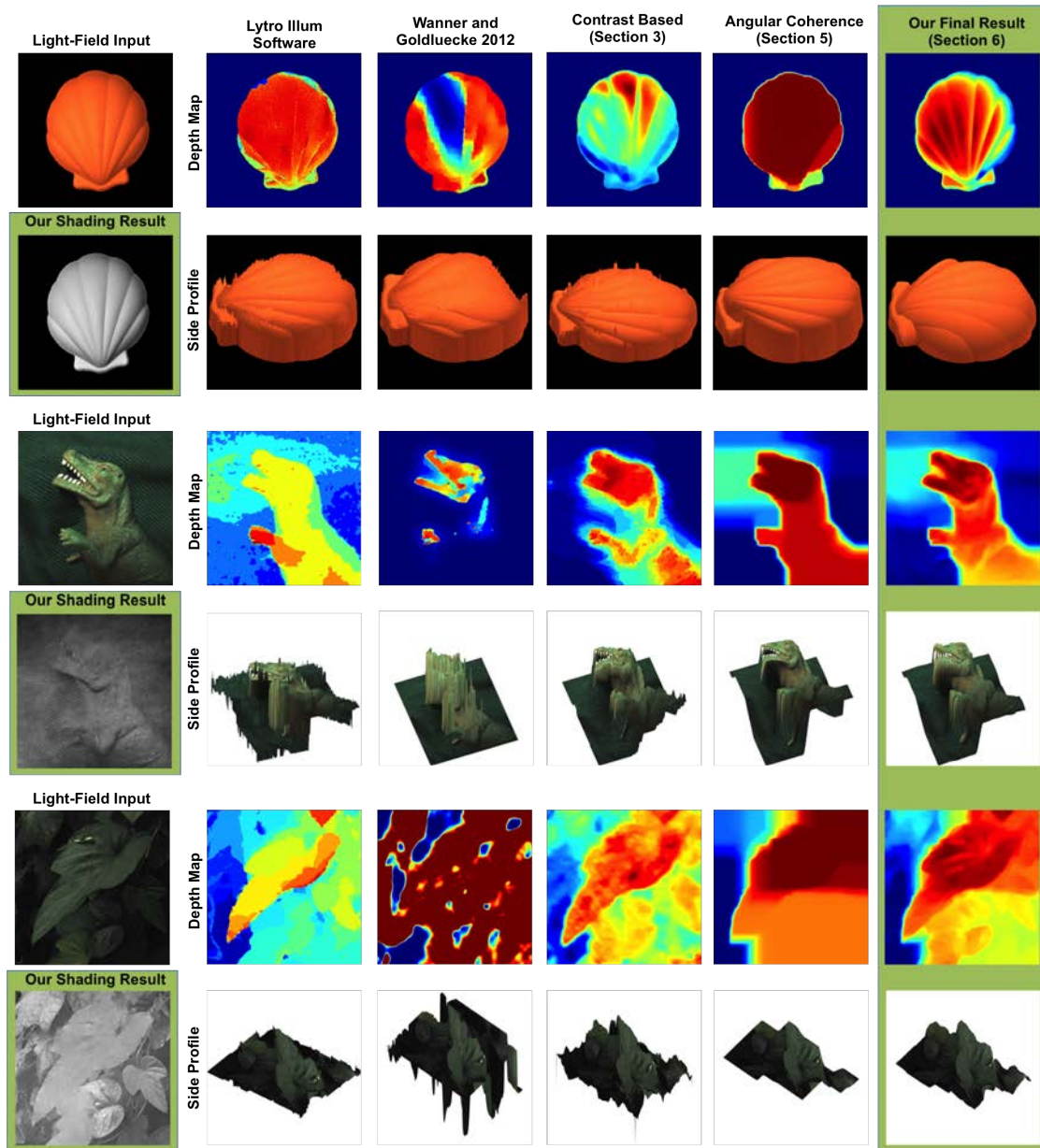
Figure 3.14: More Natural Image Examples. *On the top, we have an indoor picture of a shell. We can see that with our final result, we are able to recover the ridges of the shell. Without shading information, the shell is rendered as flat. Contrast based and Wanner and Goldluecke show errors where not enough texture is present on the shell. The Lytro Illum gives noisy results. We observe similar patterns with the dinosaur example where we have non-uniform albedo. We can see that our shading estimation shows the shadows of the dinosaur and folds of the background cloth. On the bottom, we have an outdoors example, capturing a leaf. Again, we see that our depth estimation closely represents the surface of the leaf.*

3.8 Conclusion and Future Work

We have proposed and provided quantitative validation for a new shape estimation framework that uses just a single-capture passive light-field image. Our optimization framework can be used for consumer grade light-field images to incorporate all three cues: defocus, correspondence, and shading.

For future work, more robust approaches could be used for scenes with more varying albedos and occlusions. Additionally, as seen in Fig. 3.9, image noise still corrupts both our depth and shading estimations; more advanced de-noising could be used in the future. Our shape-from-shading technique does not account for inter-reflections and shadows; therefore, future work includes incorporating better occlusion detection.

In summary, we have proposed a shape estimation algorithm for light field cameras that incorporates defocus, correspondence, and shading, suitable for passive point-and-shoot acquisition from consumer light-field cameras. In this chapter, we assume Lambertian surfaces. The following chapter describes a new method that estimates depth of scenes with glossy regions.

Chapter 4

Depth Estimation and Specular Removal for Glossy Surfaces Using Point and Line Consistency

We have a demonstrated practical algorithm for depth recovery from a passive single-shot capture. However, many light-field depth estimation methods are designed for Lambertian objects and fail or degrade for glossy or specular surfaces because photo-consistency depth measurement is a poor metric for such surfaces. In this chapter, we present a novel theory of the relationship between light-field data and reflectance from the dichromatic model. We present a physically-based and practical method to separate specular regions and estimate the light source color. As opposed to most previous work, our algorithm supports multiple lights of different colors. Our novel algorithm robustly removes the specular component for glossy objects. In addition, our approach enables depth estimation to support both specular and diffuse scenes. We show that our method outperforms current state-of-the-art specular removal and depth estimation algorithms in multiple real world scenarios using the consumer Lytro and Lytro Illum light field cameras.

4.1 Introduction

In the previous chapter, we have shown that light-field cameras enable effective passive and general depth estimation. This makes light-field cameras point-and-capture devices to recover shape. However, in the previous chapter, our depth estimation algorithms support only Lambertian surfaces, making them ineffective for glossy surfaces, which have both specular and diffuse reflections. In this chapter, we present the first light-
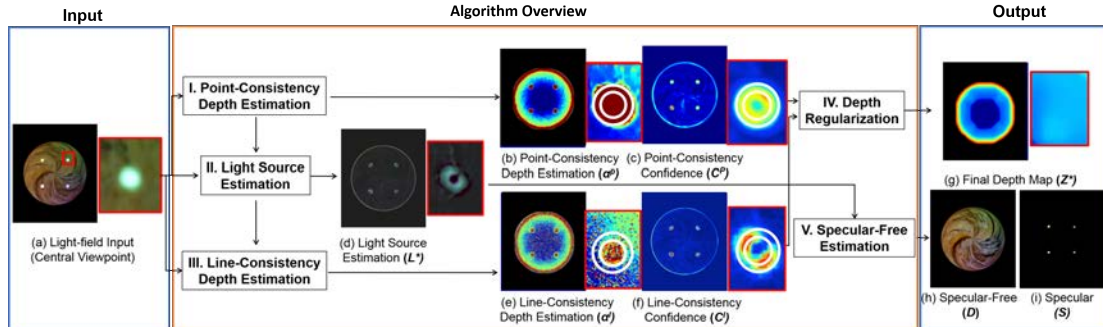
Figure 4.1: *Depth Estimation for Glossy Surfaces. Our input is a light-field image. We use PBRT (physically based ray tracer) [62] to synthesize a red wood textured glossy sphere with specular reflectance $K_s = [1, 1, 1]$ and roughness = 0.001 with four light sources of different colors (a). We use two photoconsistency metrics: point-consistency and line-consistency. By using point-consistency, we obtain depth measures suitable for diffuse only surfaces, but exhibit erroneous depth (b) and high confidence (c) at glossy regions due to overfitting data. By using the light-source color estimation (d), we seek a depth where the colors from different viewpoints represent a line, with direction corresponding to light-source color, which we call line-consistency. The new depth measurement gives correct depths at specular edges (e), but exhibits low confidence values everywhere else. We highlighted the difference of the edges by highlighting in white. The comparison between the two can be seen in Fig. 4.3. We use both of the cues to perform a depth regularization that produces an optimal result by exploiting the advantages of both cues (g). With the analysis, we can also extract a specular-free image (h) and an estimated specular image (i). In this chapter, we provide the theoretical background of using the two metrics.* Note: *The specular colors are enhanced for easier visibility throughout the chapter. For depth maps, cool to warm colors represent closer to farther respectively, and for confidence maps, less confident to more confident respectively, with a scale between 0 and 1.*

field camera depth estimation algorithm for *both diffuse and specular* surfaces using the consumer Lytro and Lytro Illum cameras (Fig. 4.1).

We build on the dichromatic model introduced by Shafer [71], but extend and apply it to the multiple views of a single point observed by a light field camera. Since diffuse and specular reflections behave differently in different viewpoints, we first discuss four different surface cases (general dichromatic, general diffuse, Lambertian plus specular, Lambertian only). We show that different cases lead to different structures in RGB space, ranging from a convex cone (for the general dichromatic case), a line pass-

ing through the origin (for the general diffuse case), a general line (for the Lambertian plus specular case), to the standard single point (for the Lambertian only case). Notice that standard multi-view stereo typically measures the variance of different views, and is accurate only when the data is well modeled by a point as for Lambertian diffuse reflection. We refer to this as *point-consistency* since we measure consistency to the model that all views correspond to a single point in RGB space; this distinguishes from the *line-consistency* condition we develop in conjunction with the dichromatic model. The dichromatic model lets us understand and analyze higher-dimensional structures involving specular reflection. In practice, we focus on Lambertian plus specular reflection, where multiple views correspond to a general line in RGB space (not passing through the origin, see Fig. 4.2 (c)).

We show that our algorithm works robustly across many different light-field images captured using the Lytro light-field camera, with both diffuse and specular reflections. We compare our specular and diffuse separation against Mallick et al. [50], Yoon et al. [92], and Tao et al. [79], and our depth estimation against Tao et al. [80, 79], Wanner et al. [24], and Lytro software (Figs. 4.11 and 4.12). Our main contributions are:

1. Dimensional analysis for the dichromatic model
We investigate the structure of pixel values of different views in the color space. We show how different surface models will affect the structure, when focused to either the correct or incorrect depth.

2. Depth estimation for glossy surfaces.
For glossy surfaces, using the point-consistency condition to estimate depth will give us wrong depth. We introduce a new photo-consistency depth measure, line-consistency, which is derived from our dichromatic model analysis. We also show how to combine both point-consistency and line-consistency cues providing us a robust framework for general scenes. Our method is based on our initial work (Tao et al. [79]), but we have more robust and better results, and there is no iteration involved.

3. Color estimation and generating a specular-free image.
We perform the multiple viewpoint light source analysis by using and rearranging the light-field's full 4D epipolar plane images (EPI) to refocus and extract multiple-viewpoints. Our algorithm (Algorithm 4) robustly estimates light source color, and measures the confidence for specular regions. The framework distinguishes itself from the traditional approach of specular and diffuse separation for conventional images by providing better results (Figs. 4.8, 4.9, 4.10, 4.11, 4.12) and supporting multiple light source colors (Figs. 4.5, 4.11).

4.2 Related Work

Depth estimation and specularity removal have been studied extensively in the computer vision community. In our work, we show that light fields give us more information to remove specularities. We generalize the photo-consistency measure, introduced by Seitz and Dyer [70], to both point and line consistency, which supports both diffuse and glossy surfaces. Our algorithm is able to robustly estimate depth in both diffuse and specular regions.

Defocus and correspondence depth estimation.

Depth estimation has been studied extensively through multiple methods. Depth from defocus requires multiple exposures [85, 90]; stereo correspondence finds matching patches from one viewpoint to another viewpoint(s) [29, 59, 48, 53]. The methods are designed for Lambertian objects and fail or degrade for glossy or specular surfaces, and also do not take advantage of the full 4D light-field data.

Multi-view stereo with specularity.

Exploiting the dichromatic surface properties has also been studied through multi-view stereo. Lin et al. [47] propose a histogram based color analysis of surfaces. However, to achieve a similar surface analysis, accurate correspondence and segmentation of specular reflections are needed. Noise and large specular reflections cause inaccurate depth estimations. Jin et al. [36] propose a method using a radiance tensor field approach to avoid such correspondence problems, but real world scenes do not follow their tensor rank model. In our implementation, we avoid the need for accurate correspondence for real scenes by exploiting the refocusing and multi-viewpoint abilities in the light-field data.

Diffuse-specular separation and color constancy.

Separating diffuse and specular components by transforming from the RGB color space to the SUV color space such that the specular color is orthogonal to the light source color has been effective; however, these methods require an accurate estimation of or known light source color [50, 51, 58]. Without multiple viewpoints, most diffuse and specular separation methods assume the light source color is known [92, 50, 3, 76, 78, 91, 40]. As noted by Artusi et al. [2], these methods are limited by the light source color, prone to noise, and work well only in controlled or synthetic settings. To alleviate the light source constraint, we use similar specularity analyses as proposed by Sato and Ikeuchi and Nishino et al. [66, 56]. However, prior to our work, the methods require

multiple captures and robustness is dependent on the number of captures. With fewer images, the results become prone to noise. We avoid both of these problems by using the complete 4D EPI of the light-field data to enable a single capture that is robust against noise. Estimating light source color (color constancy) exhibits the same limitations and does not exploit the full light-field data [21, 77]. Since we are estimating the product of light source color and the albedo for each pixel independently, we can estimate more than just one light source color.

Light-field depth estimation.

More recent work has exploited the light-field data by using the epipolar images [80, 83, 24, 84, 39, 12, 79, 65]. Because all these methods assume Lambertian surfaces, glossy or specular surfaces pose a large problem. In our work, we use the full 4D light-field data to perform specular and diffuse separation and depth estimation. Using our line-consistency measure, we directly address the problem of estimating depth for specular regions. In our comparisons, we show that specularities cause instabilities in the confidence maps computed in the previous chapter. The instabilities result from high brightness in specular regions and lower brightness in diffuse regions. Even at the most point-consistent regions, the viewpoints do not exhibit the same color. However, because of the large contrast between the neighborhood regions, these regions still register as high confidence at wrong depths. The incorrect depth and high confidence cause the regularization step by Markov random fields (MRF) to fail or produce incorrect depth propagation in most places, even when specularities affect only a part of the image (Figs. 4.1, 4.11, and 4.12).

We described a preliminary version of our algorithm in [79]. We built upon a theoretical foundation as described in Chapter 4.2 to justify our algorithm. Based on the theory, we improved results and removed the necessity of an iterative approach.

In this section, we explain the dichromatic model and its induced color subspace from multiple views of a point, imaged by a light field camera. By analyzing the pixel values in color space, we can get the type of BRDF of the point. Unlike previous dichromatic analyses, we consider multiple views of a single point, that allows us to estimate multiple light sources over the entire object. We show how to use the insights from our color analysis to develop algorithms for depth estimation from light fields.

Dichromatic Reflection Model

We first analyze the color values at multiple views from a point/pixel on the object. The dichromatic BRDF model [71] states that light reflected from objects has two inde-

pendent components, light reflected from the surface body and at the interface, which typically correspond to diffuse and specular reflection. The observed colors among the viewpoints are then a part of the span between the diffuse and specular components,

$$I(\lambda, \boldsymbol{n}, \boldsymbol{l}, \boldsymbol{v}) = I_d(\lambda, \boldsymbol{n}, \boldsymbol{l}, \boldsymbol{v}) + I_s(\lambda, \boldsymbol{n}, \boldsymbol{l}, \boldsymbol{v}) \tag{4.1}$$

where I is the radiance, λ is the wavelength of light (in practice, we will use red, green and blue, as is conventional), $\boldsymbol{n}$ is the surface normal, and $\boldsymbol{v}$ indicates the viewing direction. We assume a single light source with $\boldsymbol{l}$ being the (normalized) direction to the light. Since our analysis applies separately to each point/pixel on the object, we can consider a separate light source direction and color at each pixel, which in practice allows us to support multiple lights. As is common, we do not consider interreflections or occlusions in the theoretical model. Next, each component of the BRDF ρ can be decomposed into two parts [71]:

$$\rho(\lambda, \boldsymbol{n}, \boldsymbol{l}, \boldsymbol{v}) = k_d(\lambda)\rho_d(\boldsymbol{n}, \boldsymbol{l}, \boldsymbol{v}) + k_s(\lambda)\rho_s(\boldsymbol{n}, \boldsymbol{l}, \boldsymbol{v}) \tag{4.2}$$

where k_d and k_s are diffuse and specular spectral reflectances, which only depend on wavelength λ, ρ_d and ρ_s are diffuse and specular surface reflection multipliers, which are dependent on geometric shapes and independent of color. Now, consider the light source $L(\lambda)$, which interacts with diffuse and specular components of the BRDF:

$$\begin{aligned} I(\lambda, \boldsymbol{n}, \boldsymbol{l}, \boldsymbol{v}) &= L(\lambda) * \rho(\lambda, \boldsymbol{n}, \boldsymbol{l}, \boldsymbol{v}) \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) \\ &= L(\lambda) * [(k_d(\lambda)\rho_d(\boldsymbol{n}, \boldsymbol{l}, \boldsymbol{v}) + k_s(\lambda)\rho_s(\boldsymbol{n}, \boldsymbol{l}, \boldsymbol{v})] \cdot (\boldsymbol{n} \cdot \boldsymbol{l}). \end{aligned} \tag{4.3}$$

Here we use $*$ to represent component-wise multiplication for different wavelengths, usually represented by color channels (R, G, B), and where $\cdot$ indicates a dot product with a scalar (or vector).

Now we consider images taken by light-field cameras. For each point on the object, we can get color intensities from different views. In other words, for a given pixel, $\boldsymbol{n}$ and $\boldsymbol{l}$ are fixed while $\boldsymbol{v}$ is changing. Therefore, we can simplify $\rho_d(\boldsymbol{n}, \boldsymbol{l}, \boldsymbol{v})$ and $\rho_s(\boldsymbol{n}, \boldsymbol{l}, \boldsymbol{v})$ as $\rho_d(\boldsymbol{v})$ and $\rho_s(\boldsymbol{v})$. Furthermore, we encapsulate the spectral dependence for diffuse and specular parts as $\bar{L}_d(\lambda)$ and $\bar{L}_s(\lambda)$:

Type of BRDF	General diffuse plus specular	General diffuse
Dimension analysis	Convex cone (on a plane) $[\bar{L}_d(\lambda)\rho_d(\boldsymbol{v}) + \bar{L}_s(\lambda)\rho_s(\boldsymbol{v})]\cdot(\boldsymbol{n}\cdot\boldsymbol{l})$	Line passing through the origin $\bar{L}_d(\lambda)\rho_d(\boldsymbol{v})\cdot(\boldsymbol{n}\cdot\boldsymbol{l})$
Type of BRDF	Lambertian diffuse plus specular	Lambertian diffuse
Dimension analysis	Line not passing the origin $[c\cdot\bar{L}_d(\lambda) + \bar{L}_s(\lambda)\rho_s(\boldsymbol{v})]\cdot(\boldsymbol{n}\cdot\boldsymbol{l})$	Point $c\cdot\bar{L}_d(\lambda)\cdot(\boldsymbol{n}\cdot\boldsymbol{l})$

Table 4.1: *Dimension analysis of different types of BRDF with one light source.*

$$I(\boldsymbol{v}) = L(\lambda) * [(k_d(\lambda)\rho_d(\boldsymbol{v}) + k_s(\lambda)\rho_s(\boldsymbol{v})]\cdot(\boldsymbol{n}\cdot\boldsymbol{l})$$
$$= \boxed{[\bar{L}_d(\lambda)\rho_d(\boldsymbol{v}) + \bar{L}_s(\lambda)\rho_s(\boldsymbol{v})]\cdot(\boldsymbol{n}\cdot\boldsymbol{l})} \tag{4.4}$$

where

$$\bar{L}_d(\lambda) = L(\lambda) * k_d(\lambda)$$
$$\bar{L}_s(\lambda) = L(\lambda) * k_s(\lambda).$$

Type of BRDF and Dimension Analysis

Assuming the object surface fits the dichromatic reflection model, we can use Eq. 4.4 to analyze pixel values from multiple views. Now we discuss how those pixel values lie in RGB color space, for various simplifying assumptions on the BRDF. Table 4.1 shows a summary. In addition, we use a synthetic sphere to verify the analysis, as shown in the top two rows of Fig. 4.2. In practice, we use the common Lambertian plus specular assumption, but the theoretical framework applies more generally, as discussed below.

$$I(\boldsymbol{v}) = [\bar{L}_d(\lambda)\rho_d(\boldsymbol{v}) + \bar{L}_s(\lambda)\rho_s(\boldsymbol{v})]\cdot(\boldsymbol{n}\cdot\boldsymbol{l}) \tag{4.5}$$

General Diffuse plus Specular. For the general dichromatic case, the color intensity is governed by the general equation above. The color of diffuse component $\bar{L}_d(\lambda)$ is in general different from the specular component $\bar{L}_s(\lambda)$. In addition, $\rho_d(\boldsymbol{v})$ and $\rho_s(\boldsymbol{v})$ are scalars that vary with viewpoint. Therefore, the pixel value is a linear combination of its diffuse color and specular color. Pixel values from different views of a point will lie on a convex cone, a plane spanned by diffuse and specular colors. The synthetic result is shown in Fig. 4.2(a). Since the variance of the diffuse component is usually much smaller than the specular component, most of the variation is dominated by the effect of specular color. When the viewing direction changes and the specular component is no
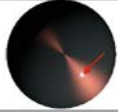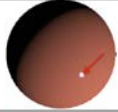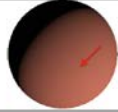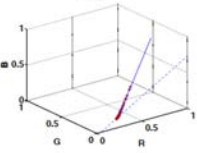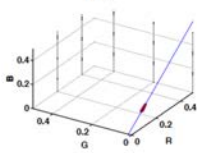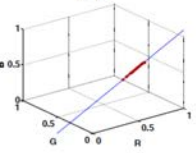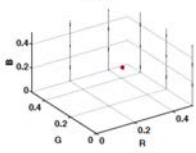
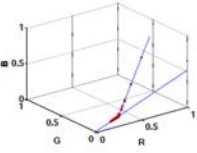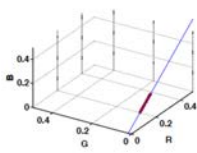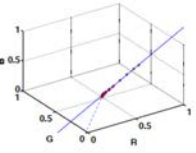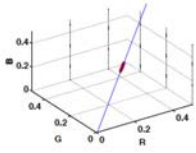| Type of BRDF	General diffuse plus specular	General diffuse	Lambertian diffuse plus specular	Lambertian diffuse
Synthetic center image and the selected point				
	(a)	(b)	(c)	(d)
Pixel values of different views of the selected point refocused at **correct depth**				
Dimension analysis	Convex cone (on a plane)	Line passing through the origin	Line not passing the origin	Point
	(e)	(f)	(g)	(h)
Pixel values of different views of the selected point refocused at **incorrect depth**				
Dimension analysis	Blending of the diffuse color $\bar{L}_d(\lambda)$ and the specular color $\bar{L}_s(\lambda)$	Line passing through the origin, but spans a wider section because neighboring pixels have different $(\boldsymbol{n} \cdot \boldsymbol{l})$	Blending of the diffuse color $\bar{L}_d(\lambda)$ and the specular color $\bar{L}_s(\lambda)$	Line passing through the origin, since it blends neighboring pixels which have different $(\boldsymbol{n} \cdot \boldsymbol{l})$

Figure 4.2: *Synthetic data for dimension analysis of different types of BRDF with one light source. The synthetic data is generated by PBRT [62] to simulate the Lytro camera. Note that for the general diffuse surface, we use a view-dependent spectral reflectance k_d for the sphere. The center images are linearly scaled for display. The scatter plots are pixel intensities in RGB color space from 49 different views, imaging the location where the red arrow points. All pixel values are scaled to $[0, 1]$. Synthetic data shows that when the light field image is refocused to the correct depth, the result corresponds to our dimension analysis (Table 4.1).*

longer dominant, the pixel values will be closer to the diffuse color (around the dashed line), but still on the convex cone.

$$I(\boldsymbol{v}) = \bar{L}_d(\lambda)\rho_d(\boldsymbol{v}) \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) \qquad (4.6)$$

General Diffuse. If the object surface does not have a specular component, the dichromatic model can be simplified as Eq. 4.6. When there is only one light source, $\bar{L}_d(\lambda)$ is fixed and $\rho_d(\boldsymbol{v}) \cdot (\boldsymbol{n} \cdot \boldsymbol{l})$ is a scalar. Thus, all possible values will lie on a line, which

passes through the origin. Figure 4.2(b) shows the result.

$$\boxed{I(\boldsymbol{v}) = [c \cdot \bar{L}_d(\lambda) + \bar{L}_s(\lambda)\rho_s(\boldsymbol{v})] \cdot (\boldsymbol{n} \cdot \boldsymbol{l})} \tag{4.7}$$

Lambertian Diffuse plus Specular. Now we consider the most common case, where the diffuse component is modeled as Lambertian as in most previous work, and there is a specular component. This is the case we will consider in our practical algorithm.

In other words, $\rho_d(\boldsymbol{v})$ is now a constant (replaced by c here), independent of the viewing angle. Under this assumption, the dichromatic model becomes Eq. 4.7. For different views, $\bar{L}_d(\lambda)$ is a constant and $\bar{L}_s(\lambda)\rho_s(\boldsymbol{v})$ is a line passing through the origin. Combining the two components, pixel values in color space will be a line not passing through the origin. Figure 4.2(c) shows the result.

A further simplification is achieved for dielectric materials, where the specular reflection takes the light source color, or $k_s(\lambda)$ is a constant, independent of wavelength. In this case, $\bar{L}_s(\lambda)$ corresponds directly to the light source color, and our practical algorithm is able to estimate the color of the light. In fact, we can handle multiple light sources, since we can assume a separate light affects the specular component for each pixel or group of pixels. Note that the common dielectric assumption is not fundamental to our algorithm, and is needed only to relate the light source color to that of the highlight.

Lambertian Diffuse

$$I(\boldsymbol{v}) = c \cdot \bar{L}_d(\lambda) \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) \tag{4.8}$$

Next, we consider the BRDF with Lambertian diffuse component only. In Eq. 4.7, c and $(\boldsymbol{n} \cdot \boldsymbol{l})$ are all constants. Therefore, all the color intensities should be the same for different views of a point. Indeed, this is just re-stating the notion of diffuse photo-consistency. In effect, we have a single point in RGB space, and we call this *point-consistency* in the rest of the chapter, to distinguish from the *line-consistency* model we later employ for Lambertian plus specular surfaces.

Depth Estimation

Figures 4.2(a-d) verify the dichromatic model applied to light field data, considering multiple views of a single point. However, this analysis assumes we have focused the light field camera to the correct depth, when in fact we want to estimate the depth of a glossy object. Therefore, we must conduct a novel analysis of the dichromatic model, where we understand how multiple views behave in color space, if we are focused at the

incorrect depth. This is shown in the bottom row of Fig. 4.2, and to our knowledge has not been analyzed in prior work.

For a depth estimation method to be robust, the structure when focused to the incorrect depth must be intrinsically different from that at the correct depth; otherwise depth estimation is ambiguous. Indeed, we see in Fig. 4.2(e-h) that pixel values when focused at incorrect depth usually either lie in a higher-dimensional space or have higher variance.

General Diffuse plus Specular

When the image is refocused to the incorrect depth, different views will actually come from different geometric locations on the object. Since our test image is a sphere with a point light, each point has a different value for $\rho_d(v)$ and $(n \cdot l)$. In addition, some of the neighboring points have only the diffuse component (since $\rho_s(v)$ is close to 0). Therefore, the pixel values have a wider span on the convex cone, as shown in Fig. 4.2(e), where fitting a line will result in larger residuals.

General Diffuse

Since each view has a different intensity due to different $(n \cdot l)$ and $\rho_d(v)$, pixel values from different views will usually span a wider section of a line, as shown in Fig. 4.2(f).

Lambertian Diffuse plus Specular

The specular color component at neighboring points will have differing $\rho_s(v)$, similar to the case when focused at the correct depth. However, the variations are larger. The diffuse color component also now varies in intensity because of different $(n \cdot l)$ values at neighboring points.

When focused at the correct depth, the points lie in a line, which we call *line-consistency*. At an incorrectly focused depth, the RGB plot diverges from a line. However, as seen in Fig. 4.2(g), this signal is weak; since the diffuse intensity variation is much less than the specular component, different views still lie almost on a line in RGB space for incorrect depth, but usually with a larger variation than when focused to the correct depth. Therefore, we need to combine point-consistency for Lambertian-only regions. We will use this observation in our practical algorithm.

Lambertian Diffuse

Different views have different values for the $(n \cdot l)$ fall-off term (the sphere in Fig. 4.2 is not textured). However, all the points have the same diffuse color. Thus, pixel values lie on a line passing through the origin, as shown in Fig. 4.2(h). Again, point-consistency is a good photo-consistency measure for Lambertian diffuse surfaces, since it holds when focused at the correct depth and not at incorrect depths.
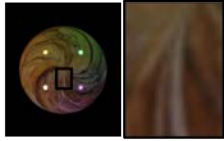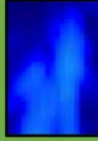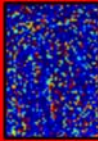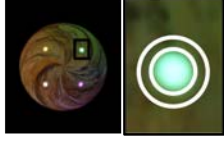
(Light-field Input (Central Viewpoint))	Point-Consistency Depth Estimation (α^p)	Point-Consistency Confidence (C^p)	Line-Consistency Depth Estimation (α^l)	Line-Consistency Confidence (C^l)
Diffuse Edges	+ Stable depth metric	+ Higher confidence	- Noisy, thrown off by texture	- Lower confidence
Specular Edges (Highlighted in White)	- High errors	- High confidence even with incorrect depth	+ Accurate at specular edges	+ High confidence at edges, low confidence otherwise

Figure 4.3: *Point-Consistency versus Line-Consistency. For point-consistency, diffuse edges exhibit high confidence and meaningful depth. However, for specular regions, the error is large with high confidence. With line-consistency, diffuse regions register depth values that are noisy and lower confidence. For specular regions, line-consistency is accurate at the edges with high confidence. For both, it is important to note that the metrics have high confidence where edges are present. Therefore, saturated pixels, often observed in large specular patches, and smooth surfaces require data propagation. Although sat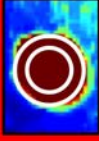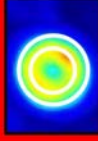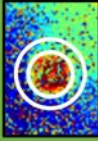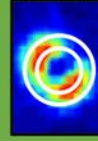urated specular regions still have high errors, we can see that line-consistency has a much lower confidence than the point-consistency metric.*

4.3 Algorithm

We now describe our practical algorithm, which builds on the theoretical model. We assume Lambertian plus specular materials, as in most previous work. The photo-consistency condition can be generalized to *line-consistency*, which estimates a best fit line. This provides a better measure than the *point-consistency* or simple variance in the Lambertian case. We then use a new regularization scheme to combine both photo-consistency measures, as seen in Fig. 4.1. We show how point and line-consistency can be combined to obtain robust depth estimation for glossy surfaces. If we want higher order reflection models, we can use best-fit elements at higher dimensions for the analysis.

We will also assume dielectric materials, where the specular component is the color of the light source, for most of our results, although it is not a limitation of our work, and Eq. 4.7 is general. The assumption is needed only for relating light source color to that of the highlight, and does not affect depth estimation. In Tao et al. [79], we

used an iterative approach of estimating the light source color line direction to generate the specular free and specular images and depth estimation just using Lambertian point-consistency. This iterative approach may lead to convergence problems in some images and artifacts associated with specular removal affect depth results. In this chapter, we introduce our new algorithm that uses both light source color estimation and depth estimation that exploits photo-consistency. We show that this eliminates the need for an iterative approach and achieves higher quality results in Figs. 4.10 and 4.11.

Since we now use the generalized photo-consistency term for specular edges, the new depth-error metric is as follows:

- *For Lambertian only surfaces,* the error metric is the variance across the viewpoints (**point-consistency measure**).[1]

- *For Lambertian plus specular surfaces,* the error metric is the residual of the best fit line, where the slope of the line represents the scene light source chromaticity (**line-consistency measure**).

The depth metrics have strengths and weaknesses, as summarized in Fig. 4.3. For point-consistency, diffuse edges exhibit high confidence and meaningful depth. However, for specular edges, the error is large with high confidence. The high confidence is caused by the fact that the specular regions are much brighter than the neighborhood pixels. Although true point-consistency does not exist, the point-consistency metric between close to point-consistency and otherwise is large among depths. Therefore, incorrect high confidence is common. With line-consistency, the measure is accurate at specular edges with high confidence. But, with the line-consistency measure, the depth estimation for diffuse regions is unstable. Even at incorrect depth (Fig. 4.2(h)), the points lie in a line. The line-consistency measure will register both correct and incorrect depth as favorable due to the low residuals of a best fit line. Texture also will introduce multiple lines since different colors from neighborhood pixels may register new best-fit-lines. Therefore, depth values are noisy and confidence is lower. For both point and line-consistency, it is important to note that more prominent edges yield higher confidence and meaningful depth estimations. Therefore, saturated pixels, often observed in large specular patches, and smooth diffuse surfaces require data propagation.

Our algorithm addresses the following challenges of using the two metrics:

---

[1] In our implementation, we used both the Lambertian photo-consistency and defocus measure from the previous chapter. We then combine the two as a measure by using the depth values with maximum confidence. This provided us cleaner results. For simplicity, we will still call the measure point-consistency.

Algorithm 4 Depth Estimation with Specular Removal

1: $\alpha^p, C^p = \text{PointConsistency}(I)$
2: $L^* = \text{LightSourceColor}(I, \alpha^p)$
3: $\alpha^l, C^l = \text{LineConsistency}(I, L^*)$
4: $Z^* = \text{DepthRegularization}(\alpha^p, C^p, \alpha^l, C^l)$
5: $D, S = \text{SpecularFree}(I, L^*)$

- *Identifying Diffuse Only and Glossy Surfaces.* We need to identify which pixels are part of a diffuse only or glossy surface to determine which depth-error metric better represents each surface or region.

- *Combining the Two Measures.* Point-consistency is a good measure for diffuse surfaces and line-consistency is a good measure for specular surfaces. We need to combine the two measures effectively.

- *Angularly Saturated Pixels.* Because of the small base-line of light-field cameras, at specular regions, surfaces with all view points saturated are common. We mitigate this problem through hole-filling.

Algorithm Overview

Our algorithm is shown in Algorithm 4. The input is the light-field image $I(x, y, u, v)$ with (x, y) spatial pixels and, for each spatial pixel, (u, v) angular pixels (viewpoints). The output of the algorithm is a refined depth, Z^*, and specular S and diffuse D components of the image, where $I = D + S$.

The algorithm consists of five steps:

1. *Point-consistency measure.* We first find a depth estimation using the point-consistency measure for all pixels. For diffuse surfaces, this error metric will give us accurate depth to distinguish between Fig. 4.2(d and h) (line 1).

2. *Estimate light-source color.* For specular surfaces, analyzing the angular pixels (u, v) allows us to estimate the light-source color(s) of the scene and determine which pixels are specular or not. (line 2, Chapter 4.3).

3. *Line-consistency measure.* Given the light-source color, we then find a depth estimation using the line-consistency measure for all pixels. For specular edges, we will then obtain the correct depth to distinguish between Fig. 4.2(c)(g) (line 3).

4. *Depth regularization.* We then regularize by using the depth and confidences computed from steps 1 and 3 (line 4).

5. *Separate specular.* Because we are able to identify the light-source color, we are able to estimate the intensity of the specular term for each pixel. We use this to estimate a specular-free separation (line 5).

Point-consistency Depth Measure [Line 1]

Given the input image $I(x, y, u, v)$, with (x, y) spatial pixels and (u, v) angular pixels, as an initial depth estimation, we use the point-consistency metric that measures the angular (viewpoint) variance for each spatial pixel. We first perform a focus sweep by shearing. As explained by Ng et al. [55], we can remap the light-field input image given the desired depth as follows:

$$
\begin{aligned}
I_\alpha(x, y, u, v) &= I(x', y', u, v) \\
x' &= x + u(1 - \frac{1}{\alpha}) \\
y' &= y + v(1 - \frac{1}{\alpha})
\end{aligned}
\tag{4.9}
$$

where α is proportional to depth. We take $\alpha = 0.2 + 0.007 * Z$ where Z is a number from 1 to 256.

We compute a point-consistency measure for each spatial pixel (x, y) at each depth α by computing the variance across the angular viewpoints, (u, v) as follows,

$$
E_p(x, y, \alpha) = \sigma^2_{(u,v)}(I_\alpha(x, y, u, v))
\tag{4.10}
$$

where $\sigma^2_{(u,v)}$ is the variance measure among (u, v). To find $\alpha^p(x, y)$, we find the α that corresponds to the lowest E_p for each (x, y). The confidence $C^p(x, y)$ of $\alpha^p(x, y)$ is the Peak Ratio analysis of the responses [27]. However, the point-consistency error metric is a poor metric for specular regions because point-consistency cannot be achieved with the viewpoint dependent specular term, as shown in Eq. 4.7 and Figs. 4.2 and 4.3.

Estimating Light Source Chromaticity, L^* [Line 2]

Before we can use the line-consistency depth measure in Line 3, we need to reduce overfitting by finding the light source color from point-consistency depth, and then optimizing the depth for line-consistency with the estimated light source color.

Although point-consistency does not provide us a correct depth measure for specular edges, the small variance in the (u, v) provides us enough information to estimate a line, as shown in Fig. 4.4. At a depth that is far from the point-consistency depth, the viewpoints contain neighboring points with different albedo colors (Fig. 4.4(d)). This throws off light source color estimation. By using the viewpoints from a point-consistency depth, the influence from neighboring points is reduced and we get a line with a slope that is very close to the true light source color (Fig. 4.4(c)).
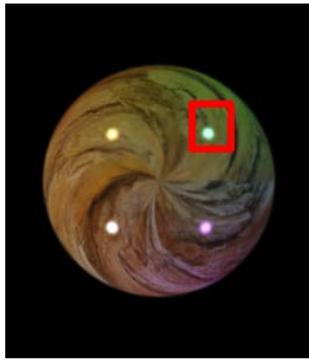
(a) Light-field Input
(Central Viewpoint)
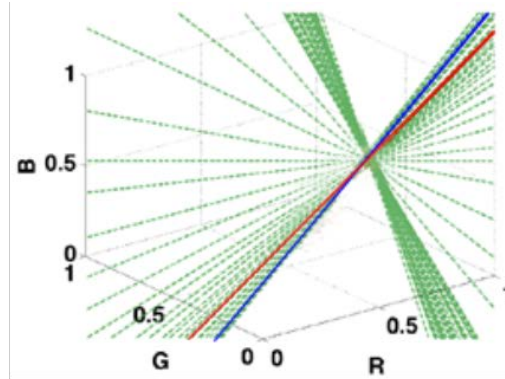
(b) Estimated Lines of Specular Region
Red (Ground Truth)
Blue (Our Estimate)
Greens (All Refocused Estimates)
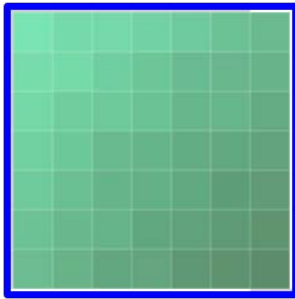
(c) (u,v) in I_d^p
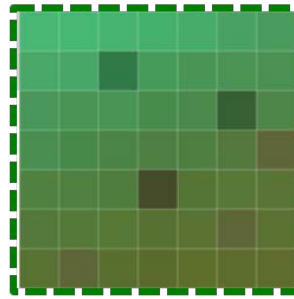
(d) (u,v) from not
point-consistent depth

Figure 4.4: *Line Estimation. With the same input image scene as Fig. 4.1 and sampled point (a), we plot the the angular pixels at the point-consistency depth, $I_{\alpha^P(u,v)}$ (b). By using the angular pixels, we can estimate the light source color (estimated line shown in blue) accurately (ground truth shown in red). With point-consistency, we reduce the influence of colors from neighboring points but still have enough color variation to estimate the light-source color (c). Without using the point-consistency set of angular pixels, we can see that neighborhood pixels from the sphere throw off the line estimations (shown in dotted green lines) (d).*

(a) Glossy cans with two light sources   (b) Light source estimation   (c) Semi-glossy crayons with two light sources   (d) Light source estimation   (e) Light source ground truth

Figure 4.5: *Estimating Multiple Light-Sources. By using our L^* estimation on the scene with two highly glossy cans with two light sources (a), we can see that the estimated RGB values in L^* (b) is consistent with the ground truth (e). The RMSE for the green and red light-sources are 0.063 and 0.106 respectively. Even with semi-glossy crayons (c), the light source estimation is consistent (d). The RMSE for the green and red light-sources are 0.1062 and 0.0495 respectively. We took photos directly of the light sources for ground truth.*

To find the set of angular pixels that represent α^p, we use the following remapping,

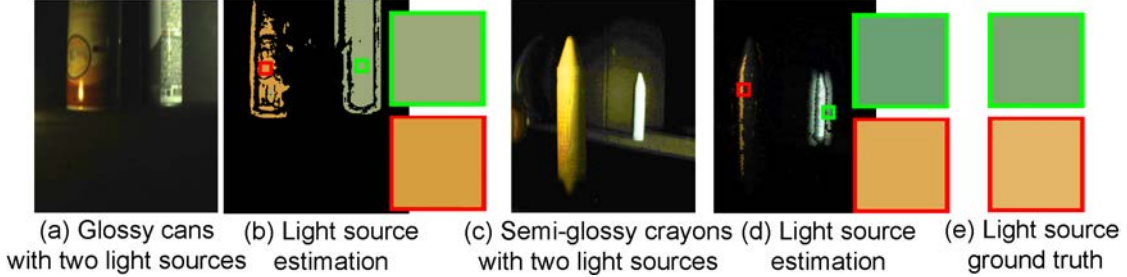$$I_{\alpha^p}(x, y, u, v) = I(x'(\alpha^p(x, y)), y'(\alpha^p(x, y)), u, v). \tag{4.11}$$

We estimate L_i, where i represents each color channel (R,G, or B). For a spatial pixel (x, y), we estimate the slope of the RGB line formed by $I_{\alpha^p}(u, v)$. For each (x, y), we find the direction of the best fit line by using the SVD of the color values across (u, v) for each (x, y). The first column of the right singular vector contains the RGB slope. Since we are interested in just the line direction, we measure the chromaticity, $L_i = L_i/(L_1 + L_2 + L_3)$.

L now gives us the light source chromaticity measure for each spatial pixel, (x, y) in the image. Theoretically, we are now able to estimate N light source colors given N spatial pixels in the image. However, in most cases, such estimation tends to be noisy in real data. We perform k-means clustering to the number of light sources, which is set by the user. For simplicity, we will use L^* as one light-source color. In Fig. 4.5, we show two real-world examples where we have two light sources. In both scenarios, our algorithm estimates L^* that is very similar to the ground truth. In Fig. 4.6, we can see that the four light source colors are estimated from the sphere input.

Line-Consistency Depth Measurement [Line 3]

Given the chromaticity of the light source, $L^*(x, y)$, we can then compute the line-consistency measure. For each α, we have two steps for computing the error metric: first, is to find depths that have angular pixels that observe the same L^* chromaticity and second, is to find the residual of the estimated line.

We compute a light-source similarity metric to prevent other lines, such as the diffuse only line, occlusions, and neighborhood points from influencing our depth measurement. We first compute the estimated light-source color at α to compare against our estimated L^*. To do so, we use the same formulation as in Chapter 4.3, where we used SVD to estimate the line direction. Given the estimated L^*, we compute the measure,

$$E_{\text{Ls Similarity}} = ||L^\alpha(x, y) - L^*(x, y)|| \qquad (4.12)$$

For each (x, y) and α, we then compute the residual of the line defined by L^α, where smaller residuals represent better line fitting:

$$E_{\text{res}} = \sum_{i=(u,v)} r_i^2 \qquad (4.13)$$

where r_i is the residual of each angular pixel in (u, v).

Given the two measures, we can then compute the line-consistency error metric:

$$E_l(x, y) = E_{\text{Ls Similarity}} \cdot E_{\text{res}}. \qquad (4.14)$$

To find $\alpha^l(x, y)$, we find the α that corresponds to the lowest E_l for each (x, y). The confidence $C^l(x, y)$ of $\alpha^l(x, y)$ is the Peak Ratio analysis of the responses.

Depth Regularization [Line 4]

Given the two depth estimations from point-consistency, α^p, and line-consistency, α^l, and their respective confidences, C^p and C^l, we need to combine the two depth mea-
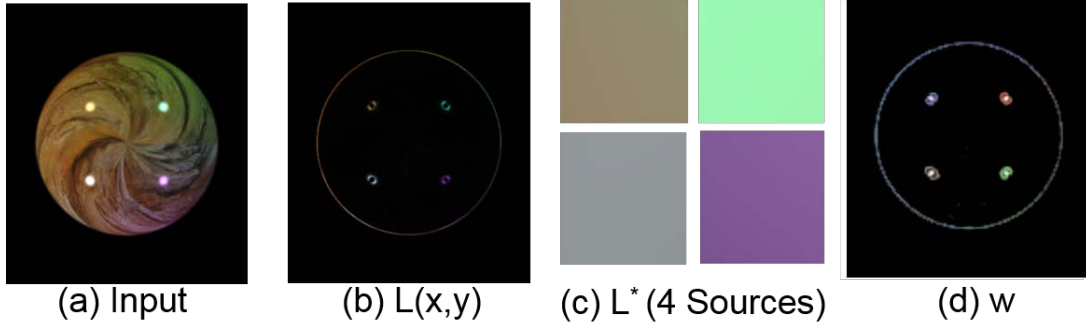
(a) Input   (b) L(x,y)   (c) L*(4 Sources)   (d) w

Figure 4.6: *Light-Source Estimation. With input image (a), we estimate the light-source color, L, for each pixel as shown in (b). We use the k-means clustering method to estimate the light-source colors, L* of the scene (c). The light source colors match the ground-truth, starting from top left to bottom right, with RMSE of 0.0676, 0.0790, 0.0115, and 0.0555. In Chapter 4.3, we show how we measure the specular intensity (d) of each pixel to estimate specular-free images.*

sures. We use these confidences in a Markov Random Field (MRF) propagation step similar to the one proposed by Janoch et al. [35]:

$$
\begin{aligned}
Z^* = \operatorname*{argmin}_{Z} \quad & \lambda_{\mathrm{p}} \sum_i C^p |Z(i) - \alpha^p(i)| \\
& + \lambda_{\mathrm{l}} \sum_i C^l |Z(i) - \alpha^l(i)| \\
& + \lambda_{\mathrm{flat}} \sum_i \left(\left| \frac{\partial Z(i)}{\partial x} \right|_{(x,y)} + \left| \frac{\partial Z(i)}{\partial y} \right|_{(x,y)} \right) \\
& + \lambda_{\mathrm{smooth}} \sum_i |(\Delta Z(i))|_{(x,y)}
\end{aligned}
\tag{4.15}
$$

where $i \in (x, y)$. Given the confidences, we are able to propagate two data terms. The MRF enables us to retain the benefits of both depth measures and mitigate the disadvantages, as shown in Fig. 4.3. C^p is high and C^l is low in diffuse regions, giving us the advantages of the point-consistency measure. However, C^l is high in specular regions, giving us the advantages of the line-consistency measure. After combining the two measures, in Fig. 4.1, we show that depth estimation artifacts from glossy regions are reduced.

In our implementation, we use $\lambda_p = \lambda_l = 1$, $\lambda_{\text{flat}} = 2$, and $\lambda_{\text{smooth}} = 1$.

Estimating Specular Free Image [Line 5]

So far we have light source chromaticity and depth estimation. Separating diffuse and specular components is useful in some applications but not required for depth. To separate the two components, we need to estimate the specular intensity to separate diffuse and specular. From Eq. 4.7, for each (u, v),

$$
\begin{aligned}
I_{Z^*} &= [c \cdot \bar{L}_d(\lambda) + L_i^* \rho_s(\boldsymbol{v})] \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) \\
&= [c \cdot \bar{L}_d(\lambda) \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) + L_i^* \cdot w]
\end{aligned}
\tag{4.16}
$$

where I_{Z^*} is the light-field image mapped to Z^*, L_i^* is the light-source chromaticity and w is the specularity intensity measure dependent on (u, v). The L_i^* takes place of the $\bar{L}_s(\lambda)$ term in Eq. 4.7. The goal is to estimate w, as shown in Fig. 4.6.

Estimating Specular Intensity, w

A straightforward way to estimate specular intensity is to use the fitted-line with L^* and subtract each (u, v) based on their position on the line. However, the results become noisy and introduce artifacts. To alleviate the artifacts, we categorize each (u, v) pixel in I_{Z^*} as diffuse only or diffuse plus specular angular pixels. We used a conservative approach by clustering the pixels on the line into two groups. From Eq. 4.1, for each spatial pixel (x, y), we categorize the pixels as

$$
\begin{aligned}
\langle c \cdot \bar{L}_d(\lambda) \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) \rangle(u, v) &= \min I_{Z^*}(u, v) \\
\langle \bar{L}_s(\lambda) \rho_s(\boldsymbol{v}) \cdot (\boldsymbol{n} \cdot \boldsymbol{l}) \rangle(u, v) &= w(u, v) \cdot L^*
\end{aligned}
\tag{4.17}
$$

where $\langle . \rangle$ denotes the expected value operator. To estimate the specular intensity, we compute w as follows,

$$
w(u, v) = (I_{Z^*}(u, v) - \min I_{Z^*}(u, v))/L^*
\tag{4.18}
$$

In a Lambertian diffuse plus specular case, (u, v) pixels that deviate more from the minimum will have a higher $w(u, v)$. In a diffuse only case, since all the spatial pixels

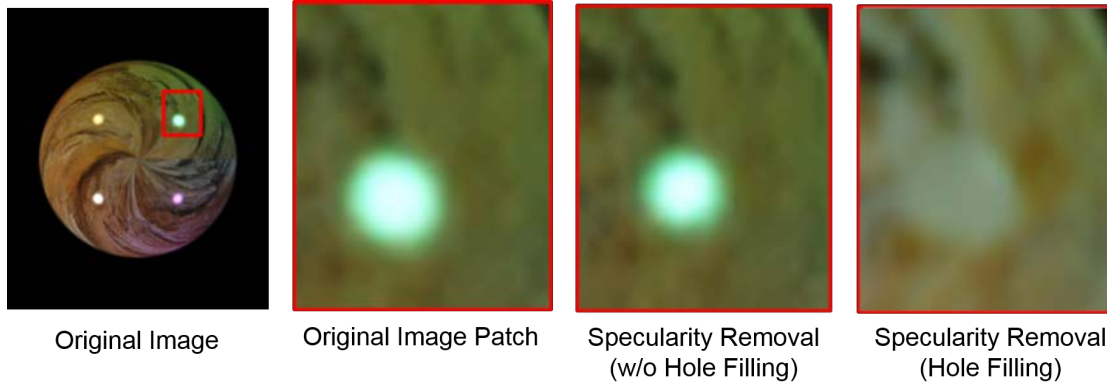| Original Image | Original Image Patch | Specularity Removal (w/o Hole Filling) | Specularity Removal (Hole Filling) |

Figure 4.7: *Specular removal. With just using the angular information, we are able to reduce specularity. However, with large specular regions such as the one from the sphere, the specular removal from angular information can only remove the specular component partially (reducing the size of the specular highlight). Therefore, spatial Poisson reconstruction hole filling is needed to completely remove large saturated specular regions.*

have point-consistency, $w(u, v) = 0$. In Fig. 4.6, we show that our method estimates both the light source colors and the specular intensity.

Removing specularities angularly

We want to average diffuse pixels in (u, v) to replace the specularity pixels, while preserving the diffuse pixels. To remove specularities, we use a weighted average approach by averaging angular pixels (u, v) within the same spatial coordinate (x, y).

$$
\begin{aligned}
D(x, y, u, v) &= \frac{1}{||W||} \sum_{(u,v)} W(x, y, u, v) \cdot I_{Z^*}(x, y, u, v) \\
W(x, y, u, v) &= 1 - w(x, y, u, v) \\
S(x, y, u, v) &= I_{Z^*}(x, y, u, v) - D(x, y, u, v)
\end{aligned}
\tag{4.19}
$$

where D is diffuse and S is specular.

Hole Filling: Removing specularities angularly only works for local estimation (edges of specular and diffuse regions). This method does not support angularly

saturated pixels, where change in light-field viewpoints is ineffective towards distinguishing pixels with both terms or just the diffuse term. Since the baseline of a light-field camera is small, angularly saturated specular terms happen often. Therefore, to remove specularities entirely, we used simple hole filling methods, as shown in Fig. 4.7.

In our implementation, we used a Poisson reconstruction method, proposed by Perez et al. [60]. We seek to construct a diffuse only image in Eq. 4.20. The gradient of the final diffuse image is the following:

$$\nabla D(x, y, u, v) = (1 - w(x, y, u, v)) \cdot \nabla I(x, y, u, v). \qquad (4.20)$$

4.4 Results

We verified our results with synthetic images, where we have ground truth for the light source, and diffuse and specular components. For all real images in the chapter, we used both the Lytro classic and Illum cameras. We tested the algorithms across images with multiple camera parameters, such as exposure, ISO, and focal length, and in controlled and natural scenes.

Quantitative validation

We use PBRT [62] to synthesize a redwood textured glossy sphere with specular reflectance $K_s = [1, 1, 1]$ and roughness 0.001 and four different colored light sources. In Fig. 4.8, we added Gaussian noise to the input image with mean of 0 and variance between 0 and 0.02. Our depth RMSE shows significant improvement over Tao et al. [79]. We can see that the other methods are prone to both noise, especially Wanner et al. [83] and glossy surfaces. For the diffuse RMSE, we can see that although noise does affect the robustness of our separation result, we still outperform previous work. The quantitative validation is reflected by the qualitative results, where we see that both depth and diffuse and specularity separation is robust across noise levels, even at high noise variance, 0.02.

In Figs. 4.5 and 4.6, we computed the RMSE against the ground truth light source colors. In both real world scenes with the glossy cans and semi-glossy crayons, the light-source estimation exhibits low RMSE. The RMSE for the green and red light-sources with the glossy cans are 0.063 and 0.106 respectively. The RMSE for the green and red light sources with the semi-glossy crayons are 0.1062 and 0.0495. We computed the difference between our estimated light source color and the ground truth synthetic
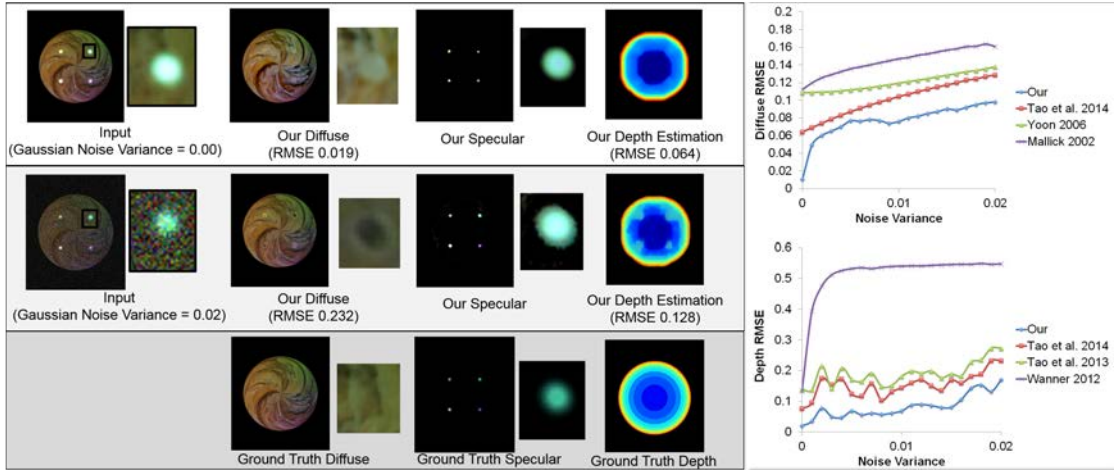
Figure 4.8: *Qualitative and Quantitative Synthetic Results. We added Gaussian noise with zero mean and the variance as the variable parameter to the input image of Fig. 4.1. We compute the RMSE of our results against the ground truth diffuse image and depth map. On the left, even with high noise, we can see that our diffuse and specular separation closely resembles the ground truth. In both cases, the algorithm is able to extract all four specular regions. For depth maps, we can see that the depth estimation in the presence of high noise still reasonably resembles the ground truth sphere. On the right, we can see that these qualitative results reflect the quantitative result. We see that our results outperform prior works by a significant margin.*

image in Fig. 4.6. The four estimated light source colors match the ground-truth, starting from the top left to bottom right, with RMSE of 0.0676, 0.0790, 0.0115, and 0.0555.

In Fig. 4.9, we have a flat glossy surface that is perpendicular to the camera. The ground truth depth is flat. With our method, the depth estimation resembles the ground truth with an RMSE of 0.0318. With the line-consistency measure, we can see that diffuse areas cause unevenness in the depth estimation with an RMSE of 0.0478. With the point-consistency measure, because of the specularities, we can see strange patterns forming along the specularities with an RMSE of 0.126. This result is similar to the Lytro depth estimation, where the RMSE is also high at 0.107.

Depth Map Comparisons

We show our depth estimation result in Figs. 4.10, 4.11, and 4.12. To qualitatively assess our depth estimation, we compare our work against Lytro software, Tao et al. (13,14) [80, 79], and Wanner et al. [83]. We tested our algorithm through multiple
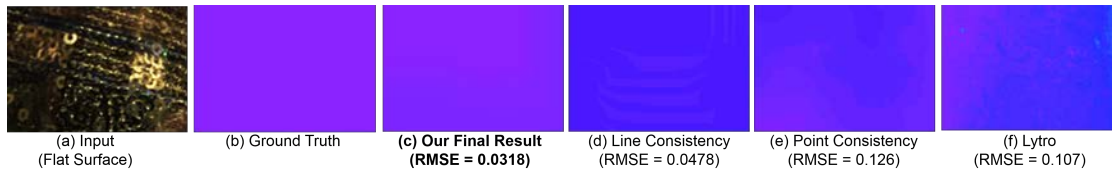
| (a) Input (Flat Surface) | (b) Ground Truth | (c) Our Final Result (RMSE = 0.0318) | (d) Line Consistency (RMSE = 0.0478) | (e) Point Consistency (RMSE = 0.126) | (f) Lytro (RMSE = 0.107) |

Figure 4.9: *Flat Glossy Surface Results. We have a completely flat glossy surface that we placed directly perpendicular to the camera (a). For the ground truth, the depth should be flat (b). We can see that our final result is also smooth and flat (c). The line-consistency provides the smoothness, but has some errors in the non-glossy regions (d). The point-consistency is thrown off by some of the glossy regions of the image (e). With the Lytro's depth estimation, we also see that the specular regions throw off the depth estimation (f).*

scenarios involving specular highlights and reflections.

In Fig. 4.10, we show three diverse examples of typical glossy surfaces. On the top, we have a smooth cat figurine with generally small glossy speckles. The paw is the most noticeable feature where the specularity affects depth estimations that assume Lambertian surfaces. Our depth estimation preserves the details of the glossy paw, whereas the other methods show strange paw shapes. In the princess example, we have several area light sources which produce large glossy highlights. We can see our depth result does not contain erroneous depth registrations at these specular regions, especially at the bow. In the Lytro depth estimation, we can see that the large patches of specularities affect the result on the bow and face. We also can resolve the contours of the dress and that the left arm is behind the body. Lytro and the previous works fail to resolve the change in depth, misrepresenting the glossy figurine. We observe similar results with the Chip and Dale figurine with multiple colors. Our depth result is not thrown off by the specular regions and is able to recover the shape of the figurine (as shown in the feet on the right). Other methods show incorrect depths for the large specular regions. In the Lytro depth estimation, we can see large patches of depth errors on the face.

In Fig. 4.11, we show more difficult examples of shooting through glare on glass. We can see that in both examples, we are able to recover clean depth results, whereas the other algorithms exhibit spikes and errors throughout the image. In the mouse example, our method is able to estimate the outline of the mouse without the glare affecting regularization results. We can see all previous results have non-plausible depth estimations. In the figurine of the couple, we observe the same result. Notice on

the left side of the image where there are bright glare and reflections. In previous works' and Lytro's depth estimation, large patches of errors exist in the specular regions.
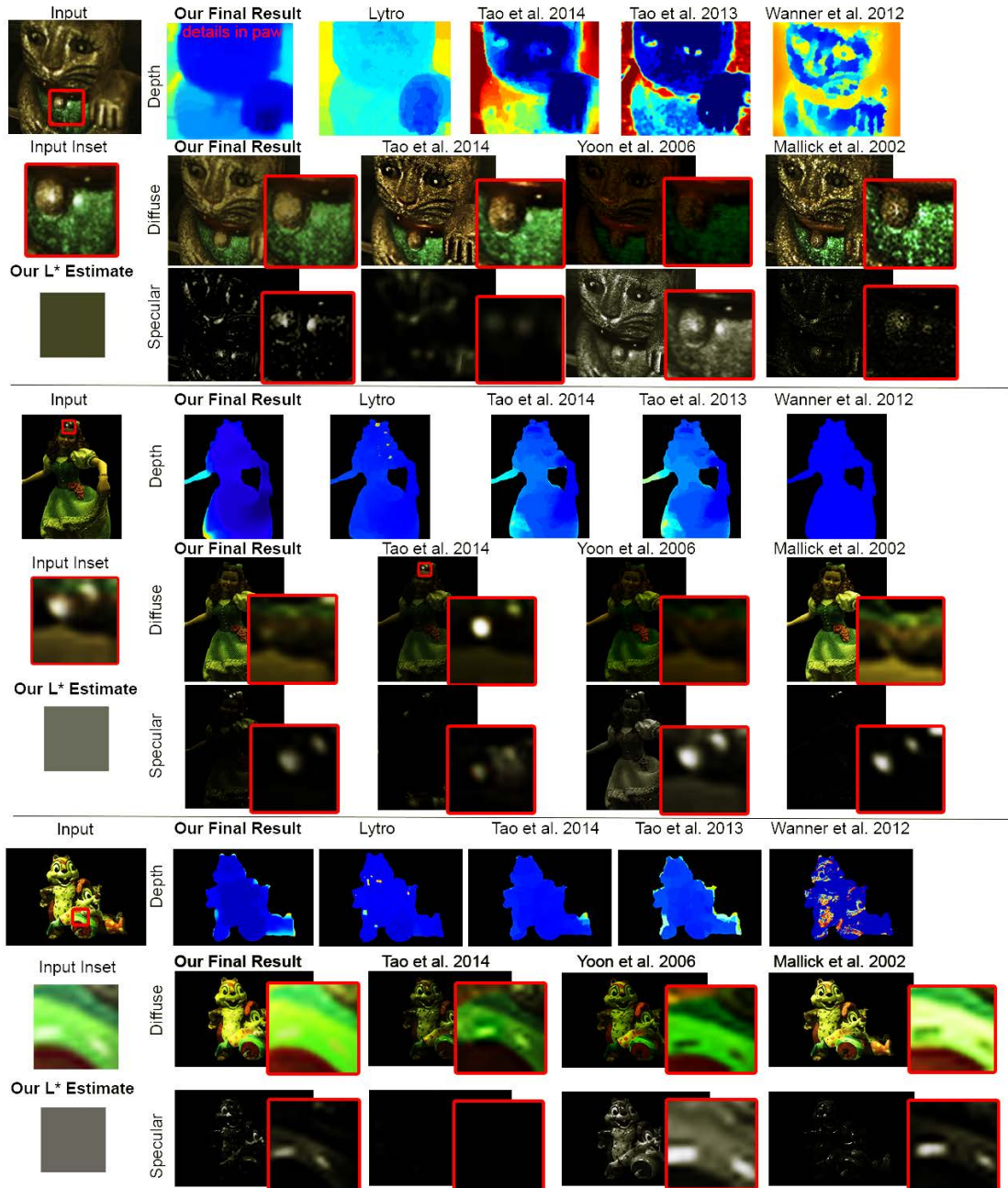
Figure 4.10: *Our Results. We compare depth results against Lytro software, Tao et al. 14 [79], Tao et al. 13 [80], and Wanner et al. [83]; and specular removal results against Tao et al. 14 [79], Yoon et al. [92], and Mallick et al. [50]. On the top, we have a smooth cat with texture and speckles. Our final depth has plausible paw details. In the second example, we have a princess figurine. Our depth estimation exhibits fewer errors at the large patches of specularities. With the Chip and Dale, we show that our depth result resembles the shape of the figurine.*

Figure 4.11: *Scenes with Glare. We have a different scenario where we took photos through a window glass, where glare becomes a prominent problem (highlighted in green). Our algorithm is robust against glare from the glass, while regularization from other algorithms propagates inconsistent results. We also see that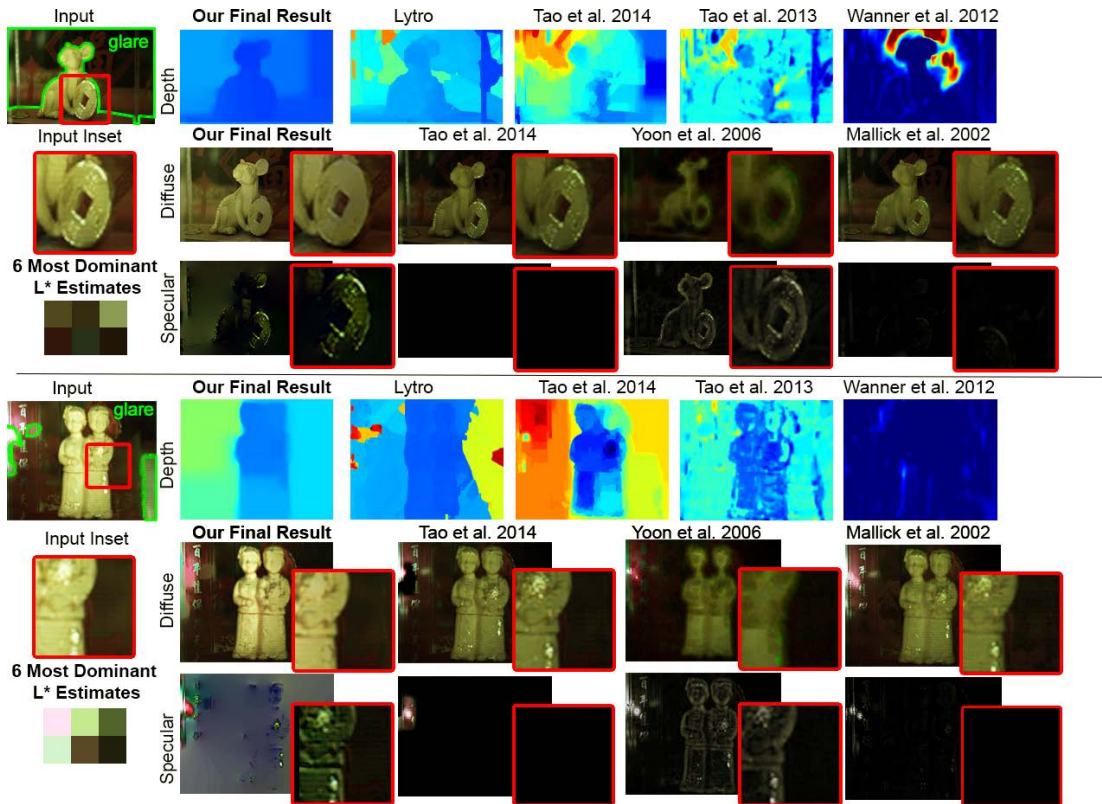 our algorithm removes specularities on the figurines while reducing the glare on the glass (although, the algorithm does not completely remove the glare), while Mallick et al. and Yoon et al. struggle due to multiple colors associated with the glass. The results are made possible because we are able to estimate multiple light-source colors, up to the number of spatial pixels in the image; whereas traditional specular removal algorithms can only remove a small set number of light source colors, not suitable for glare cases. In both examples, we show the six most prominent L^* estimates. The estimation closely resembles the glare from the window. Because we are using a gradient integration for hole filling, bleeding effects may appear.*
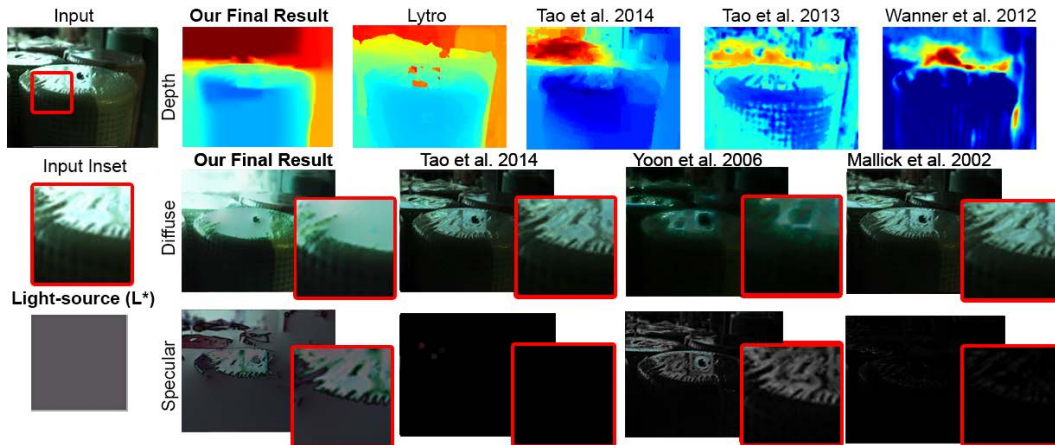
Figure 4.12: *Limitations. Here is an image of plastic wrapped yoga mats with a glass
window in the background. There are large specular regions in the background and also
on top of the closer yoga mat. Although our depth estimation is more correct compared
to other methods, the specular-free image exhibits a huge smooth patch removed from
the glass and the yoga mat. This is because there is not enough information behind the
specular reflection.*

Specular-free Image Comparisons

We compare our specular and diffuse separation against the ground truth in Fig. 4.8.
We also show that our results accurately estimate multiple light sources of real scenes in
Fig. 4.5. We compare our specular removal result against Tao et al. [79], Yoon et al. [92]
Mallick et al. [50] in Figs. 4.10, 4.11, and 4.12. With one light-source color examples of
Fig. 4.10, we can see that our specularity removal accurately removes specular regions.
In both the small speckle glossy regions (cat) and large specular regions (princess and
Chip and Dale) examples, Mallick et al. fail to remove many parts of the small speckles,
Yoon et al. incorrectly remove most of the image colors, and Tao et al. observe strange
artifacts and struggle with large specularities. Both Yoon et al. and Mallick et al. incor-
rectly estimate the light source color as [111], which becomes problematic with scenes
with non-white light source colors. The results are very distinguishable with the exam-
ples that were shot through glass in Fig. 4.11. We are able to mitigate the glare from
the glass and remove the specularities from the figurines (coin from the mouse and the
reflective speckles on the couple). In both examples, we can see that our most promi-
nent L^* estimations resemble the glare observed through the window. Even though we
cannot remove large patches of specularities such as on the Yoga mats in Fig. 4.12, we
generate reasonable results that can be fixed through better hole-filling.

Limitations and Discussion

Although glossy edges should give different pixel values for different views while diffuse edges do not, it is still hard to separate them practically because of the small-baseline nature of light-field cameras as well as the noise. Second, saturated highlights cannot be distinguished from a diffuse surface with large albedo value. In addition, the specular components of saturated specular regions cannot be completely removed. However, our confidence measure for specular regions and specular removal help alleviate those effects. In some cases, especially scenes with large specular patches or saturated color values, the specularity-removal is not able to recover the actual texture behind the specular regions. We show this with an example of a glossy plastic wrapping around a yoga mat (Fig. 4.12). The diffuse output is flat. However, this does not affect the quality of our depth result, which still outperforms previous methods. Future work includes supporting more complex BRDF models and better hole filling techniques.

4.5 Conclusion

In this chapter, we first investigate the characteristics of pixel values from different view points in color space for different BRDFs. We then present a novel and practical approach that uses light-field data to estimate light color and separate specular regions. We introduced a new depth metric that is robust for specular edges and show how we can combine the traditional point-consistency and the new line-consistency metrics to robustly estimate depth and light source color for complex real world glossy scenes.

Chapter 5

Conclusion

In this thesis, we have demonstrated how to decode the image data from the Lytro cameras and use the following four depth cues: defocus, correspondence, shading, and specularity. By using the four cues, we introduce a robust and easy way for an average user to capture and acquire depth information from a consumer camera. With the defocus and correspondence cues using angular coherence and specularity, we demonstrated a method that enables users to capture planar depth estimation of scenes that is suitable for multiple scenarios and camera settings. With the shading cue, we demonstrated a method that enables users to capture shape information, suitable for applications such as low resolution 3D printing.

For decoding, we show how we reverse engineered the encoder, color calibration, and micro-lens calibration, all of which is needed for our depth analysis. For defocus and correspondence, we show both an intuitive contrast-based method and a more robust method for general scenes. We have proposed and provided quantitative validation for a new shape estimation framework that uses just a single-capture passive light-field image. Our optimization framework can be used for consumer grade light-field images to incorporate all three cues: defocus, correspondence, and shading. For glossy scenes, we show a principled new line-consistency metric for specular regions of the scene to reduce depth errors introduced by specularities. We investigate the characteristics of pixel values from different view points in color space for different BRDFs. We then present a novel and practical approach that uses light-field data to estimate light color and separate specular regions. We introduced a new depth metric that is robust for specular edges and show how we can combine the traditional point-consistency and the new line-consistency metrics to robustly estimate depth and light source color for complex real world glossy scenes.

Prior to this work, these applications require difficult set-ups or capturing techniques. Our new depth pipeline for depth estimation on light-field cameras takes average users' needs into account and simplifies the process. Because of the robustness and low user involvement, incorporating the cues is a gateway for users to easily acquire and utilize many computer vision and graphics applications. These applications including fast matting, 3D printing, changing depth-of-field, and many other applications that were difficult with traditional 2D images.

5.1 Application Examples

We show that our algorithm produces high quality depth maps that can be used for depth-of-field manipulation, matting and selection, and surface reconstruction.

Depth-of-Field Modifying depth-of-field has been a topic of significant interest with light-field data and cannot be achieved with current commercial software, which can only perform refocusing. Using our depth estimation, we simulate both lens aperture and refocusing (Fig. 5.1 Top). We use the depth map and a user input desired focus plane depth value. Regions with depth values farther from the input depth will have larger blurs. In the figure, we can see that the flowers and background foliage are blurred naturally.

Matting and Selection Current matting and selection graph-cut methods use only color information. Instead of using RGB, we use RGBD, where D is our depth estimation. With just a simple stroke, we can select out objects of similar colors, where previous color techniques fail (Fig. 5.1 Middle).

Surface Reconstruction One common use of depth-maps is to reconstruct surfaces, which goes beyond the limited parallax shift in Lytro's software. We remap the pixels with respect to our depth-map Z buffer into 3D space with mesh interpolation (Fig. 5.1 Bottom). This enables the users to explore surface shapes and bumps. Our results show that the perspective can be changed drastically and realistically. As shown in Chapter 3, we have printed three examples. The current Lytro Illum is ideal for low resolution prints, given the limited spatial resolution (430x539).
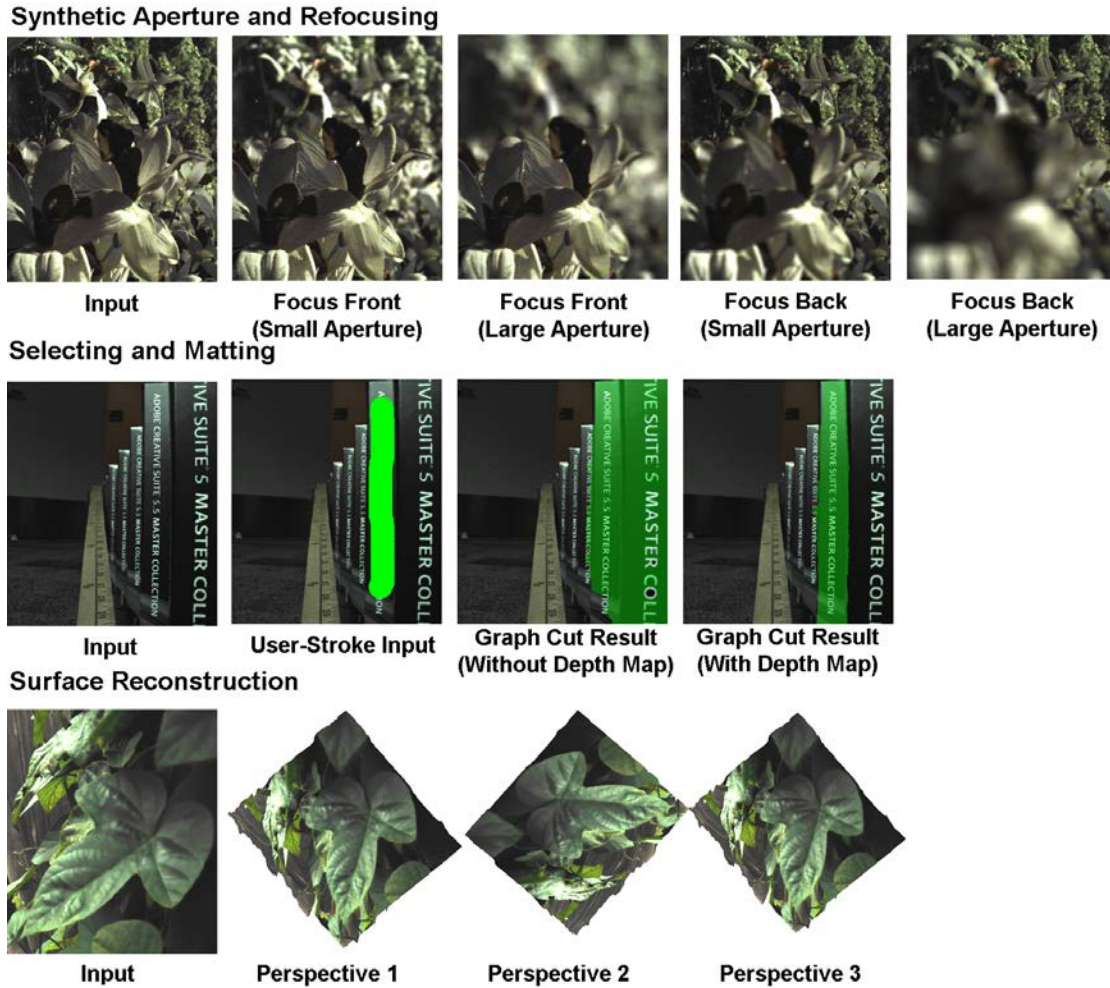
Figure 5.1: *Applications. With our extracted depth maps, synthetic adjustment of both depth of field and refocusing is possible (top). For selection and matting, objects with similar color but different depths can be selected with depth information (middle). By using the depth map as the z-buffer, we can change perspective of the image, producing a 3D look (bottom).*

5.2 Future Work

Optimization of the techniques will be a priority to provide real-time depth estimation experience to users. Although our techniques for defocus and correspondence run at 12 seconds per light-field image, further optimization by, for example, implementing the algorithms on graphic processing units (GPU), may reduce computation time significantly. Optimization could be more effective for estimating shading and specularity, as the algorithms take about 20 minutes to compute a result. The algorithms can be easily parallelized due to the minimal dependencies among subroutines. Moreover, most of the code is in MATLAB and significant speed-ups can be obtained simply by porting to C/C++. We can also work on reducing memory footprint to improve performance in smaller devices.

To improve depth estimations, as noted in Chapter 2, one area to examine further is our decoder. With better calibration of both color and center of micro-lenses, we may achieve higher quality depth estimation and shading/specular separation results. We could also investigate pre-processing techniques such as higher quality denoising or demosaicing techniques to improve input sharpness and details.

At a user-standpoint, an outstanding limitation is that the user needs to understand when the depth estimation works or does not work. For example, when the objects in the scenes are far away from the camera, the light-field cameras resolve depth very poorly. Investigating ways to automatically guide or educate the users on taking images that perform well with our depth estimation is pertinent for making our algorithm suitable for every-day users.

Applying the same technique to active stereo systems such as the Kinect will improve depth estimation results from the devices. For future studies, we can transfer the same insights about the four cues and investigate their limitations in systems that are not light-field based. In combination with active systems such as Kinect or laser scanners, light-field cameras may improve results significantly by taking advantage of each system's strengths and weaknesses. Such advantages from the active systems include more precise and denser depth acquisition than passive systems like the Lytro Illum. However, the dense angular information provided by Lytro Illum can provide refocusing and small perspective viewpoint cues. We can also combine these techniques with other passive systems such as a high resolution camera. With higher resolution cameras and the depth benefits from the light-field cameras, we may reduce the effects of noise and exploit the higher spatial resolution.

Light-field videos or image sequences have been of interest in the next gen-

eration of light-field cameras. Combining the cues for videos, such as temporally consistent depth estimation throughout frames and registration of depth between video frames, can improve results. For example, stitching multiple light-field images may assist in 3D surface reconstruction. By using image registration among different captures, stitching both 2D image information and depth may improve current shape acquisition techniques.

Finally, applications in other fields such as medial image processing for disease diagnosis in neuroimaging and cardiovascular imaging may benefit from the extended depth of field and depth acquisition. The benefit may include reducing x-ray sample CT scans, improving camera systems in surgical processes, and improving visibility through depth and view-point changes.

Bibliography

[1] E.H. Adelson and J.Y.A. Wang. "Single lens stereo with a plenoptic camera". In: *PAMI* (1992).

[2] A. Artusi, F. Banterle, and D. Chetverikov. "A survey of specularity removal methods". In: *Computer Graphics Forum* (2011).

[3] R. Bajscy, S. W. Lee, and A. Leonardis. "Detection of diffuse and specular interface reflections and inter-reflections by color image segmentation". In: *IJCV* (1996).

[4] J. Barron and J. Malik. "Color constancy, intrinsic images, and shape estimation". In: *ECCV*. 2012.

[5] J. Barron and J. Malik. "Intrinsic scene properties form a single rgb-d image". In: *CVPR*. 2013.

[6] J. Barron and J. Malik. "Shape, albedo, and illumination from a single image of an unknown object". In: *CVPR*. 2012.

[7] R. Basri and D. Jacobs. "Lambertian reflectance and linear subspaces". In: *PAMI* (2003).

[8] R. Basri and D. Jacobs. "Photometric stereo with general, unknown lighting". In: *CVPR*. 2001.

[9] A. Bermano et al. "Facial performance enhancement using dynamic shape space analysis". In: *ACM Trans. on Graph.* (2014).

[10] R. Bolles, H. Baker, and D. Marimont. "Epipolar-plane image analysis: an approach to determining structure from motion". In: *IJCV* (1997).

[11] M. K. Chadraker. "What camera motion reveals about shape with unknown BRDF". In: *CVPR*. 2014.

[12] C. Chen et al. "Light field stereo matching using bilateral statistics of surface cameras". In: *CVPR* (2014).

[13] Q. Chen and V. Koltun. "A simple model for intrinsic image decomposition with depth cues". In: *ICCV*. 2013.

[14] Y. Chen and G. Medioni. "Object modeling by registration of multiple range images". In: *Image Vision Computing* (1999).

[15] A. Criminisi et al. "Extracting layers and analyzing their specular properties using epipolar-plane-image analysis". In: *CVIU* (2005).

[16] J. Davis, R. Ramamoorthi, and S. Rusinkiewicz. "Spacetime stereo: a unifying framework for depth triangulation". In: *CVPR*. 2003.

[17] P. Debevec. "Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography". In: *SIGGRAPH*. 2012.

[18] A. van Doorn, J. Koenderink, and J. Wagemans. "Lightfield and shape from shading". In: *Journal of Vision* (2011).

[19] K.-D. Durou, M. Falcone, and M. Sagona. "Numerical methods for shape-from-shading: A new survey with benchmarks". In: *Computer Vision and Image Understanding* (2008).

[20] S. Fanello et al. "Learning to be a depth camera for close-range human capture and interaction". In: *ACM Trans. Graph.* (2014).

[21] G. Finlayson and G.. Schaefer. "Solving for color constancy using a constrained dichromatic reflection model". In: *IJCV* (2002).

[22] B. Freedman et al. "Depth mapping using projected patterns". In: *US Patent* (2009).

[23] B. Goldleucke and S. Wanner. "The variational structure of disparity and regularization of 4d light fields". In: *CVPR*. 2013.

[24] B. Goldluecke and S. Wanner. "The variational structure of disparity and regularization of 4D light fields". In: *CVPR* (2013).

[25] S. Hasinoff and K. Kutulakos. "Confocal Stereo". In: *ECCV* (2006).

[26] C. Hernandez, G. Vogiatzis, and R. Cipolla. "Multipleview photometric stereo". In: *PAMI* (2008).

[27] H. Hirschmuller, P. Innocent, and J. Garibaldi. "Real-time correlation-based stereo vision with reduced border errors". In: *IJCV* (2002).

[28] B. K. P. Horn. "Shape from shading; a method for obtaining the shape of a smooth opaque object form one view". In: *Ph.D. thesis, Massachusetts Institute of Technology* (1970).

[29] B.K.P. Horn and B.G. Schunck. "Determining optical flow". In: *Artificial Intelligence* (1981).

[30] I. Howard. "Perceiving in Depth". In: *New York: Oxford University Press* (2012).

[31] http://optics.moloush.net. *The File Format*. Blog. 2011.

[32] http://www.digitaltrends.com. *Lytro gets serious with the $1599 ileum light-field camera, full specs announced*. Press Release. 2014.

[33] http://www.lytro.com. *Lytro redefines photography with light field cameras*. Press Release. 2011.

[34] X. Hu and P. Mordohai. "A quantitative evaluation of confidence measures for stereo vision". In: *PAMI* (2012).

[35] A. Janoch et al. "A category-level 3D object dataset: putting the kinect to work". In: *ICCV*. 2011.

[36] H. Jin, S. Soatto, and A. J. Yezzi. "Multi-view stereo beyond Lambert". In: *CVPR*. 2003.

[37] M. Johnson and E. Adelson. "Shape estimation in natural illumination". In: *CVPR*. 2011.

[38] M. Kazhdan, M. Bolitho, and H. Hoppe. "Poisson surface reconstruction." In: *Eurographics*. 2006.

[39] C. Kim et al. "Scene reconstruction from high spatio-angular resolution light fields". In: *SIGGRAPH*. 2013.

[40] H. Kim et al. "Specular reflection separation using dark channel prior". In: *CVPR*. 2013.

[41] W.N. Klarquist, W.S. Geisler, and A.C. Brovic. "Maximum-likelihood depth-from-defocus for active vision". In: *Inter. Conf. Intell. Robots and Systems*. 1995.

[42] K.M. Lee and C.-C.J. Kuo. "Shape from shading with a linear triangular element surface model". In: *IEEE PAMI* (1993).

[43] A. Levin. "Analyzing depth form coded aperture sets". In: *ECCV*. 2010.

[44] M. Levoy and P. Hanrahan. "Light Field Rendering". In: *ACM SIGGRAPH*. 1996.

[45] J. Li et al. "Bundled depth-map merging for multi-view stereo". In: *CVPR*. 2010.

[46] C.K. Liang et al. "Programmable aperture photography: multiplexed light field acquisition". In: *ACM SIGGRAPH*. 2008.

[47] S. Lin et al. "Diffuse-Specular Separation and Depth Recovery". In: *ECCV*. 2002.

[48] B.D. Lucas and T. Kanade. "An iterative image registration technique with an application to stereo vision". In: *Imaging Understanding Workshop*. 1981.

[49] A. Majumder. "Perceiving Depth and Size." In: *UCI Lecture Notes* (2004).

[50] S. P. Mallick et al. "Beyond Lambert: reconstructing specular surfaces using color". In: *CVPR*. 2005.

[51] S. P. Mallick et al. "Specularity removal in images and videos: a PDE approach". In: *ECCV*. 2006.

[52] G. Mather. "Image blur as a pictorial depth cue". In: *Biological Sciences*. 1996.

[53] D. Min, J. Lu, and M. Do. "Joint histogram based cost aggregation for stereo matching". In: *PAMI* (2013).

[54] D. Nehab et al. "Color constancy, intrinsic images, and shape estimation". In: *ECCV*. 2012.

[55] R. Ng et al. "Light field photographhy with a hand-held plenoptic camera". In: *CSTR 2005-02* (2005).

[56] K. Nishino, Z. Zhang, and K. Ikeuchi. "Determining reflectance parameters and illumination distribution from a sparse set of images for view-dependent image synthesis". In: *ICCV* (2001).

[57] M. Okutomi and T. Kanade. "A multiple-baseline stereo". In: *PAMI* (1993).

[58] J. B. Park. "Efficient color representation for image segmentation under nonwhite illumination". In: *SPIE* (2003).

[59] A. P. Pentland. "A new sense for depth of field". In: *PAMI* (1987).

[60] P. Peréz, M. Gangnet, and A. Blake. "Poisson image compositing." In: *ACM SIG-GRAPH*. 2003.

[61] C. Perwass and P. Wietzke. "Single lens 3D-camera with extended depth-of-field". In: *SPIE Elect. Imaging*. 2012.

[62] M. Pharr and G. Humphreys. "Physically-based rendering: from theory to implementation". In: *Elsevier Science and Technology Books* (2004).

[63] R. Ramamoorthi and P. Hanrahan. "A signal processing framework for inverse rendering". In: *ACM Trans. on Graph.* (2001).

[64] M. Rohr. "Die Binokularen Instrumente". In: *Berlin: Julius Springer* (1907).

[65] N. Sabater et al. "Accurate disparity estimation for plenoptic images". In: *ECCV LF4CV*. 2014.

[66] Y. Sato and K. Ikeuchi. "Temporal-color space analysis of reflection". In: *JOSAA* (1994).

[67] D. Scharstein and R. Szeliski. "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms". In: *IJCV* (2002).

[68] Y. Schechner and N. Kiryati. "Depth from defocus vs. stereo" how different really are they?" In: *IJCV* (2000).

[69] S. Seitz and C. Dyer. "Photorealistic scene reconstruction by vocal coloring". In: *IJCV* (1999).

[70] S. M. Seitz and C. R. Dyer. "Photorealistic scene reconstruction by voxel coloring". In: *CVPR* (1997).

[71] S. Shafer. "Using color to separate reflection components". In: *Color research and applications* (1985).

[72] R. Sousa, E. Brenner, and J. Smeets. "Judging an unfamiliar object's distance from its retinal image size". In: *Journal of Vision* (2011).

[73] M. Subbarao and G. Surya. "Depth from defocus: a spatial domain approach". In: *IJCV* (1994).

[74] M. Subbarao, T. Yuan, and J.K. Tyan. "Integration of defocus and focus analysis with stereo for 3D shape recovery". In: *SPIE Three Dimensional Imaging and Laser-Based Systems for Metrology and Inspection III* (1998).

[75] S. Suwajanakorn, I. Kemelmacher-Shlizerman, and S. Seitz. "Total moving face reconstruction". In: *ECCV*. 2014.

[76] R. Tan and K. Ikeuchi. "Separating reflection components of textured surfaces using a single image". In: *PAMI* (2005).

[77] R. Tan, K. Nishino, and K. Ikeuchi. "Color constancy through inverse-intensity chromaticity space". In: *JOSA* (2004).

[78] R. Tan, L. Quan, and S. Lin. "Separation of highlight reflections fromÊtextured surfaces". In: *CVPR* (2006).

[79] M. Tao et al. "Depth estimation for glossy surfaces with light-field cameras". In: *ECCV LF4CV*. 2014.

[80] M. W. Tao et al. "Depth from combining defocus and correspondence using light-field cameras". In: *ICCV*. 2013.

[81] V. Vaish et al. "Reconstructing occluded surfaces using synthetic apertures: stereo, focus and robust measures". In: *CVPR*. 2006.

[82] H. Wallach and C. Zuckerman. "The constancy of stereoscopic depth". In: *Am. J. Psychology* (1963).

[83] S. Wanner and B. Goldluecke. "Globally consistent depth labeling of 4D light fields". In: *CVPR*. 2012.

[84] S. Wanner and B. Goldluecke. "Variational light field analysis for disparity estimation and super-resolution". In: *PAMI* (2014).

[85] M. Wantanabe and S.K. Nayar. "Rational filters for passive depth from defocus". In: *IJCV* (1998).

[86] A. Welchman et al. "3D shape perception from combined depth cues in human visual cortex". In: *Nature Neuroscience* (2005).

[87] R. J. Woodham. "Gradient and curvature from photometric stereo including local confidence estimation". In: *Journal of the Optical Society of America* (1994).

[88] R. J. Woodham. "Photometric method for determining surface orientation from multiple images". In: *Optical Engineering* (1980).

[89] C. Wu et al. "Real-time shading-based refinement for consumer depth cameras". In: *SIGGRAPH Asia*. 2014.

[90] Y. Xiong and S.A. Shafer. "Depth from focusing and defocusing". In: *CVPR*. 1993.

[91] Q. Yang, S. Wang, and N. Ahuja. "Real-time specular highlight removal using bilateral filtering". In: *ECCV*. 2010.

[92] K,-J. Yoon, Y. Choi, and I. S. Kweon. "Fast separation of reflection components using a specularity-invariant image representation". In: *IEEE Image Processing*. 2006.

[93] Z. Zhang et al. "Shape from shading: A survey". In: *PAMI* (1999).

[94] Q. Zhao et al. "A closed-form solution to retinex with non-local texture constraints". In: *PAMI* (2002).