COMBINING LEARNING AND COMPUTATIONAL IMAGING FOR 3D INFERENCE

by

Xinqing Guo

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

Fall 2017

COMBINING LEARNING AND COMPUTATIONAL IMAGING FOR 3D INFERENCE

by

Xinqing Guo

Approved: ______________________________

Kathleen McCoy, Ph.D.
Chair of the Department of Computer and Information Sciences

Approved: ______________________________

Babatunde A. Ogunnaike, Ph.D.
Dean of the College of Engineering

Approved: ______________________________

Ann L. Ardis, Ph.D.
Senior Vice Provost for Graduate and Professional Education

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ______________________________

Jingyi Yu, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ______________________________

Chandra Kambhamettu, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ______________________________

Christopher Rasmussen, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ______________________________

Li Liao, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: ______________________________

Liu Ren, Ph.D.
Member of dissertation committee

ACKNOWLEDGEMENTS

This dissertation could not have been completed without the great support that I have received from so many people over the years. I wish to offer my most heartfelt thanks to the following people.

First and foremost, I thank my academic advisor, Prof. Jingyi Yu for his wisdom, advice and support that allowed me to pursue research on topics for which I am truly passionate. He made my Ph.D. study such a rewarding journey by engaging me in new ideas and demanding a high quality of work in my endeavors. Additionally, my deep gratitude also goes to my other committee members, Prof. Chandra Kambhamettu, Prof. Christopher Rasmussen, Prof. Li Liao and Dr. Liu Ren for their insightful comments and valuable advice.

I would like to thank my mentors during my internship, Dr. Scott McCloskey, Dr. Yu Sheng, Dr. Yen-Lin Chen. I am especially grateful for having had the opportunity to work on a commercial product and gaining valuable experience that will largely benefit my professional development.

I am thankful to my labmates in Graphics and Imaging Lab at the University of Delaware and Visual Computing Lab at ShanghaiTech University, for the sleepless nights we were working together before deadline, and for the joy and laughter we shared over the past years.

This dissertation is supported by the Delaware INBRE under a grant from NIGMS(8P20GM103446) at National Institutes of Health, and grants IIS-1218177 and IIS-1218156 from National Science Foundation. Part of the research was conducted when I was an intern at Honeywell ACS Lab and Plex-VR.

Personally, this dissertation is impossible without my parents' love and sacrifice. My gratitude to them is beyond words. Lastly, I must express my gratitude to my wife, Yusu, for her continued support and encouragement. She takes good care of the family and I cannot thank her enough for that.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Acquiring 3D geometry of the scene is a key task in computer vision. Applications are numerous, from classical object reconstruction and scene understanding to the more recent visual SLAM and autonomous driving. Recent advances in computational imaging have enabled many new solutions to tackle the problem of 3D reconstruction. By modifying the camera's components, computational imaging optically encodes the scene, then decodes it with tailored algorithms.

This dissertation focuses on exploring new computational imaging techniques, combined with recent advances in deep learning, to infer 3D geometry of the scene. In general, our approaches can be categorized into active and passive 3D sensing.

For active illumination methods, we propose two solutions: first, we present a multi-flash (MF) system implemented on the mobile platform. Using the sequence of images captured by the MF system, we can extract the depth edges of the scene, and further estimate a depth map on a mobile device. Next, we show a portable immersive system that is capable of acquiring and displaying high fidelity 3D reconstructions using a set of RGB-D sensors. The system is based on structured light technique and is able to recover 3D geometry of the scene in real time. We have also developed a visualization system that allows users to dynamically visualize the event from new perspectives at arbitrary time instances in real time.

For passive sensing methods, we focus on light field based depth estimation. For depth inference from a single light field, we present an algorithm that is tailored for barcode images. Our algorithm analyzes the statistics of raw light field images and conducts depth estimation with real time speed for fast refocusing and decoding. To mimic the human vision system, we investigate the dual light field input and propose a unified deep learning based framework to extract depth from both disparity cue and

focus cue. To facilitate training, we have created a large dual focal stack database with ground truth disparity. While above solution focuses on fusing depth from focus and stereo, we also exploit combing depth from defocus and stereo, with an all-focus stereo pair and a defocused image of one of the stereo views as input. We have adopted the hourglass network architecture to extract depth from the image triplets. We have then studied and explored multiple neural network architectures to improve depth inference. We demonstrate that our deep learning based approaches preserve the strength of focus/defocus cue and disparity cue while effectively suppressing their weaknesses.

Chapter 1

INTRODUCTION

Inferring shape from images is one of the fundamental challenges in computer vision. Applications include object reconstruction, scene understanding, robotic navigation, visual SLAM and autonomous driving. Recent advances in computational imaging have enabled many new solutions to recover the geometry of the scene. By modifying the camera's components, computational imaging optically encodes the scene then decodes it with tailored algorithms. For the purpose of depth inference, existing approaches can be generally categorized into active and passive 3D sensing.

Active methods rely on programmable light source to illuminate the camera's field of view. Laser range finder [17, 63] projects a light stripe on the scene while observing it from an offset viewpoint. The deformation of the stripe in the image infers the 3D location the scene by using the optical triangulation. Alternatively, time-of-flight (ToF) sensor [44, 37] obtains the depth information by measuring the time that it takes for laser light to travel between the object and the sensor. Traditional depth sensor requires complex hardware and features expensive price until Microsoft has released Kinect sensor [50, 82], making such sensor accessible to the average consumers. Kinect sensor utilizes structured light technique or ToF (updated version) to generate real-time depth map of the physical scene.

Passive depth sensing acquire depth information by modifying camera components, which are typically cost-effective and can conduct non-intrusive depth measurements. Light field camera [84, 68] features a microlens array that is placed on top of the sensor to optically sort the rays by direction onto the pixels underneath. A single shot of the light field camera is amount to capturing the same scene from multiple perspectives, where large overlap between views could be observed. Given the light field, one can recover depth by first matching the correspondences between views, then patching gaps by imposing specific priors, e.g., induced by the Markov Random Field [57]. Further, the regularly sampled light field exhibits a special line pattern on the epipolar plane image (EPI), where the pixel's depth is associated with the slope of the line [18, 126, 52, 113]. While above methods rely on photo-consistency between views to estimate depth, another important passive method, namely depth from focus/defocus (DfF/DfD), captures a sequence of images with different focus setting, then infers depth by analyzing blur variations at same pixel. The capturing process utilizes complex optical design, such as the telecentric optics [114], or focal sweep camera [135, 74, 124] with moving sensor or deformable lens.

Recent advances in neural network have revolutionized both high-level and low-level vision by learning a non-linear mapping between the input and output. Contrasting to the conventional methods that rely on hand-crafted features and engineered cost functions, the data driven approach is capable of learning more discriminative features from the images and inferring the depth with robustness and efficiency. Existing solutions mainly focuses on extracting depth from single image [22], or stereo pair [127, 128, 69]. Applying deep learning to light field stereo has also been investigated [43].

Although impressive progress have been achieved, there are still several open problems. Compared with the decreasing price of smart devices, active 3D sensing methods on the mobile platform are still less affordable. Meanwhile, active sensing for large scale 3D reconstruction faces the problems of huge data bandwidth, imperfect registration, and limited 3D viewing tools. For passive methods, especially for light field data, real time depth estimation is still challenging. Finally, combining the disparity cue and the focus cue in a deep learning framework has not been investigated.

1.1 Dissertation Statement

This dissertation focuses on exploring new computational imaging techniques, combined with recent advances in deep learning, to infer 3D geometry of the scene.

In general, our approaches can be generally categorized into active and passive 3D sensing.

1.1.1 Active 3D Sensing

Depth Acquisition using Mobile Multi-flash System To acquire depth on mobile platform, we implement the multi-flash (MF) system on mobile platform. MF system offers a number of advantages over regular photography since the location and width of the shadow encodes the geometrical information of the scene. Implementing MF system on mobile devices, however, is challenging due to their restricted form factors, limited synchronization capabilities, low computational power and limited interface connectivity. To overcome these limitations, we develop a novel mobile MF system that achieves comparable performance as conventional MF. We first construct a mobile flash ring using four LED lights and design a special mobile flash-camera synchronization unit. The mobile devices own flash first triggers the flash ring via an auxiliary photocell. The mobile flashes are then triggered consecutively in sync with the mobile cameras frame rate, to guarantee that each image is captured with only one LED flash on. To process the acquired MF images, we further develop a class of fast mobile image processing techniques for image registration, depth edge extraction [90]. We also adopt shape-from-shadow [20, 27] techniques to obtain a qualitative depth map. With the depth map and its corresponding color image, we can construct a light field that enables us to synthesis shallow depth-of-field effects and interpolate novel views.

A Portable Immersive System using RGB-D Sensor To achieve 3D reconstruction of a room-sized environment in real time, we develop a new portable immersive system that is capable of acquiring and displaying high fidelity 3D reconstructions. Traditional solutions [31, 32, 91, 95, 65, 116], e.g., from Fuch's group at UNC, Bajcsy's group at Penn, Kanade's group at CMU, and Gross's group at ETH, have pioneered the use of a "sea of cameras" around a room. However, the system infrastructure is bulky, and recovering 3D scene geometry from images is still one of the open problem in computer vision. Instead, we resolve both system and reconstruction problems by using a small number ($2 \sim 4$) of the emerging 3D sensors, namely Microsoft Kinect. Our system consists of three major components. The first component, image acquisition, captures images and depth data using a set of Microsoft Kinect cameras and recovers the camera calibration matrix [130, 46] for each view. Next, the data fusion and 3D stereoscopic rendering module combines the image and depth data to generate a 3D point cloud from each view and utilizes the camera calibration parameters to fuse individual data into a global 3D point cloud, which is subsequently rendered as a 3D stereoscopic view of the scene. Finally, the data navigation module allows users to dynamically visualize the surgical event from new perspectives at arbitrary time instances in real time. To deliver an immersive experience, our system supports the display of stereo contents either on a 3D monitor, 3D projector or an autostereoscopic display.

1.1.2 Passive 3D Sensing

Depth from a Single Light Field We develop a class of advanced algorithm to facilitate the depth estimation of a special target: barcode. Relative to standard barcode readers, which typically use fixed-focus cameras in order to reduce mechanical complexity and shutter lag, employing a light field camera significantly increases the scanners depth of field. However, the increased computational complexity that comes with software-based focusing is a major limitation on these approaches. Whereas traditional light field rendering involves time-consuming steps intended to produce a focus stack in which all objects appear sharply-focused, a scanner only needs to produce an image of the barcode region that falls within the decoders inherent robustness to defocus. With this in mind, we speed up image processing by segmenting the barcode region before refocus is applied. We then estimate the barcodes depth directly from the raw sensor image, using a lookup table characterizing the relationship between depth and the codes spatial frequency. In contrast to depth estimation method [18, 105, 30, 113] for general scenes which are geometrically complex, our work focuses on barcode imaging based on its unique frequency characteristics, thus largely reducing the computational cost. We demonstrate that our system can produce a decodable image in near real time.

Depth from Dual Light Fields The human visual system relies on both binocular stereo cues and monocular focusness cues to gain effective 3D perception. In computer vision, the two problems are traditionally solved in separate tracks. To simultaneously uses both types of cues for depth inference, we develop a unified learning-based technique. Specifically, we use a pair of focal stacks as input to emulate human perception. We first construct a comprehensive focal stack training dataset synthesized by depth-guided light field rendering. We then construct three individual networks: a Focus-Net to extract depth from a single focal stack, a EDoF-Net to obtain the extended depth of field (EDoF) image from the focal stack, and a Stereo-Net to conduct stereo matching. We show how to integrate them into a unified solution to obtain high quality depth maps. Comprehensive experiments show that our approach outperforms the state-of-the-art in both accuracy and speed and effectively emulates human vision systems.

Hybrid Depth from Defocus and Stereo Imaging Depth from defocus (DfD) and stereo matching are two most studied passive depth sensing schemes. The techniques are essentially complementary: DfD can robustly handle repetitive textures that are problematic for stereo matching whereas stereo matching is insensitive to defocus blurs and can handle large depth range. We present a unified learning-based technique to conduct hybrid DfD and stereo matching. Our input is image triplets: a stereo pair and a defocused image of one of the stereo views. We first apply depth-guided light field rendering to construct a comprehensive training dataset for such hybrid sensing setups. Next, we adopt the hourglass network architecture to separately conduct depth inference from DfD and stereo. Finally, we exploit different connection methods between the two separate networks for integrating them into a unified solution to produce high fidelity 3D disparity maps. Comprehensive experiments on real and synthetic data show that our new learning-based hybrid 3D sensing technique can significantly improve accuracy and robustness in 3D reconstruction.

1.2 Dissertation Overview

This dissertation is organized as follows. Chapter 2 reviews the background on the computational imaging techniques, including the active and passive methods, as well as deep learning based approach for 3D inference.

Chapter 3 introduces the active methods for geometry estimation. We first describe a mobile multi-flash system to obtain depth on a mobile platform. We then explore using multiple RGB-D sensors to reconstruct room size geometry in real time.

Chapter 4 discusses an approach to efficiently extract depth from a single light field by analyzing the variance of pixel intensities in the raw light field image [1].

Chapter 5 presents a learning based framework to acquire depth from binocular focal stacks that are generated by dual light field [2].

Chapter 6 proposes a learning based framework to combine disparity cue and defocus cue for depth estimation and exploits different network architectures for enhanced performance [3].

Chapter 7 concludes the dissertation and discusses the future directions for this work.

---

[1] This research was done while I was an intern at Honeywell ACS Lab

[2] This research was done while I was an intern at Plex-VR

[3] This research was done while I was an intern at Plex-VR

Chapter 2

RELATED WORK

This chapter introduces the background and the related work on 3D inference using computational imaging, which can be generally categorized into active and passive 3D sensing. I will first review the methods based on active illumination, including flash-based computational photography and structured light. For passive methods, I will discuss the recent advances in light field stereo and depth from focus/defocus. Finally, I will review the existing work on deep learning technique and how it could be applied to computational imaging for depth inference.

2.1 Active Illumination

Active methods use a programmable light source to illuminate the scene. In this dissertation, we mainly focus on using flash based computational photography and structured light technique.

Flash-based Computational Photography Flash-based computational photography has attracted much attention in the past decade. Earlier approaches aim to enhance imaging quality by fusing photographs captured with and without flash. The seminal flash/no-flash pair imaging applies edge preserving filters to enhance noisy no-flash images with high quality flash images. Eisemann and Durand [23] and Petschnigg *et al.* [87] used the no-flash image to preserve the original ambient illumination while inserting sharpness and details from the flash image. Krishnan *et al.* [58] explored the use of non-visible light (UV/IR) flashes and demonstrated how different wavelength imagery can be used to for image denoising.

Raskar *et al.* presented the first multi-flash(MF) camera [90] that used an array of flashes surrounding the central SLR camera. They take multiple shots of the scene, each with only one flash. Each flash casts a different shadow abutting the occlusion boundary of the object and they extract the boundaries by traversing along the flash-camera epipolar line. Feris *et al.* [27] further showed that one can obtain a qualitative depth map using MF photography, which assists stereo matching. They derived object depths (disparities) from shadow widths and then applied belief propagation for scene reconstruction. Under industrial applications, [66] mounted MF cameras on robots for enhancing object detection, localization and pose estimation in heavy clutter.

Previous MF photography is sensitive to specular surfaces, thin objects, lack of background, and moving objects, and a number of extensions have been proposed to address these issues. To find proper flash-camera configuration, Vaquero *et al.* [109] investigated the epipolar geometry of all possible camera-light pairs to characterize the space of shadows. Their analysis can be used to derive the lower bound on the number of flashes, as well as the optimal flash positions. Tremendous efforts have also been made to reduce the number of flashes or shots in MF. Feris *et al.* [28] used color multiplexing to more robustly handle multi-scale depth changes and object motion. They have shown that for some special scene configurations, a single shot with the color flash is sufficient for depth edge extraction whereas for general scenes, a color/monochrome flash pair would be enough. Recently, Taguchi *et al.* [103] utilized a ring color flashes of continuous hues for extracting the orientation of depth edges.

Structured Light Early approaches of structured light [2, 5, 8, 16] project a special light pattern onto the scene, then infer the scene depth using a single projector and a single camera. For a detailed overview, we refer the readers to [60, 33]. The rapid advances in structured light technique are enabling us to capture geometric data with unprecedented ease and accuracy. Recent depth sensors project a dots patterns onto the scene and analyze the distribution of the dots to obtain depth in real time. Based on such technique, Microsoft Kinect sensor offers real time high resolution depth maps. With Kinect sensor, Izadi *et al.* [50] presented KinectFusion, a real time method to simultaneously reconstruct a room-size scene and track the camera. The reconstructed scene could be used in a variety of interactive scenarios and augmented reality. To address the scalability issue for real-time volumetric surface reconstruction, [12] designed a memory efficient, hierarchical data structure which fuses overlapping depth maps into single volumetric representation.

2.2 Passive Methods

Without modifying the light source, passive methods change camera components to encode information when capturing an image and later decode with signal processing.

2.2.1 Light Field Stereo

Integral or light field photography describe the scene by recording radiances of rays emanated from objects' surface. A distinct advantage of light field is its capability to produce novel views with high photorealism [62, 36] and conduct refocusing after exposure [49]. To describe the radiance along all rays in 3D space, Adelson and Bergen [1] proposed the 5D plenoptic function that describes both position and direction of the ray. However, when light travels in free space without occluders, one can reduce the 5D plenoptic function to 4D representation: a ray can be parameterized by its intersection with two parallel planes Π_{st} and Π_{uv} and described as $L(u, v, s, t)$, where st is the camera plane and uv is the image plane. This representation is called 2-plane parameterization (2PP) [62, 36]. By fixing s and t, one obtain the sub-aperture image $L_{(s^{\star}t^{\star})}(u, v)$ that is amount to the image captured using a sub-region of the main lens aperture. Therefore, light field can be regarded as a collection of images captured from multiple viewpoints.

Light field acquisition proves to be a challenging task due to its high dimensionality. Early approaches [122, 108, 117, 118, 107] utilized camera arrays that deliver high spatial resolution and moderate angular resolution at the expense of bulky and complex system infrastructure. Alternatively, Ng [84] designed a hand-held light field camera where a microlens array is placed on top of the sensor to optically sort the rays by direction onto the pixels underneath. To increase the spatial resolution, Lumsdaine *et al.* [68] presented a slightly different design by focusing the microlens array on a virtual plane inside the camera. Other acquisition methods, such as coded masks [110] and gantry-based camera systems [106], have also been proposed. In contrast to the aforementioned special camera system, Davis *et al.* [21] use a hand-held commodity camera to interactively acquire and render light field in an unstructured manner.

Light field encodes not only the visual appearance, but also the geometric information of the scene. In essence, light field can be seen as a set of multi-view images, thus could be solved by conventional multi-view stereo approaches [57, 119]. However, the regular sampling pattern of the light field enables novel depth estimation methods that are more efficient and accurate than conventional approach. Wanner and Goldlücke [112] employed structure tensor on the 2D epipolar plane image (EPI) to find the direction of local level lines and enforce globally consistent visibility for depth labeling. Tao *et al.* [105] analyzed the EPI and found that the horizontal variance after vertical integration of the EPI encodes defocus cue, while vertical variances encode disparity cue. The two cues were then jointly optimized in an MRF framework. Kim *et al.* [52] first estimated depth from EPI on depth pixels, then propagated the depth value along the EPI and finally to the coarser EPI resolution. Without global optimization process, their method is able to handle light field with high spatial-angular resolution. Heber and Pock [42] observed a large amount of overlap among light field data and formulate the model to perform a low rank minimization on the stack of warped images. Most recently, Jeon *et al.* [51] described a method to extract depth from lenslet light field images with extremely small baseline. At its core is applying the phase shift theorem in the Fourier domain to achieve sub-pixel accuracy.

2.2.2 Depth from Focus/Defocus

The amount of blur carries information about the objects' distance. Depth from focus/defocus(DfF/DfD) recovers scene depth from a collection of images captured under varying focus settings.

Depth from Focus In general, depth from focus (DfF) [80, 81, 71] exploits differentiations of sharpness at each pixel across a focal stack and assigns the layer with the highest sharpness as its depth. To avoid ambiguity in the textureless region, Moreno-Noguer *et al.* [76] used active illumination to project a sparse set of dots onto the scene. The defocus of the dots offers depth cue, which could be further used for realistic refocusing. [39] combined focal stack with varying aperture to recover scene geometry. Moeller *et al.* [75] applied an efficient nonconvex minimization technique to solve DfF in a variational framework. Suwajanakorn *et al.* [101] proposed the DFF with mobile phone under uncalibrated setting. They first aligned the focal stack, then jointly optimized the camera parameters and depth map, and further refined the depth map using anisotropic regularization.

Depth from Defocus Depth from defocus (DfD) infers depth based on the amount of the spatially varying blur at each pixel. Earlier DfD techniques [96, 88, 115] rely on images captured with different focus setting (moving the objects, the lense or the sensor, changing the aperture size, etc). More recently, Favaro and Soatto [26] formulated the DfD problem as a forward diffusion process where the amount of diffusion depends on the depth of the scene. [61, 134] recovered scene depth and all-focused image from images captured by camera with binary coded aperture. Based on a per-pixel linear constraint from image derivatives, Alexander *et al.* [4] introduced a monocular computational sensor to simultaneously recover depth and motion of the scene. Varying the size of the aperture [86, 24, 100, 9] has also been extensively investigated. This approach will not change the distance between the lens and sensor, thus avoiding the magnification effects.

Combining Stereo and DfD In the computational imaging community, there has been a handful of works that aim to combine stereo and DfD. Early approaches [55, 97] use a coarse estimation from DfD to reduce the search space of correspondence matching in stereo. Rajagopalan *et al.* [89] used a defocused stereo pair to recover depth and restore all-focus image. Recently, Tao *et al.* [105] analyzed the variances of the epipolar image (EPI) to infer depth: the horizontal variance after vertical integration of the EPI encodes the defocus cue, while vertical variance represents the disparity cue. Both cues are then jointly optimized in a MRF framework. Takeda *et al.* [104] exploited the relationship between point spread function and binocular disparity in the frequency domain, and jointly resolved the depth and deblurred the image. Wang *et al.* [111] presented a hybrid camera system that is composed of two calibrated auxiliary cameras and an uncalibrated main camera. The calibrated cameras were used to infer depth and the main camera provides DfD cues for boundary refinement.

2.2.3 Deep Learning based Stereo Methods

Recent advances in neural network have revolutionized both high-level and low-level vision by learning a non-linear mapping between the input and output. Here we mainly discuss the learning based stereo techniques.

One stream focuses on learning the matching function. The seminal work by Žbontar and LeCun [128] leveraged convolutional neural network (CNN) to predict the matching cost of image patches, then enforced smoothness constraints to refine depth estimation. [127] investigated multiple network architectures to learn a general similarity function for wide baseline stereo. Han *et al.* [38] described a unified approach that includes both feature representation and feature comparison functions. Luo *et al.* [69] used a product layer to facilitate the matching process, and formulate the depth estimation as a multi-class classification problem. Other network architectures [14, 67, 85] have also been proposed to serve a similar purpose.

Another stream of studies exploits CNN to predict the confidence of disparity map for outlier removal. Seki and Pollefeys [93] designed a novel two channels disparity patch and incorporated the inferred confidence into Semi Global Matching by adjusting its parameters. Mostegel *et al.* [77] checked the contradictions and consistencies between multiple depth maps produced by the same stereo algorithm to automatically generate the dataset for confidence prediction.

There also exist works that apply end-to-end learning approach. Mayer *et al.* [73] proposed a multi-scale network with contractive part and expanding part for real-time disparity prediction. They also generated three synthetic dataset for disparity, optical flow and scene flow estimation. Knöbelreiter *et al.* [56] presented a hybrid

CNN+CRF model. They first utilized CNNs for computing unary and pairwise cost, then feed the costs into CRF for optimization. The hybrid model is trained in an end-to-end fashion.

Chapter 3

ACTIVE METHODS FOR 3D RECONSTRUCTION

Active illumination uses a programmable light source to illuminate the camera's field of view. In this chapter, I discuss two approaches that use active computational imaging methods for depth estimation. First, I present a multi-flash (MF) system implemented on a mobile platform. Using the sequence of images captured by the MF system, I can extract the depth edges of the scene, and further infer a depth map on a mobile device. Next, I show a portable immersive system that is capable of acquiring and displaying high fidelity 3D reconstructions using a set of RGB-D sensors. Based on structured light technique, our proposed system is able to recover 3D geometry of the scene in real time.

3.1 Depth Inference using Mobile Multi-flash System

3.1.1 Background

Multi-flash (MF) photography takes successive photos of a scene, each with a different flashlight located close to the camera's center of projection (CoP). Due to the small baseline between the camera CoP and the flash, a narrow sliver of shadow would appear attached to each depth edge. By analyzing shadow variations across different flashes, we can recover a depth map of the scene [27] and robustly distinguish depth edges from material edges [90]. MF photography hence can be used to obtain scene geometry, extract occlusion contour, as well as remove the effects of illumination, color and texture in images. Previous MF cameras, however, tend to be bulky and unwieldy in order to accommodate the flash array and the control unit. In this section, we present a mobile MF photography technique suitable for mobile devices such as smartphones or tablets.

Implementing mobile MF photography is challenging due to restricted form factor, limited synchronization capabilities, low computational power and limited interface connectivity of mobile devices. We resolve these issues by developing an effective and inexpensive pseudo flash-camera synchronization unit as well as a class of tailored image processing algorithms. We first construct a mobile flash ring using four LED lights and control it using the mobile device's own flash. Specifically, the mobile flash first triggers the flash ring via an auxiliary photocell, as shown in Fig. 3.1. It then activates a simple micro-controller to consecutively trigger the LED flashes in sync with the mobile camera's frame rate, to guarantee that each image is captured with only one LED flash on.

To process the acquired MF images, we further develop a class of fast mobile image processing techniques for image registration, depth edge and depth map extraction, and edge-preserving smoothing. We demonstrate our mobile MF on a number of mobile imaging applications, including light field construction and rendering, occlusion detection, image thumbnailing, and image abstraction. Compared with traditional MF cameras, our design is low-cost (less than \$25) and compact ($1.75'' \times 2.75''$). Our solution is also universal, i.e., it uses the device's flash, a universal feature on most mobile devices, rather than device-specific external interfaces such as USBs. Experimental results show that our mobile MF technique is robust and efficient and can benefit a broad range of mobile imaging tasks.

3.1.2 Mobile Multi-Flash Hardware

3.1.2.1 System Construction

Figure 3.1 shows our prototype mobile MF device that uses a micro-controller to trigger an array of LED flashes. To control the micro-controller, the simplest approach would be to directly use the mobile device's external interface, e.g., the USB. For example, the recent Belkin camera add-on for iPhone allows the user to have a more camera-like hold on their phone while capturing images by connecting to the data port. However, this scheme has several disadvantages. First, it requires additional wiring on

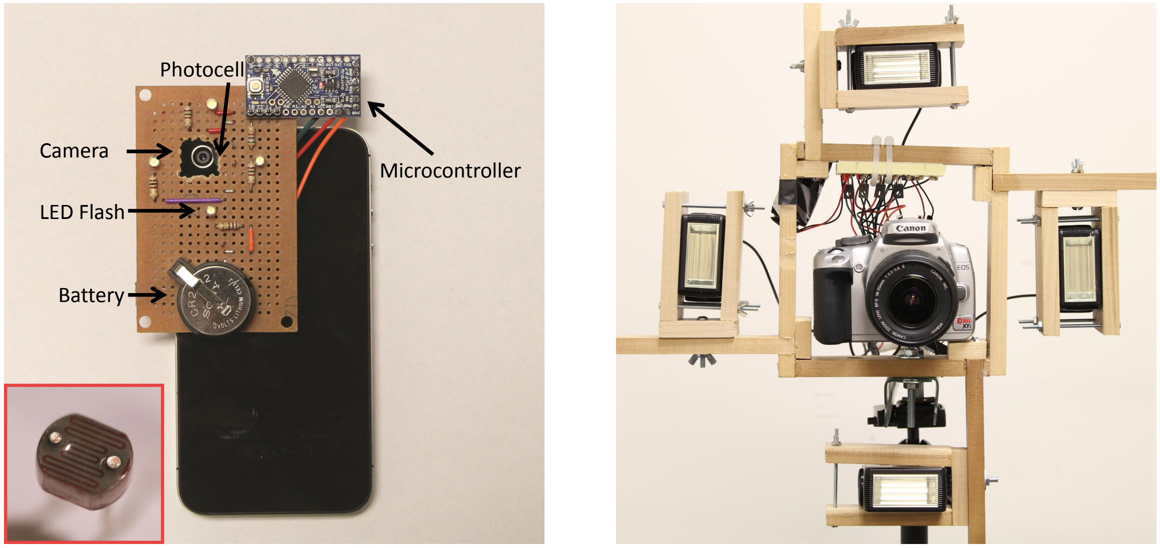


Figure 3.1: (Left) Our prototype mobile MF system. The photocell is hidden on the back of the system. The red highlighted region shows the closeup of the photocell. (Right) Traditional MF system with SLR-camera.

top of the already complex setup. Second, it will occupy the USB interface and limit the use of other application. Finally, each platform (Samsung vs. Apple vs. Nokia) will need to implement its own version of the control due to the heterogeneity of the interface. Other alternatives include the Wi-Fi and the audio jack. However, it would require modifying sophisticated circuitry and the communication protocols.

Our strategy is to implement a cross-platform solution: we use the original flash on the mobile device to trigger the LED flashes. We implement our solution on a perfboard. To reduce the form factor, we choose the Arduino pro mini micro-controller, a minimal design approach ($0.7'' \times 1.3''$) of the Arduino family. We also use small sized but bright LEDs, e.g., the 3mm InGaN white LED from Dialight with a luminous intensity of 1100 mcd and a beam angle of 45 degree. It is worth noting that brighter LEDs are available but many require higher forward current which can cause damage to the micro-controller. In our setup, the baseline between the LED and the camera is about $0.6''$.

To trigger the micro-controller, we put a photocell in front of the device's own flash. The photocell serves as a light sensor that takes the flash signal from the mobile device to trigger the multi-flash array. In our setup, we use a CdS photoconductive photocell from Advanced Photonix. The photocell is designed to sense light from 400 to 700 nm wavelength and its response time is around 30 ms. The resistance is 200k Ohms in a dark environment and will drop to 10k Ohms if illuminated at 10 lux. The complete system is powered by two button cell batteries, making it self-contained. Its overall size is $1.75'' \times 2.75''$ and therefore can be mounted on a wide range of mobile devices, ranging from the iPhone family to the Samsung Galaxy and Note family. For example, even for the smallest sized iPhone 4/4S ($2.31'' \times 4.54''$), our system fits perfectly.

3.1.2.2 Image Acquisition

To avoid the device's flash to interfere with the LED flashes, we initiate the image acquisition process only after the device's flash goes off. The frame rates of the camera and the LED flash ring are set to be identical by software (e.g., the AVFoundation SDK for iOS) and by micro-controller respectively. After acquiring four images, we turn on the device's flash to stop the acquisition module. We also provide a quick preview mode to allow users to easy navigate the captured four images. If the user is unsatisfied with the results, with a single click, he/she can reacquire the image and discard the previous results.

Conceptually, it is ideal to capture images at the highest possible frame rate of the device (e.g., 30 fps on iPhone 4S). In practice, we discover that a frame rate higher than 10 will cause the camera out-of-sync with the flash. This is because the iPhone and the Arduino micro-controller use different system clocks and are only perfectly sync'ed at the acquisition initiation stage. In our implementation, we generally capture four flash images at a resolution of 640×480 images in 0.4s. The low frame rate can lead to image misalignment since the device is commonly held by a hand. We compensate for hand motion by applying image registration (Section 3.1.3.1) directly on mobile devices.

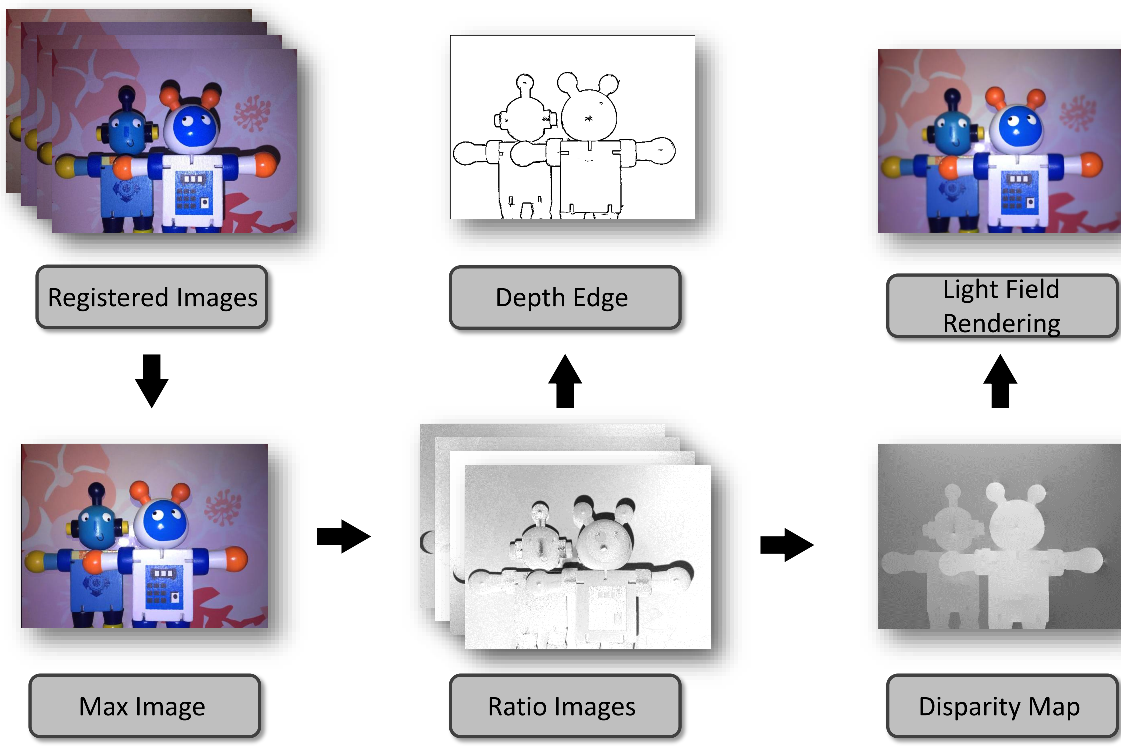


Figure 3.2: The pipeline for disparity map generation and depth edge extraction.

A unique feature of our system is its extensibility, i.e., we can potentially use many more flashes if needed. The Arduino pro mini microcontroller in our system has 14 digital I/O pins: one serves as an input for the triggering signal and the others as output for the LED flashes. Therefore, in theory, we can control 13 flashes with minimum modification.

3.1.3 MF Image Processing

3.1.3.1 Depth Edge Extraction

Traditional MF photography assumes that the images are captured from a fixed viewpoint. In contrast, our mobile MF photography uses a hand-held device and the images are usually shifted across different flashes as we capture with a low frame rate. Extracting depth edges without image alignment will lead to errors as shown in Fig. 3.7(b). In particular, the texture edges are likely to be detected as depth edges. We therefore implement a simple image registration algorithm by first detecting SIFT features and then use them to estimate the homography between images. This scheme works well for scenes that contain textured foreground (Fig. 3.9) or background (Fig. 3.8). It fails in the rare scenario that the scene contains very few textures and the shadow edges become the dominating SIFT features in homography estimations.

Once we align the images, we adopt the shadow traversing algorithm in [90] to extract the depth edges. Figure 3.2 shows the processing pipeline. The captured MF images contain noise, especially under low-light environment. We therefore first convert the color images to grey scale and apply Gaussian smoothing. We denote the resulting four images as $I_k, k = 1..4$ and construct a maximum composite image I_{max} where $I_{max}(x, y) = max_k(I_k(x, y))$. To detect the shadow regions, we take the ratio of a shadow image with the maximum composite image as $R_k = I_k/I_{max}$. The ratio is close to 1 for non-shadow pixels and is close to 0 for shadow pixels. A pixel on the depth edge must transition from the non-shadow region to the shadow region and we apply Sobel filter on each of the ratio images to detect such transitions. In the final step, we apply a median filter to the depth edge image to further suppress the noise. The complete process takes about 1.2s for images with a resolution of 640×480 on an iPhone 4S.

3.1.3.2 Qualitative Depth Map

In addition to the depth edge, we can also infer a qualitative depth map from the MF images. We adopt the method described in [27], which is closely related to shape-from-shadow techniques [20]. Here we will briefly reiterate the method.

Figure 3.3 illustrates the imaging geometry of the shadows: B is the camera-flash baseline, D is the real world shadow width, d is the shadow width on the image, f denotes camera's focal length, z_1 and z_2 are the depths to the shadowing and shadowed edges respectively. Using principles of similar triangle, we can describe the relationship

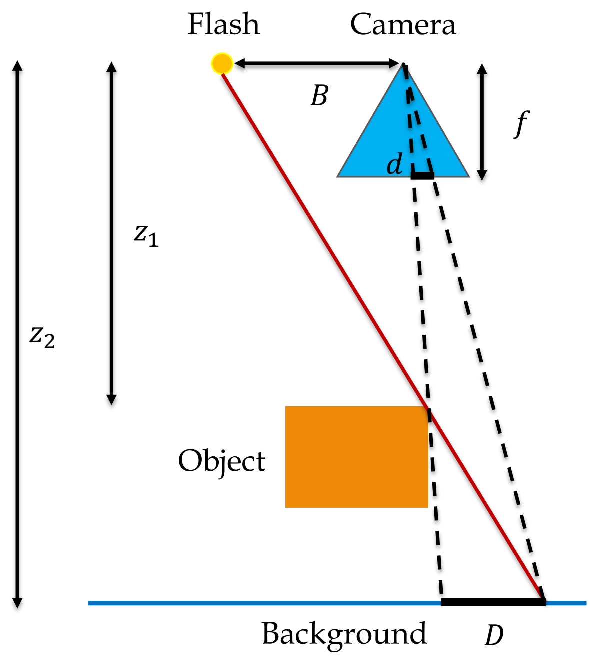


Figure 3.3: An illustration of the relationship between shadow width and relative depth.

between shadow width and relative depth as

$$d = \frac{fB(z_2 - z_1)}{z_1 z_2} \tag{3.1}$$

and we can rewrite the equation as

$$\frac{d}{fB} = \frac{1}{z_1} - \frac{1}{z_2} = \nabla \frac{1}{Z} \tag{3.2}$$

where $Z(x, y)$ is the unknown depth map of the scene. Equation 3.2 indicates that, at the depth edge locations, the shadow depth on the image plane directly encodes the gradient of the inverse depth value.

$$G(x,y) = \begin{cases} (0,0) & \text{if } (x,y) \notin \text{depth edge pixel} \\ \nabla \frac{1}{Z(x,y)} & \text{if } (x,y) \in \text{edge pixel} \end{cases}$$

Next, we can recover the depth by solving an optimization problem $|\nabla M - G|^2$, which amounts to solving a Poisson differential equation. The final depth map can be obtained by $\frac{1}{M}$.

3.1.3.3 Light Field Construction

Light field is a set of rays that depict a scene in place of geometry. In free space, light field is defined by its intersection with two parallel planes, namely the camera plane Π_{st} and image plane Π_{xy}. This definition is in accord with a common practice by storing the light field as a 2D array of images, where (s, t) is the image index and (x, y) is the pixel index.

We can obtain the light field by moving a camera in a 2D grid and capturing the scene. However, the capturing process is time-consuming and will generate a large dataset. Given the fact that the real scene usually contains a large fraction of Lambertian surfaces, the light field is fairly sparse, i.e., for a 3D point, the light field is constant along the angular dimension. Therefore, we can explore this sparse prior for light field reconstruction.

Specifically, we use the maximum composite image as the reference view R_{00} and warp the image based on the disparity map M to construct the light field [125]. Suppose that a 3D point P is captured by R_{00} at pixel $p(x_0, y_0)$, then for light field camera R_{st} we can locate its pixel (x, y) that passes through P,

$$(x, y) = (x_0, y_0) + disparity * (s, t) \tag{3.3}$$

Based on Eqn. 3.3, we can directly warp the pixel in R_{00} onto R_{st}. Since multiple pixels in R_{00} may warp to the same pixel location in R_{st}, we also warp the disparity map to reject pixel whose disparity is smaller than the one stored in the map. In this way we can ensure correct visibility. Meanwhile, the forward warping will introduce holes from occlusion. These holes are filled with image inpainting algorithm [6]. With the light field, we can interpolate novel views and synthesis shallow depth-of-field effects with add-and-shift algorithm [84].

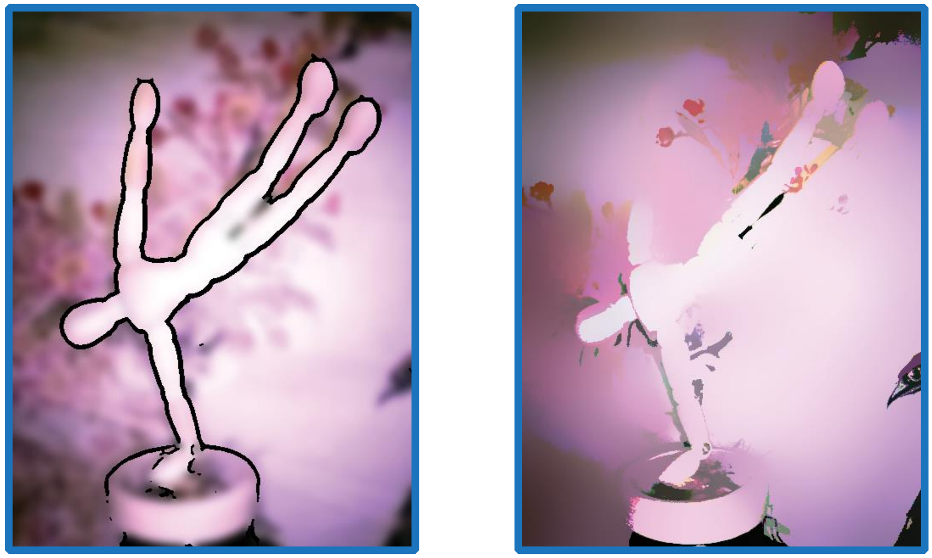


Figure 3.4: (Left) Image abstraction by using anisotropic diffusion. (Right) Image abstraction by using bilateral filter.

3.1.3.4 Non-photorealistic Rendering

From the depth edge image, we can further perform post-processing techniques for synthesizing various non-photorealistic effects.

Line-art Rendering Line-art image [53] is a simple yet powerful way to display an object. Lines not only represent the contour of an object but also exhibit high artistic value. Raskar *et al.* [90] convert the edge image to a linked list of pixels via skeletonization and then re-render each edge stroke. However, it is computationally expensive. We adopt a much simpler approach using simple filtering. We first downsample the image by bicubic interpolation, then apply the gaussian filter, and finally upsample the image. Both bicubic interpolation and gaussian filter serve as low pass filters, which will blur the binary depth edge image. Also users are capable of adjusting the kernel size to control the smoothness. Our processing pipeline is simple, making it suitable for implementation on the mobile platform. iPhone 4S takes about half a second for processing an 640×480 image.

Image Abstraction The most straightforward approach is to use edge-preserving filters such as bilateral filters or anisotropic diffusion [7] to suppress texture edges while preserving depth edges. For example, we can apply the joint bilateral filters [87] that uses the depth image for computing the blur kernel and then blurring the max image I_{max}. A downside of this approach is that the result may exhibit color blending across the occlusion boundaries, as shown in Fig. 3.4. This is because bilateral filters do not explicitly encode the boundary constraint in the blurring process, i.e., the contents to the left and to the right of the edge are treated equally.

To avoid this issue, we apply anisotropic diffusion instead. Specifically, we diffuse the value of each pixel to its neighboring pixels iteratively and use the depth edge as constraints. To ensure that pixels will not diffuse across the depth edge, at the nth iteration, we compute the mask M_n

$$M_n(x,y) = \begin{cases} I_n(x,y) & \text{if } (x,y) \notin \text{edge pixel} \\ 0 & \text{if } (x,y) \in \text{edge pixel} \end{cases}$$

and

$$I_{n+1}(x,y) = \frac{w \sum_{(x_t,y_t)\in N} M_n(x_t,y_t) + M_n(x,y)}{1+4w} \tag{3.4}$$

where N are the neighboring pixels to (x,y) and w is the assigned weight to the neighboring pixels. In our implementation, we simply set $w = 5$. Notice that large w will make the diffusion converge faster and we limit the number of iterations to 15. Finally, we add the edge map to the texture de-emphasized results. On an iPhone 4S, this process takes about 1.5s.

Image Thumbnailing Image thumbnailing [72] reduces the size of the normal image for better organizing and storing. By using bicubic interpolation, we can downsample the de-emphasized image to create a stylized thumbnail image. The depth edges are preserved while the texture regions are blurred, making it suitable for creating icons.

3.1.4 Object Category Classification using Depth Edges

The effectiveness of using depth edges (occluding contours) in object category classification has been reported by recent study [99]. Specifically, depth edges can serve as *feature filter* which help high-level vision tasks to get "purified" shape related features. Here we use similar bag-of-visual-word classification framework as in [99] for evaluation on a dataset collected by the proposed mobile multi-flash camera.

Category Classification Using Bag-of-Visual-Word Model The main idea of bag-of-visual-word (BOW) approach is to represent image as histogram of visual words. 128-dimensional SIFT descriptor is used as independent feature. The dictionary of visual words is learned from training data using clustering method such as k-means. Each training and testing image is represented by histogram of visual words in the dictionary. A classifier is then learned in the space of these visual words for classification task. In this experiment we use Support Vector Machine (SVM) due to its simplicity and discriminative power. As for implementation detail, we chose the LibSVM package and Gaussian kernel.

Feature Filtering Using Depth Edges Sun *et al.* [99] proposed to enhance the BOW framework by filtering out irrelevant features in images using depth edges. Let an image be $I : \Lambda \to [0, 1]$, where $\Lambda \in R^2$ defines the 2D grid. The set of feature descriptors are:

$$\mathcal{F}(I) = \{(\mathbf{x}_i, \mathbf{f}_i)\}, \tag{3.5}$$

where $\mathbf{x}_i$ is the position of the i^{th} feature $\mathbf{f}_i$. After obtaining the depth edge image I_{DE} according to steps mentioned in previous sections, any feature that is far away from valid nonzero I_{DE} pixels will be eliminated. The new feature set $\mathcal{G}$ is defined as:

$$\mathcal{G}(\mathcal{F}(I), I_{mask}) = \{(\mathbf{x}_i, \mathbf{f}_i) \in \mathcal{F}(I) \mid I_{mask}(\mathbf{x}_i) < \tau\}, \tag{3.6}$$

where $I_{mask}(\cdot)$ is the distance transform map of $I_{DE}(\cdot)$ and τ is a preset distance threshold. After filtering, feature descriptors become concentrated around depth edges of objects.

3.1.5 Implementation and Application

3.1.5.1 Implementation

We have implemented our mobile MF system on an iPhone 4S. iPhone 4S features a 1 GHz dual core, a 512 MB RAM and an 8 megapixel camera with a fixed aperture of f/2.4. All examples are captured and rendered at an image resolution of 640×480. The images are captured under indoor conditions to avoid outdoor ambient light overshadowing the LED flash light which would make it difficult to identify shadow regions. Further, the iPhone 4S does not allow the user to control the shutter speed. As a result, under a relatively dim environment, the camera uses a high ISO setting and the acquired images, even under the LED flashes, exhibit noise. However, this is not a major issue for our targeted applications such as depth edge detection, depth map extraction and image abstraction where the smoothing operator for reducing textures also effectively reduces noise.

The camera-flash baseline determines the effective acquisition ranges (i.e., to capture distinctive shadows). If we place the camera too far away, the shadows will be too narrow to be observed due to the small baseline. On the other hand, if we place the camera too close to the object, the LED cannot cover the complete region where the camera is imaging as the LED beam has a relatively small FoV. In practice, we find that the suitable distance for acquiring an object is about $6''$ to $10''$ and the object to background distance is about $2''$ to $3''$. For example, assume the camera-object distance is $9''$ and the object background distance is $2.5''$, reusing the derivation from [90] we can obtain that the shadow width in the image is about 9 pixels on the iPhone 4S camera which uses a focal length of $0.17''$. Further, if the width of the object is smaller than $0.14''$, the shadows can appear detached.

3.1.5.2 Imaging Application

First we demonstrate using our mobile MF camera to recover depth on toy robots of $4''$ in height. We acquire the images with the device held by hand and we rely on the textures on the background and the robots to provide useful features

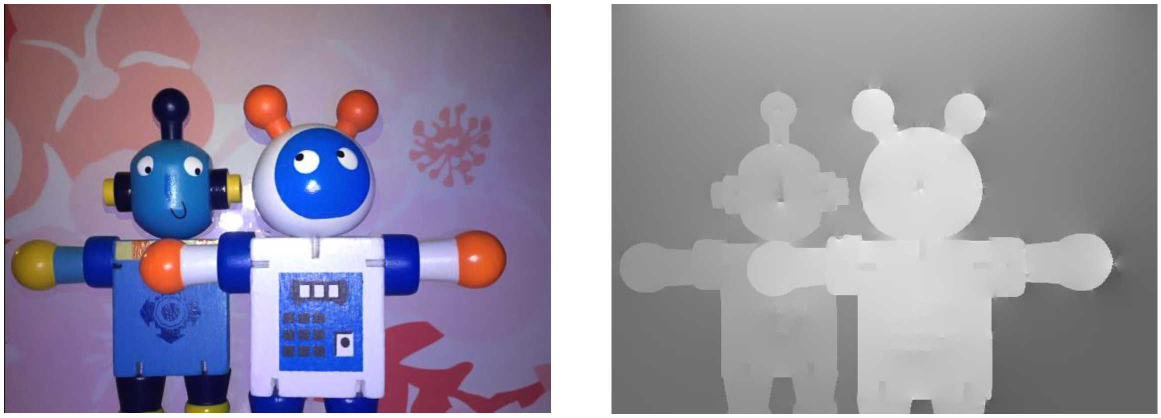
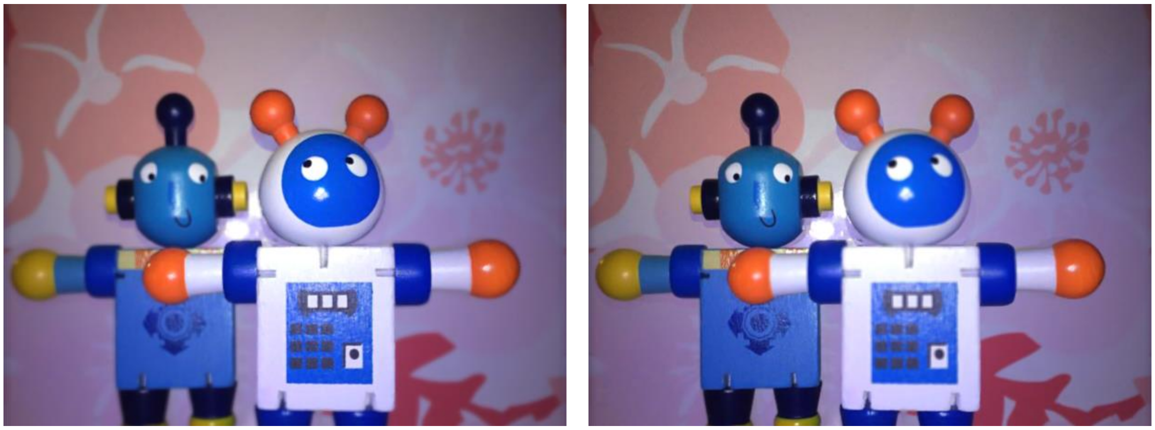


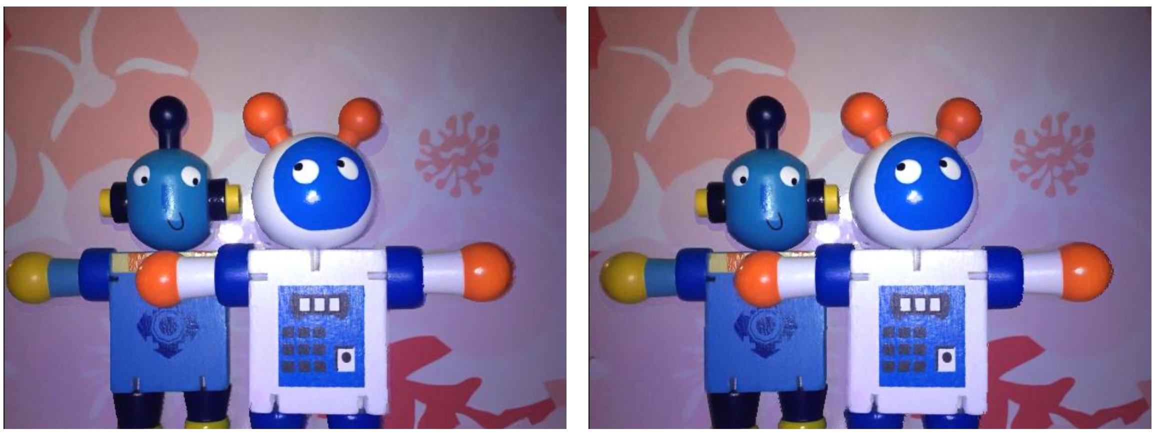
Figure 3.5: The recovered depth map from MF images.

for registering the images. After the image alignment, we can recover a qualitative depth map from the input MF images, as shown in Fig. 3.5. Note that due to the small camera-flash baseline and image noise, the depth map exhibits some artifacts. However, one can still readily identify different depth layers of the scene. Once we obtain the depth map, we are able to generate a light field from the depth map and its corresponding maximum composite image using image warping. The holes in the warped images will be filled with image inpainting algorithm [6]. From the light field we can synthesis shadow depth-of-field effects and produce novel views, as shown in Fig. 3.6.

Next we show depth edge extraction and non-photorealistic rendering on our mobile MF platform. Figure 3.7 shows the MF results on a 6″ cowboy model in front of a white background. We acquire the images with the device held by hand. Fig. 3.7(a) shows one of the LED flashed image and Fig. 3.7(b) shows the extracted depth edges. Compared with Canny edge detection (Fig. 3.7(c)), the MF edge map is of much better quality despite slight hand moves. The results after image registration are further improved as shown in Fig. 3.7(d). We observe though a spurious edge appear on the hat of the cowboy which is caused by detaching shadows due to the small size of the hat. Fig. 3.7(e) and (f) show various non-photorealistic rendering



(a)



(b)

Figure 3.6: (a) The shallow depth-of-field effect with varying position of the focal plane. (b) The interpolated novel view.

effects. The color of the scene is also washed out by the flash and we normalize the maximum composite color images using linear mapping to enhance the color.

Figure 3.8 demonstrates using our mobile MF camera on a headstand mannequin of 5.5″ in height. The mannequin is placed in front of a highly textured background to illustrate the robustness of our technique. Fig. 3.8(b) and (d) show the depth edge results with and without image registration. Despite some spurious edges caused by the specular pedestal, our recovered occlusion contours are generally of good quality. Our technique fails though to capture the inner contour of the legs of the model. We

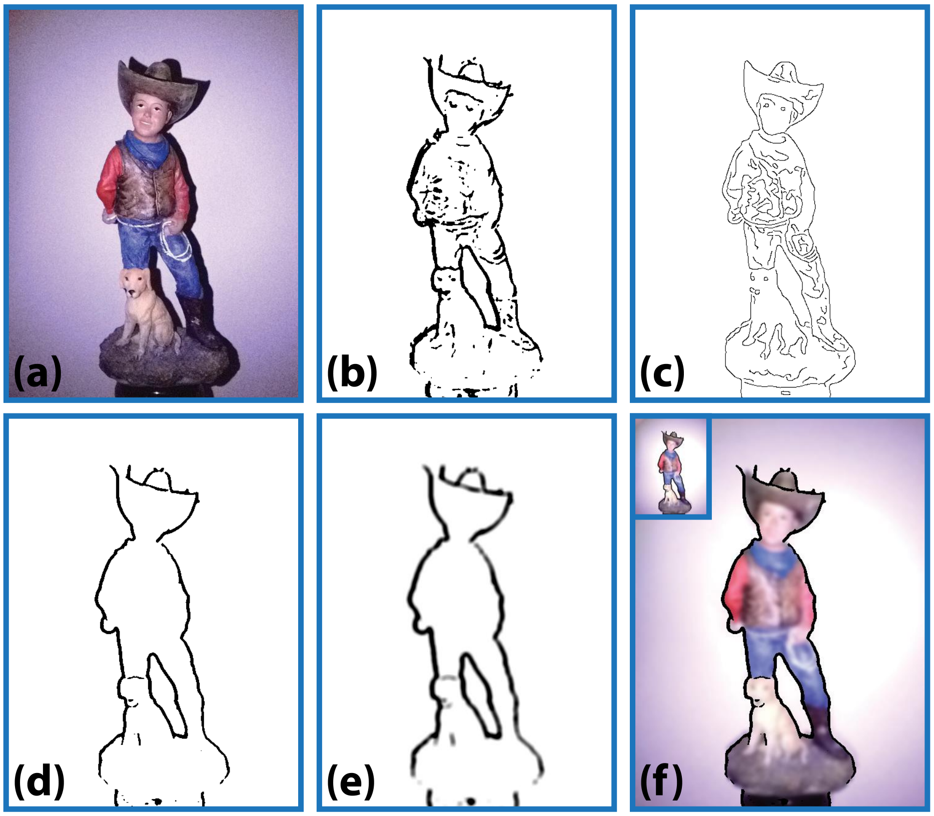


Figure 3.7: (a) The shadowed image. (b) Extracted depth edge image before image registration. (c) Detected depth edge image using Canny edge detector. (d) Extracted depth edge image after image registration and translation. (e) Line-art Rendering. (f) Image abstraction and image thumbnailing.

observe in the maximum image that this area was not well illuminated by any of the four flashes, as shown in Fig. 3.8(a). The problem, however, can be alleviated by using more flashes.

In Fig. 3.9, we show using mobile MF for acquiring a complex plant that are covered by leaves and branches. The scene is challenging for traditional stereo matching algorithms because of heavy occlusions and high similarity between different parts of the scene. Previous SLR-camera based MF systems [90] have shown great success on recovering depth edges on such complex scenes but it uses a bulky setup (Fig. 3.1) and bright flashes. Our mobile MF camera produces comparable results as shown in

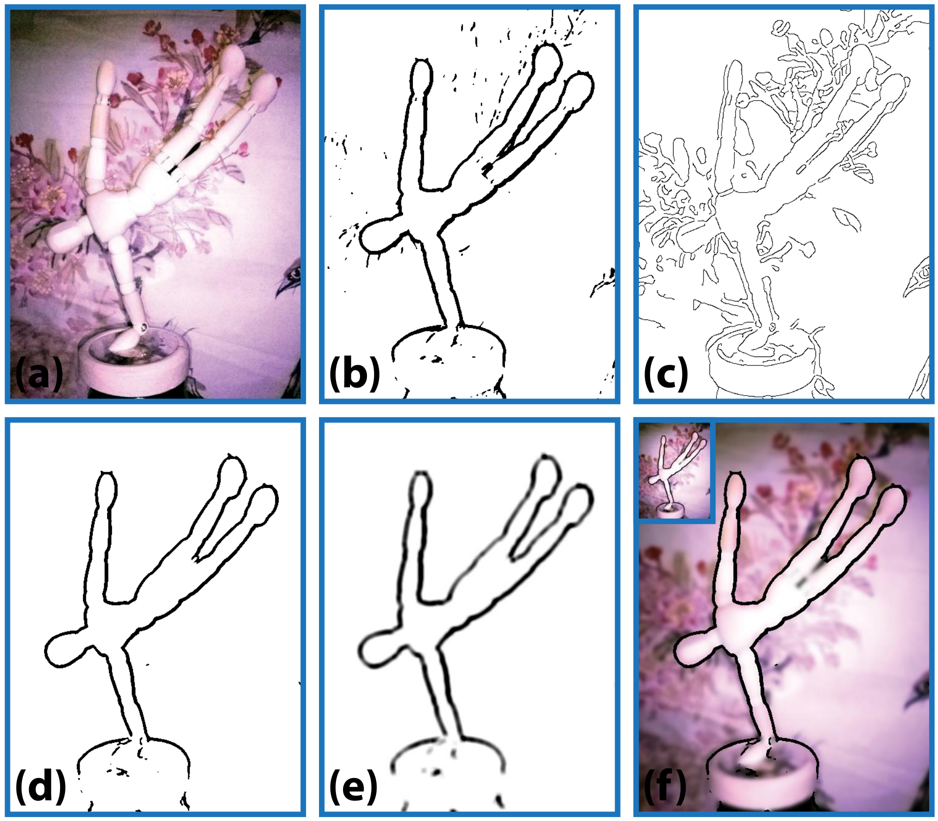


Figure 3.8: (a) The maximum composite image. (b) Extracted depth edge image before image registration. (c) Detected depth edge image using Canny edge detector. (d) Extracted depth edge image after image registration and translation. (e) Line-art Rendering. (f) Image abstraction and image thumbnailing.

Fig. 3.9(b). The thin tip of the leaves cause detached shadows and leads to splitting edges, an artifacts commonly observed in MF-based techniques.

Figure 3.10 demonstrates the potential of using our mobile MF to enhance human-device interactions. We use the mobile MF device for acquiring the contour of hands. Fig. 3.10(c) and (e) compares the foreground segmentation vs. our MF-based edge extraction. As the hand and the background shirt contain similar color and textures, segmentation based method fails to obtain accurate hand contours. In contrast, our mobile MF technique faithfully reconstructs the contours and the results

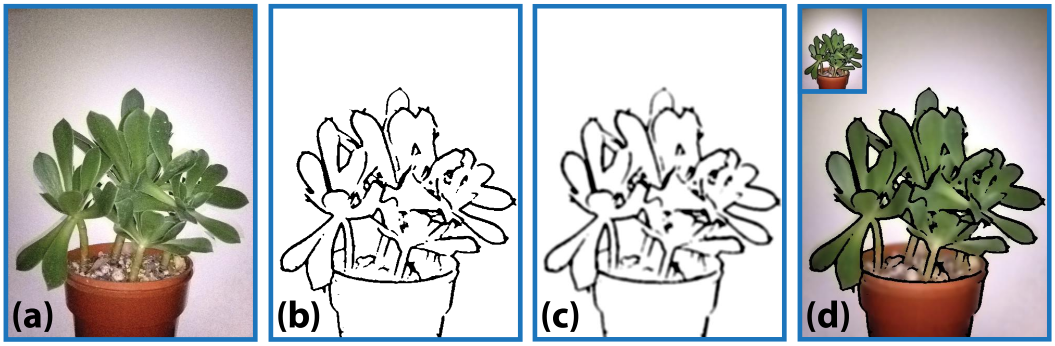


Figure 3.9: (a) The maximum composite image. (b) Extracted depth edge image. (c) Line-art Rendering. (d) Image abstraction and image thumbnailing.

can be used as input to gesture-based interfaces. One downside of our technique though is that the flashes cause visual disturbances. The problem can be potentially resolved by coupling infrared LED flashes such as 1W 850 nm infrared LED from Super Bright LEDs and the infrared camera that is already available on latest mobile devices.

3.1.5.3 Visual Inference Application

For object category classification, we created a dataset containing 5 categories similar to the *Category-5* dataset used in [99]. Each of the 5 categories contains 25 images (accompanied with depth edge images) taken from 5 objects. For each object, images are taken from 5 poses (0°, 90°, 135°, 180°, 270°) with 5 different background. Each image is generated along with depth edges using the proposed mobile multi-flash camera.

Standard bag-of-visual-word (BOW) and BOW with depth edge filtering (BOW+DE) are compared to evaluate the effectiveness of proposed camera. Training and testing sets are randomly divided into half for each run and the experimental result is summarized over 100 such random splits. The performance of BOW and BOW+DE are reported in terms of recognition rate in Table 3.1.

The result has shown that using depth edge images has significant improvement (about 10%) in recognition rate. This result is consistent with that found in [99]. It

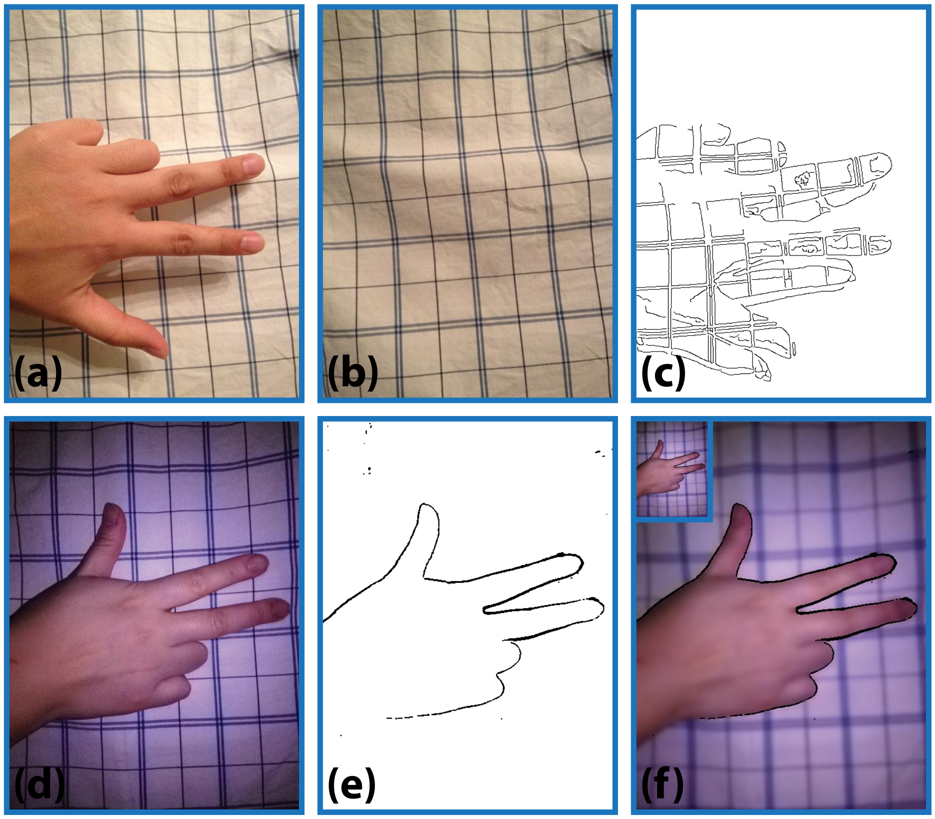


Figure 3.10: (a) The foreground image. (b) The background image. (c) Foreground contour from foreground-background substraction. (d) One shadowed image. (e) The depth edge image. (f) Image abstraction and image thumbnailing.

suggests that the proposed mobile multi-flash camera shares the similar performance with traditional multi-flash camera system but it is much compact and light-weighted.

Method	BOW	BOW+DE
Classification Accuracy (%)	66.52± 4.85	75.42 ± 3.42

Table 3.1: Category classification result.

3.1.6 Discussion

In this section we have presented a new mobile multi-flash camera that uses the mobile device's own flash as a pseudo synchronization unit. Our mobile MF camera is compact, light-weight, inexpensive and can be mounted on most smartphones and tablets as a hand-held imaging system. To process the MF images, we have exported the OpenCV library onto mobile platforms and have developed a class of imaging processing algorithms. Our system is able to register misaligned images due to hand motions, extract depth map and depth edges by analyzing shadow variations, construct a light field and produce non-photorealistic effects. Our solution showcases the potential of exporting computational photography techniques onto mobile platforms.

3.2 A Portable Immersive System using RGB-D Sensor

3.2.1 Background

Acquiring and displaying high-fidelity 3D reconstruction in large scale are challenging tasks in computer vision. Most existing approaches [91, 31, 32, 95, 65, 116], e.g., from Fuchs's group at UNC, Bajcsy's group at Penn, Kanade's group at CMU, and Gross's group at ETH, have pioneered the use of a "sea of cameras" around a room for reconstruction. However, this approach presents difficulties in several aspects: On the system front, it is literally impractical to mount "a sea of cameras" within a room. Most existing multi-camera systems (including the immersive solutions mentioned above) require using multiple workstations just for data transmission and storage. The system infrastructure, such as camera mountings, interconnects, and workstations, is bulky, making them unsuitable for on-site tasks. On the reconstruction front, recovering 3D scene geometry from images is still one of the open problems in computer vision [3, 79]. To make the problem tractable, many existing algorithms tend to make simplified assumptions about scenes, such as Lambertian surface and distant light sources. However, in most environments, we simply cannot assume these factors. For example, in a surgical environments, specular highlights and changing lighting are the norm, easily causing the classical computer vision algorithms, such as binocular stereo or shape-from-shading to break down.

In this section, we present a new immersive system that focuses on room size 3D reconstruction in real time. Our proposed solution resolves both the system and reconstruction problems by leveraging emerging 3D imaging technologies and multi-modal fusion algorithms. Instead of using a large number of cameras, we use a small number ($2 \sim 4$) of 3D sensors, namely Microsoft Kinect. Kinect camera is able to produce real-time depth maps using a structured light technique: it projects a special infrared dots pattern onto the objects, and compute its depth by computing the distribution of the dots. These sensors are uniformly controlled by a single workstation and their range and imagery data are fused via a companion computer vision algorithm for robustly recovering the 3D surgical scene. We further develop a user interface to allow the users to navigate the 3D environment in both space and time.

We have conducted our experiments in a surgical environment, making our system an immersive surgical training system. Although videotaped instruction has long served as a workhorse for teaching surgical procedures, they are marginally effective: videotapes only provide 2D imagery that lacks depth perception and the trainee cannot freely change viewpoints as the inputs are captured from a fixed location. Our preliminary experiments, conducted at the Virtual Education and Simulation Technology (VEST) Center at Christiana Care Health System (CCHS), show that our system can effectively capture and reconstruct 3D surgical procedures performed by an expert. These three-dimensional recordings can be presented in a virtual operation theater in which medical students can perceive solid stereoscopic views without glasses (e.g. on an autostereo display) or with special glasses on a commercial 3D TV, as if they were present in the room.

To summarize, the contributions of this section are the following: (i) We present a portable 3D acquisition system that is capable of acquiring scene geometry in real time. (ii) We fuse the 3D point cloud from each view into a global 3D point cloud representation. (iii) We develop a space-time navigator that allows the user to dynamically explore the scene over space and time.

3.2.2 Methods and Materials

Figure 3.11 shows our proposed immersive system that can automatically recover 3D scenes. Our system consists of three major components. The first component, Image Acquisition, captures images and depth data using a set of Microsoft Kinect cameras and recovers the camera calibration matrix for each view. Next, the Data Fusion and 3D stereoscopic rendering module combines the image and depth data to generate a 3D point cloud from each view and utilizes the camera calibration parameters to fuse individual data into a global 3D point cloud, which is subsequently rendered as a 3D stereoscopic view of the scene. Finally, the Data Navigation module allows users to

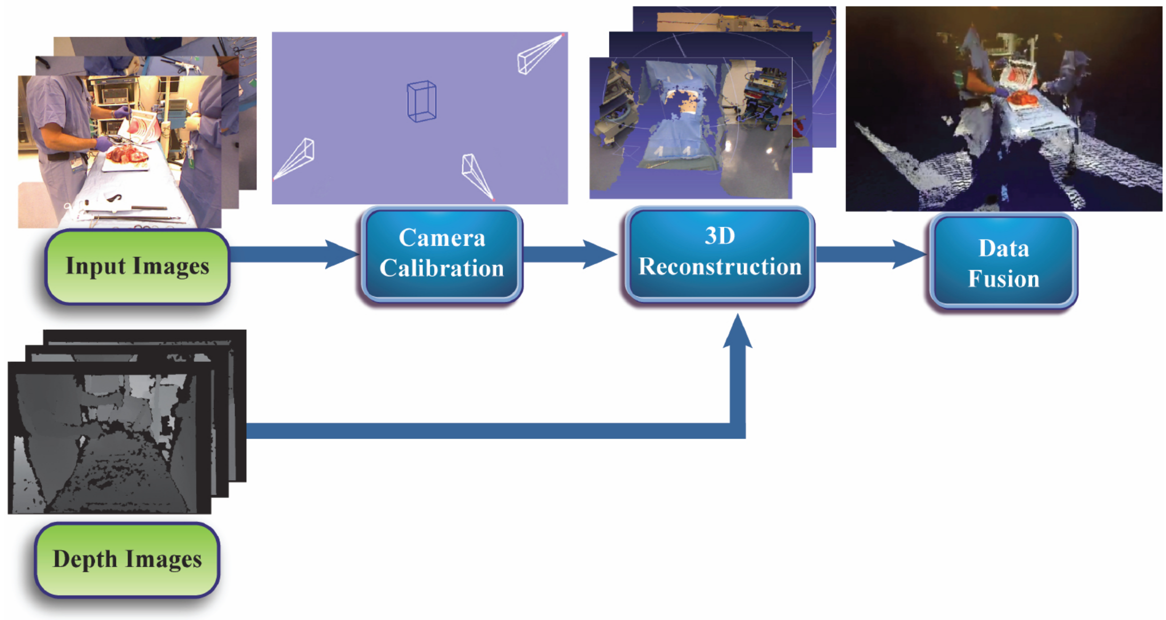


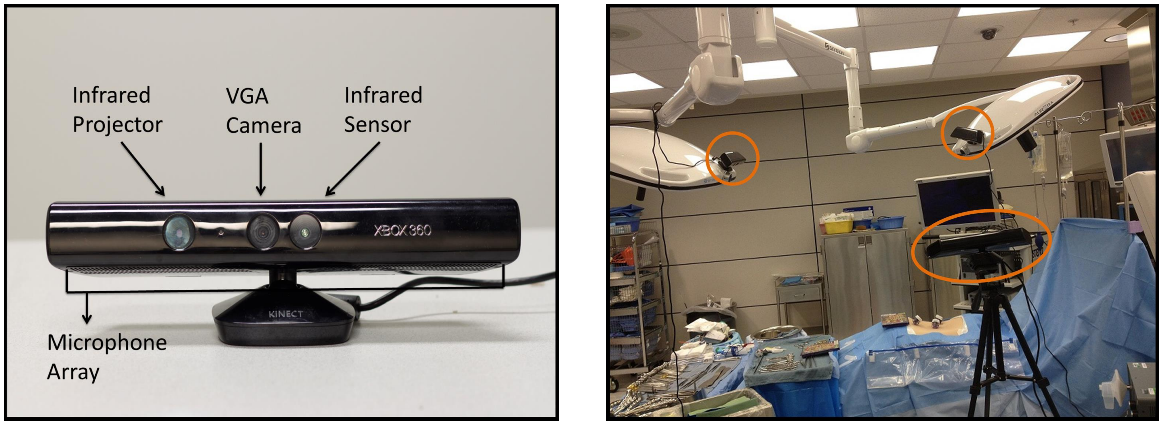
Figure 3.11: Our proposed pipeline for reconstructing and visualizing 3D environments.

dynamically visualize the event from new perspectives at arbitrary time instances in real time.

3.2.2.1 Image Acquisition and Camera Pose Recovery

Figure 3.12 shows our image acquisition system that uses a set of three Microsoft Kinect sensors. Each Kinect sensor consists of an infrared projector, a RGB camera with a res- olution of 640 × 480 pixels, and an infrared sensor. Also, a calibration pattern is used to determine point correspondence to automatically recover the camera calibration parameters. To get access to both depth and RGB image streams, we develop our data fetching module based upon the open source nestk library [64].

In our experiments we strategically mount the Kinect sensors around the operating table to cover both the organs and surgeons during the surgical procedure. A computer with an i7-3930k processor is used to communicate with the Kinect sensors through USB interfaces. During acquisition, both depth and RGB images are captured



(a) (b)

Figure 3.12: (a) Microsoft Kinect has a microphone array, an infrared projector, an infrared sensor and a VGA camera. (b) Acquisition system consists of a set of three Microsoft Kinect cameras.

at a rate of 15 frames per second for all Kinect sensors.

Similar to previous approaches [91, 120, 133, 132], our method requires obtaining the camera calibration matrix for each view. In our solution, the operating room has very similar colors without textures and the occlusion patterns vary significantly across views due to sparse sampling, making it challenging to robustly compute the point correspondence across views. To resolve this issue, we manually identify point correspondences between the corners of the calibration pattern in the acquisition system. We then recover the camera calibration parameters for each view, using the approach described in [70].

3.2.2.2 Data Fusion and 3D Stereoscopic Rendering

Next, we perform a two pass rendering approach that first recovers a point cloud for each viewpoint, and then fuses the individual results to generate a dense set of 3D points that faithfully reconstruct the 3D scene.

Given a depth-RGB image pair, our solution first traces a ray for each pixel in the image and utilizes the depth data to find the corresponding 3D coordinates.

Specifically, for each pixel in the input image we trace a ray originating at the center of projection $\mathbf{C}$ toward the image plane. Let $\bar{\mathbf{r}}$ denote a ray originating from $\mathbf{C}$ toward pixel (u, v) in the image plane. The trajectory of the ray can be described as

$$\bar{\mathbf{r}} = \mathbf{C} + \lambda\bar{\mathbf{d}} \tag{3.7}$$

where $\bar{\mathbf{d}}$ is the direction vector. In camera coordinate system, the direction vector $\bar{\mathbf{d}}$ can be written in terms of camera image plane axis $\bar{\mathbf{d}}_{\mathbf{x}}$ $\bar{\mathbf{d}}_{\mathbf{x}}$, $\bar{\mathbf{d}}$ and the optical axis $\bar{\mathbf{d}}_{\mathbf{z}}$ as

$$\bar{\mathbf{d}} = u\bar{\mathbf{d}}_{\mathbf{x}} + v\bar{\mathbf{d}}_{\mathbf{y}} + f\bar{\mathbf{d}}_{\mathbf{z}} \tag{3.8}$$

Here (u, v) is the pixel coordinate in the image plane and f is the focal length of the camera. Therefore, the original equation can be described as

$$\bar{\mathbf{r}} = \mathbf{C} + \lambda(u\bar{\mathbf{d}}_{\mathbf{x}} + v\bar{\mathbf{d}}_{\mathbf{y}} + f\bar{\mathbf{d}}_{\mathbf{z}}) \tag{3.9}$$

Note that the ray intersects the image plane when $\lambda = 1$. Since the depth image contains a measure of depth along the optical axis, we can conveniently determine λ for each pixel. Thus, for each Kinect sensor we can compute a 3D textured point from each input pixel in the 2D image.

Next, we use the camera calibration parameters to transform the point cloud of each Kinect sensor from local coordinate into a global coordinate system. Then we fuse multiple point clouds into one global point cloud representation. Note that Kinect is de- signed as a stand-alone solution. While a single Kinect sensor delivers quite robust depth maps, simultaneously running multiple sensors may lead to deteriorated results. With the generated point cloud, we set out to render a 3D stereoscopic view of the scene. Traditional 3D rendering generates a single perspective view by synthesizing a pinhole camera image in the scene. We extend this approach by simultaneously setting two cameras in the scene with a user specified baseline. In a single frame, two cameras capture two views of the point cloud and render them with red-cyan anaglyph. We also utilized the NVIDIA 3D API to render two regular color images and synchronize with the NVIDIA 3D glasses to deliver a better user experience. Both passes are mapped onto Mi- crosoft Direct3D graphics framework. Using the state of the art NVIDIA GeForce GTX 580 graphics card, we can render stereoscopic views at over 1000 fps with a resolution of 1280×1024.

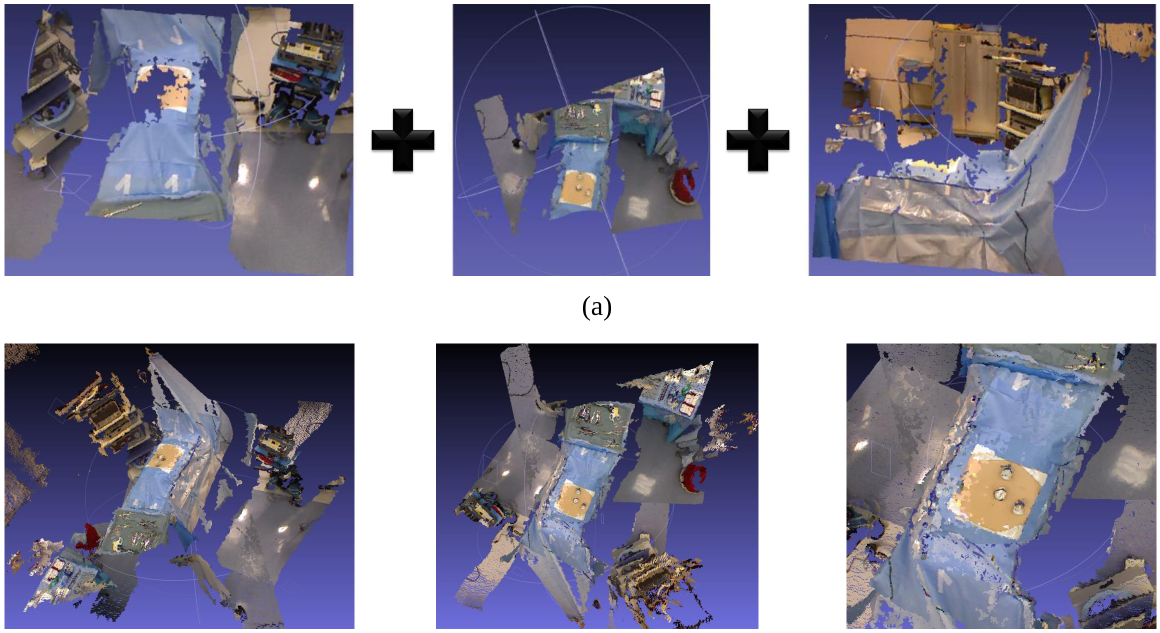
3.2.2.3 Data Navigation

To better help the users to navigate the scene, we have developed a space- time visualization system to display the acquired data. The system includes an interface, which allows users to pick a specific time frame in an event, pause or re- play that time frame and dynamically change viewpoints and have close-up views as if they were there. Our new navigation system thus allows the users to review a surgical procedure without any space or time constraint, as shown in the videos from our project website [102].

3.2.3 Results

To evaluate our proposed system, we bring together both researchers and clinical trainees.We have worked closely with the VEST Center at CCHS, which supports the entire Christiana Care Community (physicians, nurses, allied health professionals, residents, students and regional health services). The VEST Center includes adult and pediatric high-fidelity human patient simulators, a working laparoscopy station with simulated tissues, an endoscopy/bronchoscopy simulator, 3D visualization software and display and numerous task trainers to meet departmental needs. In addition, the VEST center has two operative theaters approved for tissue block surgery, fully fitted with all instrumentation and equipment for surgical procedures.

We used our system to capture a cholecystectomy (gallbladder surgery) on animal tissue blocks conducted by highly trained surgeons at the VEST Center. To cover as many details as possible on the operating table, we used three Kinect sensors facing the table. For training purposes, the surgery took half an hour and we were able to capture five video clips. Figure 3.13(a) shows three point clouds acquired from the



(a)

(b) (c) (d)

Figure 3.13: (a) Combining point clouds acquired from each Kinect sensor. (b) A global point cloud representation. (c) Change of viewpoint. (d) A close-up view.

three Kinect sensors. Figure 3.13(b) shows the global point cloud representation by combining three point clouds. As shown in Fig. 3.13(c) and (d), one can change viewpoints and zoom in and out using our system. Fig. 3.14 shows the 3D stereoscopic view using red-cyan anaglyph. Initial Feedback from the residents shows that our system is much more effective than the conventional videotaped system. These results along with additional videos can be found at [102].

3.2.4 Discussion

We have developed a new immersive system by coupling emerging 3D imaging technologies with advanced computer vision and graphics techniques. Specifically, we use the Microsoft Kinect platform, an inexpensive commercial 3D camera, as the main acquisition device and develop a class of multi-view 3D fusion techniques to faithfully reconstruct the surgical procedure. We have conducted preliminary tests of the system

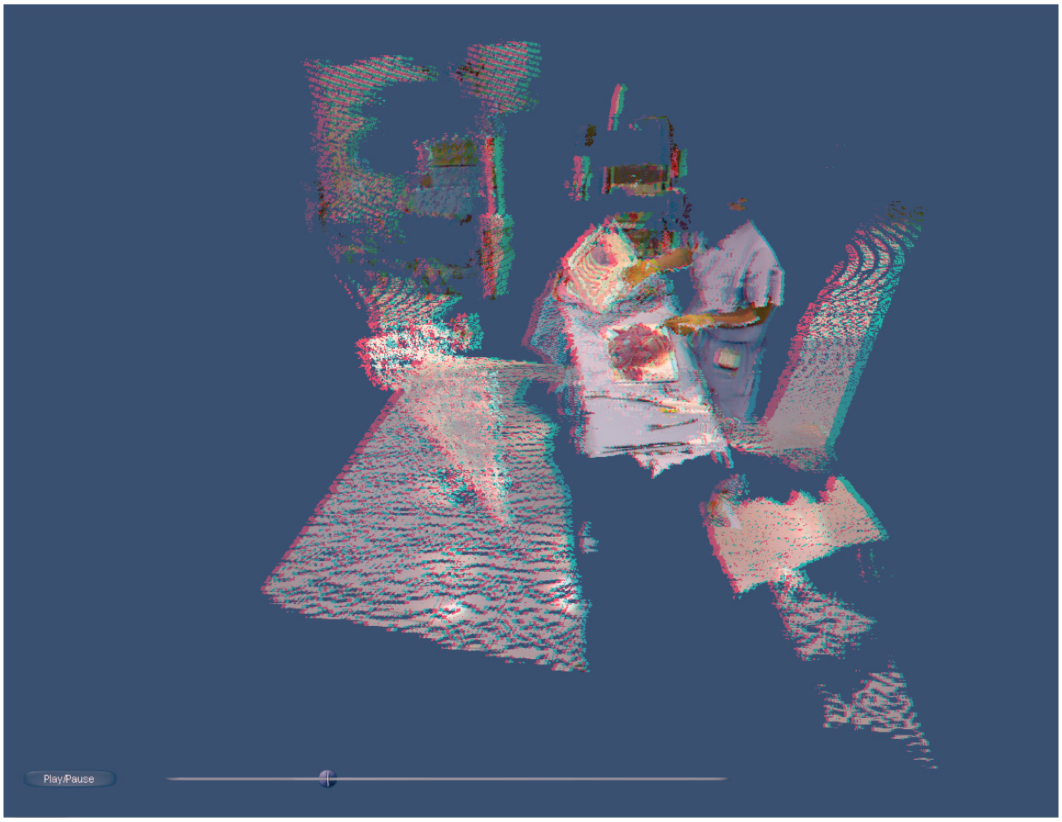


Figure 3.14: 3D stereoscopic view using red-cyan anaglyph.

fidelity for cholecystectomy (gallbladder surgery) training and have developed a space-time visualization system to display the acquired data. Furthermore, we integrate our system with 3D stereoscopic displays to enhance the user experience.

Chapter 4

DEPTH FROM A SINGLE LIGHT FIELD

4.1 Background

A light field (LF) consists of a large collection of rays that store radiance information in both spatial and angular dimensions. An important application of light fields is light field rendering. It has been shown that light field is capable of rendering photorealistic images of complex scene without geometric information. Moreover, one can conduct digital refocusing after exposure, making it suitable for scenarios where auto-focusing is not available or high focusing speed is required.

However, accurately refocusing the target object is a non-trivia problem. Ideally, one can utilize the captured light field to synthesis a focal stack, from which the image with target object in focus can be picked. But synthesizing the complete focal stack requires applying computationally expensive light field rendering schemes, making it prohibited for time-sensitive applications. In order to reduce the time for correct refocusing, knowing the depth of the target object is critical. Tremendous effects [18, 52, 105, 113, 30] has been made to extract depth from LFs. However, most existing methods aim to recover the depth for the general scene. The complexity of different scenes require high computational cost for depth extraction, which largely mitigates the benefit of using depth prior in refocusing tasks.

In this section, we set out to recover the depth of a special target: barcode. A barcode is an optical machine-readable representation of data relating to the object to which it is attached. Nowadays the ubiquitous barcodes found on product packaging significantly improve the speed and accuracy of computer data entry. With increasing popularity of barcodes, 2D imagers have been used to automate the process of barcode reading. These 2D scanners are fundamental low-cost cameras. Therefore, a user would need to manually move the barcode towards or away from the scanner to ensure it is within the depth of field of the scanner. Alternatively, the scanner can conduct a focal sweep and select the proper focal slice to decode. However, implementing focal sweep requires adding moving parts to the scanner, which reduces robustness to mechanical shock. The overwhelming majority of purpose-built scanners are fixed focus for these reasons.

Therefore, a barcode scanning system using the light field camera will address above-noted issues. A light field camera such as Lytro and Raytrix uses a microlens array to capture multiple views of the scene in a single shot. The capability of digital refocusing reduces the mechanical complexity of moving parts in the scanner. To speed up the refocusing process, we develop an image processing algorithm to recover the barcode's depth in real time. We first segment out the barcode region, which we detect from a sub-sampled version of the raw sensor image. Then, we directly estimate the depth of the barcode by analyzing the variance of pixel intensities in the lenslet images formed behind each microlens. Finally, we conduct refocusing only at the estimated depth.

Compared with 2D imagers, our system only adds two extra steps: depth estimation and barcode image rendering. With little computational cost, we gain a system with its range of depth of field nearly triples that of a conventional camera. Comprehensive experiments demonstrate our new light-field based barcode scanner system is fast, accurate and robust to barcode orientation, size variation, and lighting.

4.2 Related Work on Barcode Imaging

Recently, there has been an emerging interest in barcode reading using 2D imagers. Barcode reading consists of two distinct stages: localization and decoding. Tremendous efforts have been made to enhance the performance of both stages. Muniz *et al.* [78] applied hough transform to the image to locate the barcode and find its optimal orientation for further decoding. Zhang *et al.* [129] jointly analyzed the texture and shape information to search for the barcode. Chai and Hock [11] improved the barcode localization by using morphological operator to identify parallel line patterns at block level. Gallo and Manduchi [34] employed a deformable template matching method and enforced global spatial coherence to correctly read barcodes in difficult situations. Xu and McCloskey [121] described a system for localizing and deblurring motion-blurred image using a flutter shutter camera. In contrast to their methods, our system features a better light efficiency and aims at reducing the defocus blur of the barcode image.

4.3 Barcode Scanning Overview

This section explains important differences in barcode reading devices, establishes that current barcode readers are limited by the depth of field, and that reducing spatial resolution is an acceptable tradeoff. Key to this is understanding the distinction between 1D scanline barcode reader and 2D barcode reader.

4.3.1 1D Scanline Barcode Readers

The commonly-used term "scanner" comes from the fact that early 1D readers consisted of a fixed laser source and a rotating prismatic mirror which caused the illumination to scan across the barcode. A single photodiode recorded the temporal variations of the laser reflection, which is roughly equivalent to a 1D image slice. Similar devices are still used at retail checkout counters, but the need for a rotating element makes them ill-suited for hand-held readers. Instead, mobile 1D readers are solid state, with a linear array of photodiodes whose field of view is illuminated by an optically diffused stationary laser. Because only a single line through the barcode needs to be imaged, illumination is high and 1D readers have a large depth of field.

4.3.2 2D Barcode Readers

Because 1D barcodes have limited capacity, several different 2D codes have been introduced, readers of which are essentially monochrome cameras. Unlike consumer cameras, however, 2D barcode scanners are solid state for robustness to physical shock[1]. As such, a fixed focal position is used which, in combination with the aperture, determines the depth of field over which a sufficiently sharp barcode image can be obtained. The well-known tradeoff with exposure, made by changing the aperture, is used in scanner design, but apertures smaller than f/10 are rarely used due to limitations on active illumination[2]. To work around this limitation, 2D barcode readers (e.g., Honeywell Xenon 1900 and Motorola DS4208) are offered in multiple sub-models with different focal positions, but this is inconvenient for many users.

4.3.3 Resolution to Trade

Without being able to increase illumination, and with modern sensors already close to 100% quantum efficiencies, we argue that a microlens-based LF camera is the best way to expand scanning depth of field. As in the Lytro camera, this approach gives an image with diminished spatial resolution relative to that of the sensor, but barcode scanners - most of which currently use sub-megapixel sensors - are *not* resolution limited. Most symbologies can be decoded with a resolution of 50 pixels per cm on the target, meaning that a solid state LF scanner producing a 1 megapixel refocused image can decode a 10 mil data matrix from 0 to 29 cm, a range which currently requires the use of three separate fixed-focus devices.

4.4 Overview and Assumptions of Approach

Figure 4.1 shows an overview of our approach. Starting from the raw image captured by a sensor behind a microlens array, we sub-sample one pixel per microlens to get a sub-aperture image within which we locate the barcode. We then crop out the corresponding region of the raw sensor image, which gives a LF covering only the barcode. Using the pixels in this region, we estimate the depth (the distance from the

[1] Handheld scanners are typically designed to withstand multiple drops from 2m onto a concrete floor.

[2] Not only is the effectiveness of active illumination reduced at longer distances, but illumination is limited by the fact that batteries or USB connections limit the available power.

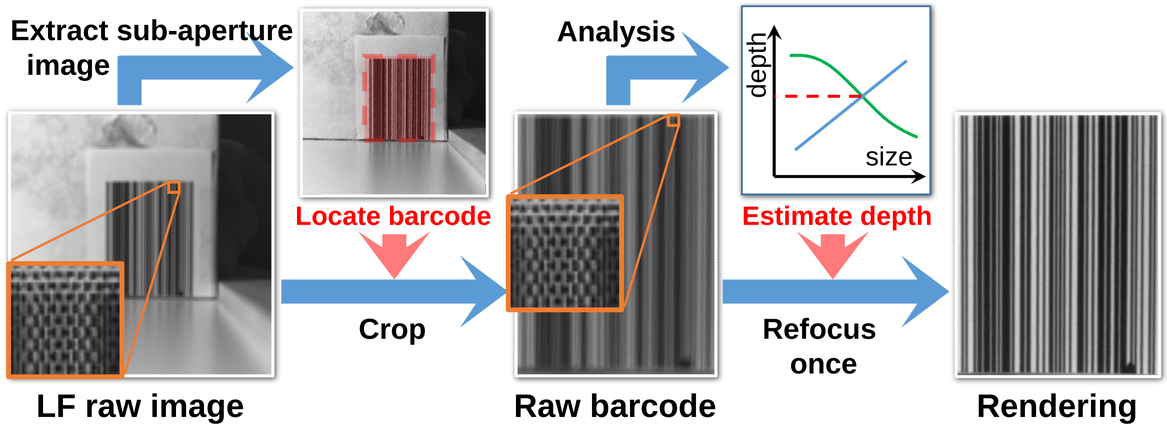


Figure 4.1: Our overall system for barcode refocusing.

lens to the barcode) and then refocus a single image which sharply renders the target. The sharply-focused barcode region is then supplied to a decoder which determines both the symbology and the barcode's contents.

While barcodes in the real world may appear at any orientation relative to the sensor, our analysis assumes that the barcode is approximately frontal parallel, and we only consider one depth value. In practice, slanted barcodes will often appear sufficiently well-focused, though effective scanning in these situations is limited by decoder performance on skewed inputs.

4.5 Barcode Depth Estimation

The key to fast rendering of the refocused barcode image is depth estimation. While existing methods [18, 52, 105, 113, 30] are applicable to general scenes, the expensive computational cost prohibits them from being used in time-sensitive scenarios like barcode scanning. In this section, we speed up the process by first segmenting out the barcode region, and then analyzing the statistics of pixel intensities in the lenslet image. Compared with traditional depth estimation methods, our approach is application specific and much faster.

Figure 4.2: Raw LF image. Note that lenslet image pattern changes with the depth of the barcode.

4.5.1 Barcode Localization

Conventional barcodes are composed of high contrast black and white bars or patches, which facilitate the localization process. Several approaches have been proposed and optimized to take advantage of the texture information for localization, but barcode detection is still an active area of research [15] for traditional cameras. However, the imaging mechanism of LF camera will distort and deteriorate these features, making existing approaches less effective, even unusable. The structure of LF camera is similar to a conventional camera, except that it adds a microlens array in front of the sensor to further diverge the rays based on their directions. Thus, the resultant raw LF image consists of hundreds of thousands of lenslet images, as shown in Fig. 4.2. Directly locating the barcode on the raw LF image would be extremely challenging: each lenslet image contains a small number of pixels (*e.g.*, 10×10 in Lytro camera);

and the high contrast in the boundary region of lenslet image will fail gradient based detection algorithms.

In order to address these issues, we aim to first localize the barcode on a sub-aperture image instead of the raw image. A sub-aperture image is a 2D image composed of pixels at the same position beneath each microlens. It can be regarded as an image taken by a virtual camera with its center of projection on the main lens. In our case, we pre-calibrate the center of each lenslet image and pick the center pixels to generate a central sub-aperture image. Interpolation is required since the lenslet arrangement is hexagonal.

Although the sub-aperture image is of low resolution (about 328×378 for Lytro) which inhibits direct decoding, it is detailed enough for barcode localization. We extend the method proposed in [34] by incorporating the barcode orientation into the feature computation, and analyze the shape of the region with high average feature responses for robust localization. For each angle $\theta \in \{-90, -85, ..., 90\}$, feature response $I_e^\theta(p) = |I_{x_\theta}(p)| - |I_{y_\theta}(p)|$ is evaluated at each pixel p, where $I_{x_\theta}(p)$ and $I_{y_\theta}(p)$ are the image gradient along orthogonal directions $x_\theta(\cos\theta, \sin\theta)$ and $y_\theta(-\sin\theta, \cos\theta)$ respectively. A box filter is applied to I_e^θ to get locally averaged feature response $\bar{I}_e^\theta$. The potential barcode region is identified by a connected region of constantly high average response $\bar{I}_e^{\theta^*}$ with θ^* maximizing the mean of $\bar{I}_e^\theta(p)$'s within the region. The shape of this region is also required to be tightly bounded by an oriented rectangle. Within this rectangle, we compute the size of the candidate barcode as the distance between the first and the last black bars. In order to eliminate the effects of illumination variations, the input sub-aperture image is preprocessed using local histogram equalization. Fig. 4.3 shows an example of our barcode localization algorithm.

Note that our localization method is designed for 1D barcode. We refer the reader to [121] and other related work for 2D barcode localization. After we locate the barcode in the sub-aperture image, we can continue to crop the corresponding barcode region in the raw light field image and only process this region to speed up our following ray tracing algorithm.

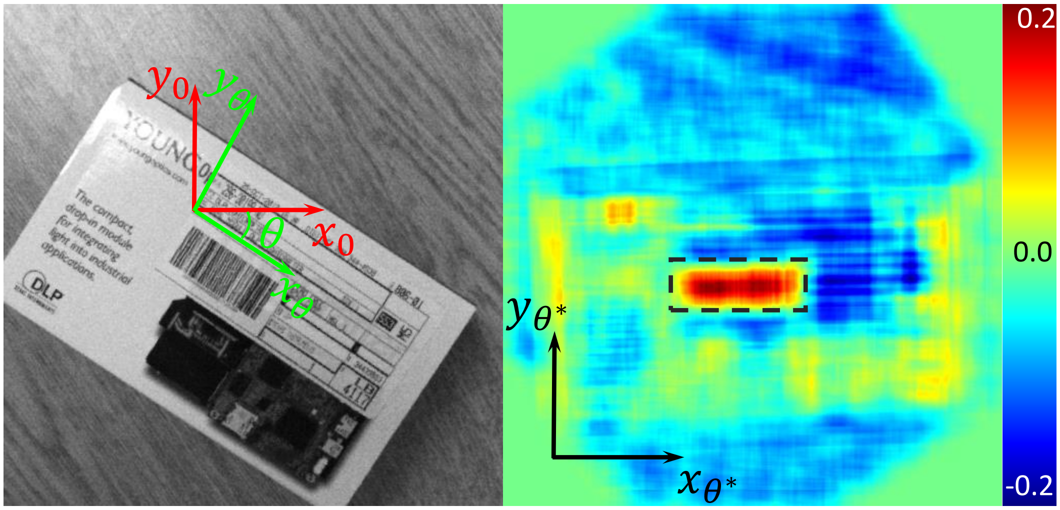


Figure 4.3: A barcode localization example. An optimal rotation angle θ^* is determined maximizing the mean feature response of the potential barcode region.

4.5.2 Spatial Frequency *vs.* Depth

We first study the correlation between the spatial frequency of the raw barcode region and its depth. Here we assume that the barcode is approximately frontal parallel to the camera so we only consider one depth value. As shown in Fig. 4.2, barcodes positioned at different depth exhibits different lenslet image patterns. In the first inset, each lenslet image shows uniform color, indicating the image plane of the main lens coincides with the plane of the microlens array. As the barcode moves nearer to the camera, increasing intensity variations are evident in lenslet images. Therefore, our intuition is to use this statistical characteristics of barcode for depth estimation.

To better illustrate our algorithm, we simplify the barcode as evenly distributed black and white bars. The spatial frequency of the barcode is defined as the number of line pairs per unit length. Fig. 4.4 shows two cases of formation of lenslet images. In the first case, the image plane of the main lens falls in front of the microlens array, where each lenslet image is a real image. On the contrary, when the image plane is behind

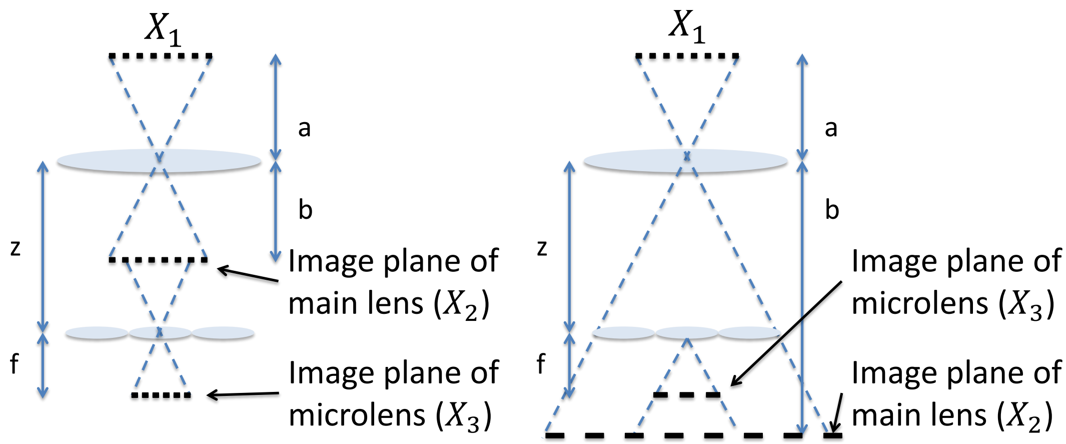


Figure 4.4: Spatial frequencies of the barcode image at different image planes.

the microlens array, a virtual image will be observed. Given the spatial frequency of the barcode X_1, we apply thin lens equation to compute the spatial frequency at the image plane of the main lens $X_2 = \frac{a}{b} \cdot X_1 = \frac{a-F}{F} \cdot X_1$, where a is the object distance and F is the focal length of the main lens. We repeat this process to obtain the spatial frequency of the barcode image at the sensor plane $X_3 = \frac{z-b}{f} \cdot X_2 = \frac{a(z-F)-zF}{Ff} \cdot X_1$ when the main lens image plane is in front of the microlens and $X_3 = \frac{z-b}{f} \cdot X_2 = \frac{a(F-z)+zF}{Ff} \cdot X_1$ when the image plane is behind the microlens. Here z represents the distance between the main lens and the microlens, b is the image distance and f is the focal length of the microlens. In both cases, a linear relationship between the barcode's spatial frequency at the sensor and its depth can be observed.

4.5.3 Variance *vs.* Depth

Although we can mathematically compute the sensor plane's spatial frequency X_3, it is very challenging to robustly measure this frequency since each lenslet image is only of size 10×10 pixels–*i.e.* a very small portion of the barcode, with its boundary region corrupted by vignetting. In our experiments, we observe at most two color transitions inside each lenslet image. Therefore, we instead use variance to represent the spatial frequency of each lenslet image. Specifically, we define a window around

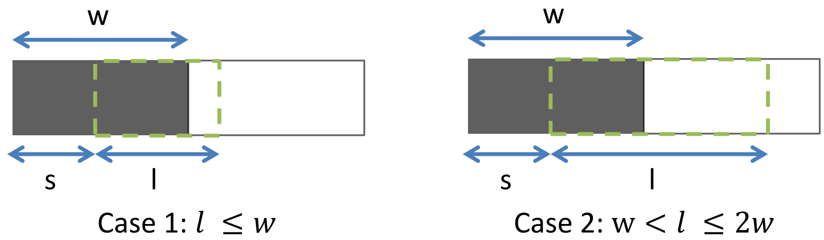


Figure 4.5: Lenslet images function as a sliding window across the barcode region.

each lenslet center and measure the variance of pixel intensities within the window. Our intuition is that the higher the spatial frequency, the larger the chance to observe intensity transitions inside the window. We then compute the overall variance as the spatial frequency measurement by averaging the variances from the lenslet images inside the barcode region.

To formulate the correlation between variance and depth, we make following assumptions based on the observation that at most two intensity transitions appear within each lenslet image. Next, we regard the light field camera as a relay imaging system, which consists of mainlens and microlenses as pinhole cameras. We first analyze the image captured by the microlens, then extend our analysis to the whole system.

First we want to define variance σ^2. Suppose our target is evenly distributed black/white bars. Our pinhole camera has N pixels and the captured image contains m white pixels and n black pixels. And we further denote the intensity of the white pixel as 1 and that of the black pixel as 0. Then we can get

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 = \frac{mn}{(m+n)^2} \tag{4.1}$$

Next we only consider the lenslet image. As each lenslet image only observes a very small portion of the barcode, its variance changes with its relative positions with the bar. As shown in Fig. 4.5, we denote the bar width of the image as w, the sensor size at the barcode image plane as l and the distance between the starting point of the lenslet image and a intensity transition as s. Then we continue our analysis in two cases:1) If $l \leq w$, then

$$\sigma^2 = \begin{cases} 0 & s \leq w - l \\ \frac{-s^2+(2w-l)s+lw-w^2}{l^2} & w - l < s \leq w \end{cases} \tag{4.2}$$

We only compute the variance σ^2 as a function of s ranging from 0 to w because it is a periodic function. Since the lenslets are hexagonally arranged, their images can be considered as a sliding windows across the entire barcode image. From the distribution of σ^2, we can get the average variance $\bar{\sigma}^2$ as:

$$\bar{\sigma}^2 = \frac{\int_0^w \sigma^2 ds}{w} = \frac{1}{w}(\int_0^{w-l} \sigma^2 ds + \int_{w-l}^{w} \sigma^2 ds) = \frac{l}{6w} \tag{4.3}$$

It is evident that average variance $\bar{\sigma}^2$ is linearly relates to l. We can further map l through the mainlens to the real barcode as L. By using similar triangles, we have $L = \frac{al}{b} = \frac{A}{Ff}[(z-F)a - zF] or \frac{A}{Ff}[(F-z)a + zF]$ and $l = \frac{A(z - \frac{a-F}{aF})}{f}$, where A is the size of the sensor and a, b, F, f, z are defined in last section. Therefore, each lenslet image covers an area of l on the barcode image through mainlens, and an area of L on the real barcode. Because l increases monotonically with the increase of a, we can obtain a one-on-one mapping between the depth a and average variance $\bar{\sigma}^2$.

2) if $w < l \leq 2w$, we have

$$\sigma^2 = \begin{cases} \frac{-s^2+(2w-l)s+lw-w^2}{l^2}, & 0 < s \leq 2w - l \\ \frac{lw-w^2}{l^2}, & 2w - l < s \leq w \end{cases} \tag{4.4}$$

Similarly, we compute its average variance $\bar{\sigma}^2$ as:

$$\bar{\sigma}^2 = \frac{\int_0^w \sigma^2 ds}{w} = \frac{1}{w}(\int_0^{2w-l} \sigma^2 ds + \int_{2w-l}^{w} \sigma^2 ds) = \frac{w^2}{3}l^{-2} - \frac{1}{6w}l - wl^{-1} + 1 \tag{4.5}$$

To prove $\bar{\sigma}^2$ monotonically increases with l, we compute its first and second order derivative as $(\bar{\sigma}^2)' = -\frac{2w^2}{3}l^{-3} - \frac{1}{6w} + wl^{-2}$ and $(\bar{\sigma}^2)'' = 2w^2 - 3wl$. Since $w < l \leq 2w$, $(\bar{\sigma}^2)'' < 0$. We further examine the value of $(\bar{\sigma}^2)'$ at $l = w$ and $l = 2w$, they are both larger than 0. Therefore, we can prove that $(\bar{\sigma}^2)' > 0$, so $\bar{\sigma}^2$ monotonically increases with l. Similar to the first case, we can also obtain a one-to-one mapping between the depth and average variance.

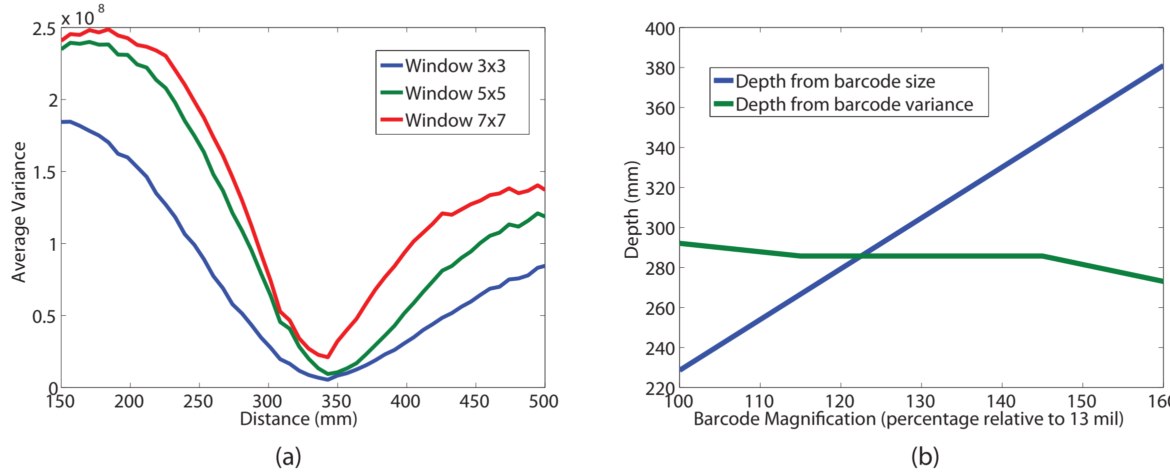


Figure 4.6: (a) The average variances of the barcode image using different window sizes *vs.* its depth. (b) The depth of the barcode region is determined jointly by the variance and the size of the detected barcode region.

4.6 Efficient Refocusing

Our analysis above reveals that we can quickly use the variance to determine the depth of the barcode. This allows us to conduct refocusing with high efficiency.

4.6.1 Barcode Depth Estimation

To validate our use of variance as a depth cue, we measure the average variance of a randomly selected UPC-A barcode over a range of distances from the camera. Fig. 4.6(a) shows the results using different window sizes for variance computation. Clearly we can see valley shaped curves with two approximately linear regions. The bottom of the curve indicates the main lens image plane falls on the microlens, so the lenslet image gets uniform intensity which results in a minimum overall variance. Here one variance value may correspond to two different depths. To resolve this two-fold ambiguity, we only use the left linear region in our experiments, as barcodes of practical sizes at depths in the right linear side are resolution limited even when properly focused. If necessary, the right linear side can be used similarly to estimate another depth in the case that the depth from the left side leads to an undecodable result. Notice that due to defocus blur and resolution limitation [35] in the lenslet image, the curve fluctuates in both ends, making these regions unusable. For robustness reasons, we estimate three depth values independently based on different window sizes 3×3, 5×5 and 7×7, and compute the mean of the corresponding depths as the final estimation.

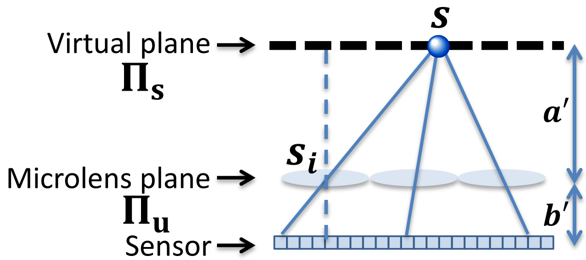
The variance *vs.* depth curve in Fig. 4.6(a) is for standard 13 mil barcodes. Scaling the size of the overall barcode will change the underlying spatial frequency X_1, and change the relationship between depth and variance. This is inevitable since product manufacturers tend to adjust the size of the barcode to suit the package. Our solution is first to build a look-up table indexed by variances per barcode size. Then we jointly determine the final depth based on both the variance and the size of the detected barcode region in the central sub-aperture image. From projective geometry, we obtain the relationship between the barcode image size s and the depth d as $s \propto S/d$, where S is the original size of the barcode. Fig. 4.6(b) illustrates our depth determination strategy. Given a detected barcode size, the larger the barcode's original size, the further its distance. Given a measured variance, another size *vs.* depth curve is formed by collecting depths from the look-up tables for corresponding barcode sizes. The ground truth original barcode size and the depth are therefore indicated by the intersection of these two lines/curves.

4.6.2 Refocusing

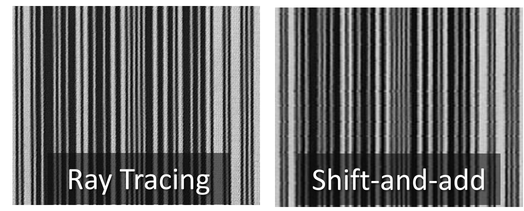
The final step in our light field barcode imaging system renders a focused image of the barcode region, using the depth estimated from the variance and size of this region. We set out to perform ray tracing to generate the in focus barcode image. Ray tracing mimics the physical process of image formation. The intensity of a point on the target image plane (virtual plane) is computed by integrating all the rays of different directions passing through it.

As shown in Fig. 4.7(a), we use two parallel plane parameterization (2PP) [62] to represent a ray. Then the formation process of the target image I' can be summarized as:

$$I'(\mathbf{s}) = \sum_i I((\mathbf{s}_i - \mathbf{s})\frac{b'}{a'} + \mathbf{s}_i), \tag{4.6}$$



(a) Ray tracing



(b) Results of ray tracing and Shift-and-add

Figure 4.7: (a)High quality barcode rendering by ray tracing. (b)Results from two implementations of refocusing algorithm.

here $\mathbf{s}_i$ denote the location of the optical center of lenslet, a' the distance from target image plane $\Pi_{\mathbf{s}}$ to the microlens plane $\Pi_{\mathbf{u}}$ and b' the distance from $\Pi_{\mathbf{u}}$ to the sensor plane, and I is the raw image on the sensor.

In our experiments, we first adopt the method proposed by [19] and use pre-loaded white images from Lytro camera to locate the lenslet centers $\mathbf{s}_i$ according to the camera's focal length setting. The target image plane is then determined based on the estimated depth and is discretized into pixels. Next we conduct ray tracing for each pixel $\mathbf{s}$ to gather the recorded irradiance of the rays and apply bilinear interpolation to achieve a better approximation of the pixel value. Notice that there is a tradeoff between the resolution of the barcode image and its computational cost. The ray tracing technique provides the flexibility to vary the resolution by simply changing the sampling rate on the virtual plane. In our experiments, we render a barcode image of approximately 200×200 pixels to balance these two factors. Compared to the shift-and-add refocusing algorithm in [84], which requires rectified light field images (lenslet images arranged on grids), our method produces sharper rendering results as shown in Fig. 4.7(b). The blur artifacts in the shift-and-add result are due to the interpolation operation conducted when generating the rectified light field image from Lytro data. Generating images with even higher quality is still possible [126, 113], but impractical due to its high computational cost.

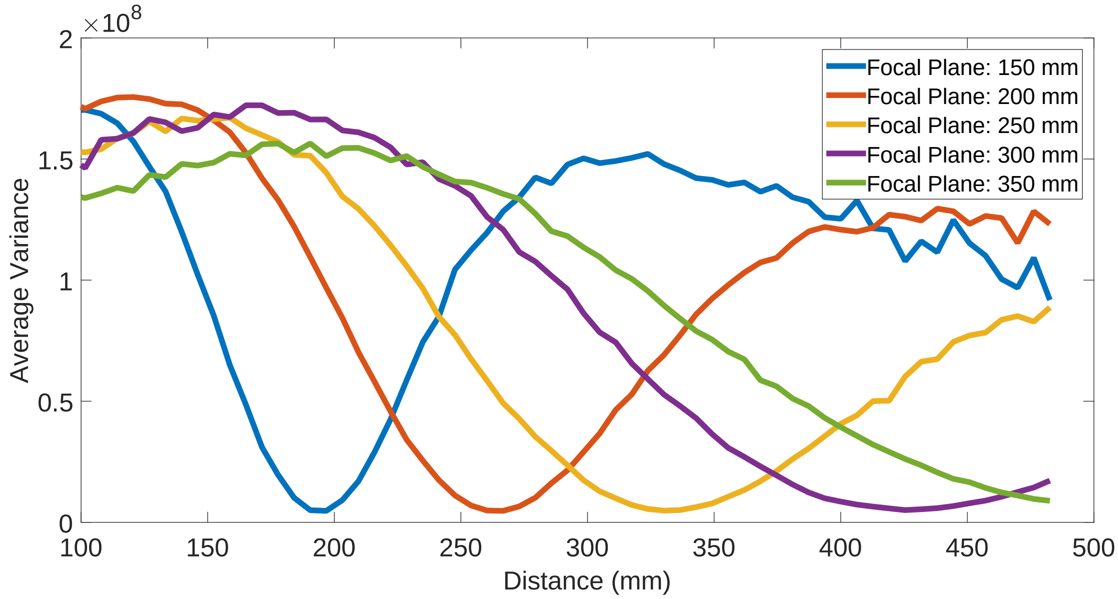


Figure 4.8: We progressively move the LF camera's main focus plane from 150 mm to 350 mm with an incremental of 50 mm and plot the depth *vs.* variance curve with window size 3×3.

4.7 Experiments

We use Lytro camera as our prototype light field camera. The raw images are preprocessed according to the metadata from Lytro's proprietary file format [19] and the vignetting effects are removed using the pre-stored calibration images in Lytro camera. Demosaicing is then applied to get the final raw light field image. While capturing, we keep both the focal length and focal plane unchanged to simulate a light field camera without active parts.

Focal Plane Determination Although the main focus plane of the LF camera is unchanged during capture, we still need to investigate its impact on the "depth *vs.* variance" curve and find the optimal focal plane. To this end, we keep all other settings unchanged and click Lytro's interface to progressively move the focal plane away from the camera. Fig. 4.8 shows that position of the minimum value changes with its focal plane, as well as the slope of the curve. A tradeoff exists between the depth resolution and effective range: smaller focal plane corresponds to larger slope and higher depth

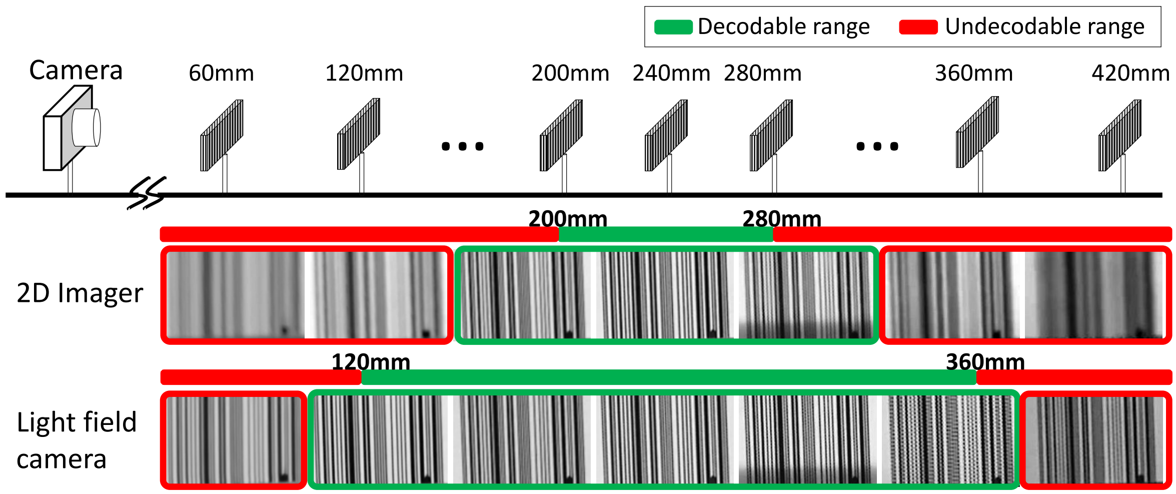


Figure 4.9: Barcode images captured at variant depths using different devices. Light field camera largely extends the decodable range while keeping the noise level low.

resolution, but it suffers from smaller effective range. This conclusion infers that the optimal focal plane is application specific. In our experiments we set the focal plane to 250 mm for its moderate range and depth resolution.

Depth of Field Our first experiment is to determine the amount of extended depth of field the light field camera has over a conventional camera. We collect a set of images of the barcode positioned at 60 mm to 420 mm from the camera with an incremental step of 6.9 mm. Using Lytro's desktop application, we generate two groups of images using the same focal length and aperture size: 1) one with focal plane coincides with the moving barcode and 2) the other one with a fixed focal plane simulating the conventional scanner. We test the decodability of the barcode images with a proprietary decoder. Results show that images from the conventional camera are only decodable within a range of 80 mm due to the defocus blur. On the contrary, the images from light field camera feature extended depth of field, with a decodable region of 240 mm, which nearly triples the range of the conventional camera. Fig. 4.9 shows the comparison of the decodable range of 2D scanner and the light field camera, as well as the sharpness of their resultant images.

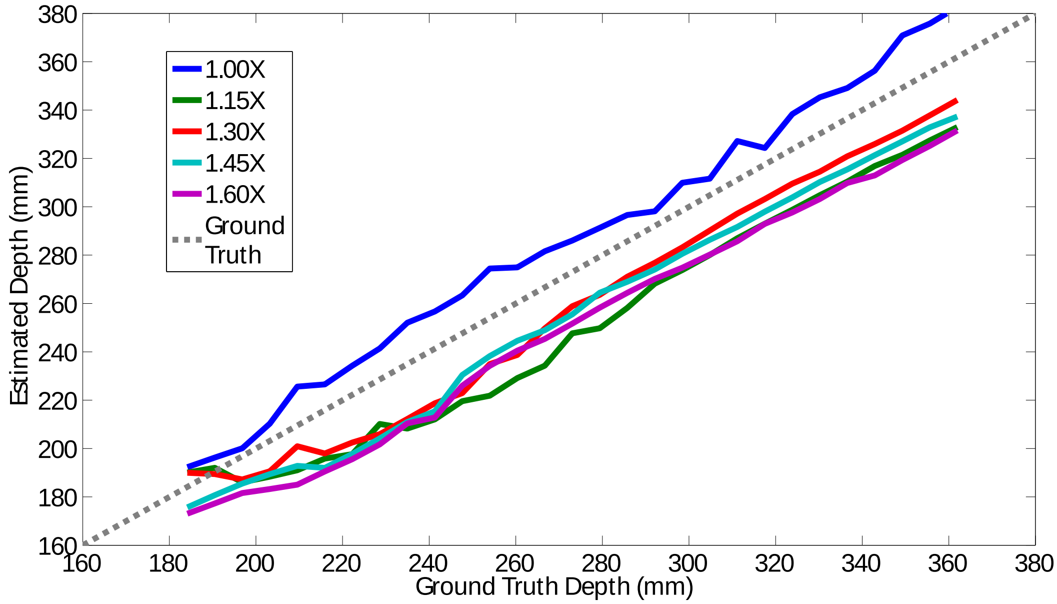


Figure 4.10: Comparison between measured depths and the ground truth depths for barcodes of different sizes.

Depth Estimation and Image Rendering Our subsequent experiments are to validate our barcode localization and depth estimation algorithm. We set our recognition target to be the standard 13 mil UPC-A barcode with 1.0x, 1.15x, 1.3x, 1.45x and 1.6x magnifications. Our variance *vs.* depth look up tables and size *vs.* depth curves are calibrated based on training data of random UPC-A codes. Barcodes with codes different from the training data are used for test. Fig. 4.10 shows the comparison between the estimated depths and the ground truth depths for barcodes of different sizes. The estimation errors are less than 60 mm which is within the decodable range. Fig. 4.11 shows our rendering results for barcodes on real products. Notice that our algorithm is robust to different sizes, orientations and nonuniform lighting conditions. However, severe distortions will lead to failure cases as shown in the last result of Fig. 4.11. The main reason for this failure case is that our barcode localization algorithm detects a rectangle rather than a tight parallelogram only encloses the barcode. The non-barcode region inside our rectangle pollutes the variance estimation for depth estimation.

Running time We compare the processing speed/time of our system and a 2D scanner.

Figure 4.11: Rendering results of real barcodes using our scanning system. The full image on the left of each barcode example is the in focus image at the ground truth depth. We also show an example where our algorithm fails due to severe distortion.

A 2D scanner directly locates and decodes the barcode after exposure, while our system requires two extra steps: depth estimation and rendering of the barcode region. In our C++ implementation, the extra steps take around 0.2s for each light field image. Note that the result is not fully optimized. With application-specific integrated circuit (ASIC), as is implemented in most scanners, the overall processing time can be further reduced.

4.8 Discussion

From the experiment, we can conclude that a light field camera could be used to replace current barcode scanner to gain extended depth of field with advanced algorithm and higher light efficiency with its larger aperture. While a purpose-built LF scanner would likely use a smaller aperture than the Lytro camera, our emphasis has been on algorithmic improvements that would apply to such hardware. The core of our algorithm is fast depth estimation of barcode by jointly analyzing the size and statistical characteristics of the barcode. Therefore, only the necessary focal slice will be rendered and decoded. Depending on the size of the barcode in the image, and on the depth complexity of the scene, these improvements can dramatically reduce the amount of time needed to produce a decodable image.

Chapter 5

DEPTH FROM DUAL LIGHT FIELDS

5.1 Background

Human visual system relies on a variety of depth cues to gain 3D perception. The most important ones are binocular, defocus, and motion cues. Binocular cues such as stereopsis, eye convergence, and disparity yield depth from binocular vision through the exploitation of parallax. Defocus cue allows depth perception even with a single eye by correlating variation of defocus blurs with the motion of the ciliary muscles surrounding the lens. Motion parallax also provides useful input to assess depth, but arrives over time and depends on texture gradients.

Computer vision algorithms such as stereo matching [92, 10] and depth-from-focus/defocus [80, 81, 71, 25, 26] seek to directly employ binocular and defocus cues which are available instantaneously without scene statistics. Recent studies have shown that the two types of cues complement each other to provide 3D perception [45]. In this chapter, we seek to develop learning based approaches to emulate this process.

To exploit binocular cues, traditional stereo matching algorithms rely on feature matching and optimization to maintain the Markov Random Field property: the disparity field should be smooth everywhere with abrupt changes at the occlusion boundaries. Existing solutions such as graph-cut, belief propagation [57, 98], although effective, tend to be slow. In contrast, depth-from-focus (DfF) exploits differentiations of sharpness at each pixel across a focal stack and assigns the layer with the highest sharpness as its depth. Compared with stereo, DfF generally presents a low fidelity estimation due to depth layer discretization. Earlier DfF techniques use a focal sweep camera to produce a coarse focal stack due to mechanical limitations whereas more recent ones attempt to use a light field to synthetically produce a denser focal stack.

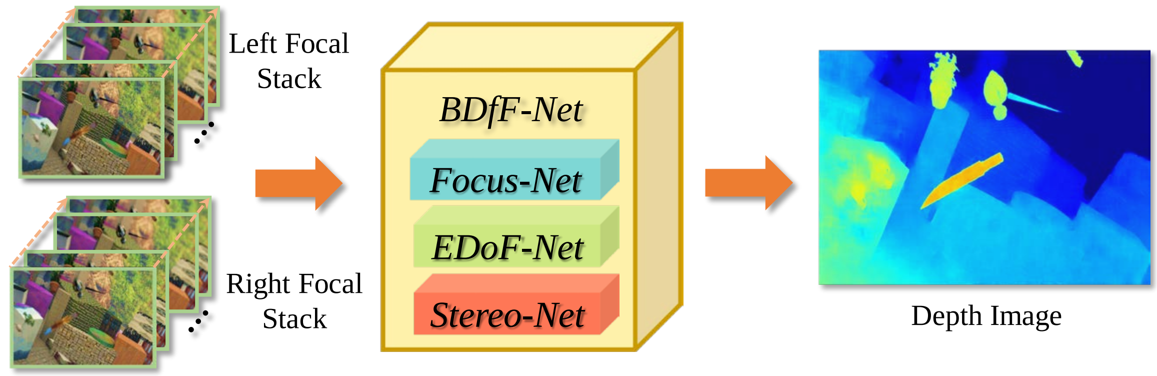


Figure 5.1: *BDfF-Net* integrates *Focus-Net*, *EDoF-Net* and *Stereo-Net* to predict high quality depth map from binocular focal stacks.

Our solution benefits from the recent advance on computational photography and we present an efficient and reliable learning based technique to conduct depth inference from a focal stack pair, emulating the process of how human eyes work. We call our technique binocular DfF or B-DfF. Our approach leverages deep learning techniques that can effectively extract features learned from a large amount of imagery data. Such a deep representation has shown great promise in stereo matching [128, 127, 69]. Little work, however, has been proposed on using deep learning for DfF or more importantly, integrating stereo and DfF. This is mainly due to the lack of fully annotated DfF datasets.

We first construct a comprehensive focal stack dataset. Our dataset is based on the highly diversified dataset from [73], which contains both stereo color images and ground truth disparity maps. Then we adopt the algorithm from *Virtual DSLR* [123] to generate the refocused images. [123] uses color and depth image pair as input for light field synthesis and rendering, but without the need to actually create the light field. The quality of the rendered focal stacks is comparable to those captured by expensive DSLR camera. Next, we propose three individual networks: (1) *Focus-Net*, a multi-scale network to extract depth from a single focal stack (2) *EDoF-Net*, a deep network consisting of small convolution kernels to obtain the extended depth of field

(EDoF) image from the focal stack and (3) *Stereo-Net* to obtain depth directly from a stereo pair. The EDoF image from *EDoF-Net* serves to both guide the refinement of the depth from *Focus-Net* and provide inputs for *Stereo-Net*. We also show how to integrate them into a unified solution *BDfF-Net* to obtain high quality depth maps. Fig. 5.1 illustrates the pipeline.

We evaluate our approach on both synthetic and real data. To physically implement B-DfF, we construct a light field stereo pair by using two Lytro Illum cameras. Light field rendering is then applied to produce the two focal stacks as input to our framework. Comprehensive experiments show that our technique outperforms the state-of-the-art techniques in both accuracy and speed. More importantly, we believe our solution provides important insights on developing future sensors and companion 3D reconstruction solutions analogous to human eyes.

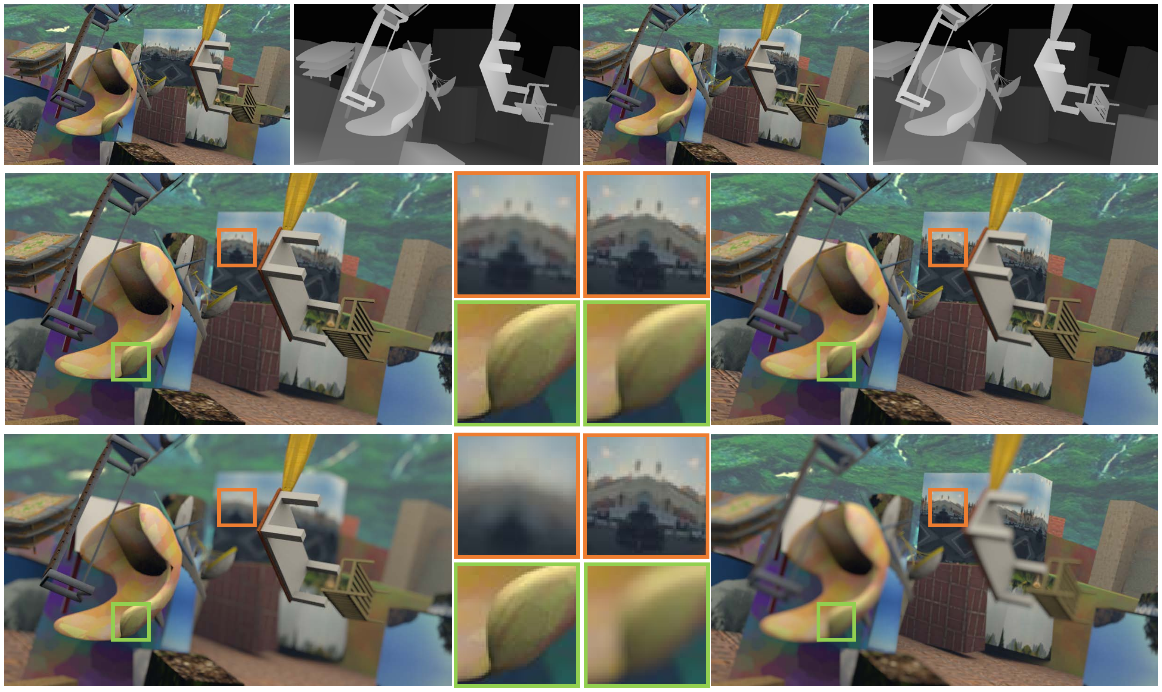
5.2 Dual Focal Stack Dataset

With fast advances in the data driven methods, numerous datasets have been created for various applications. However, by far, there are limited resources on focal stacks. To this end, we generate our dual focal stack dataset based on FlyingThings3D from [73]. FlyingThings3D is an entirely synthetic dataset, consisting of everyday objects flying along randomized 3D paths. Their 3D models and textures are separated into disjointed training and testing parts. In total, the dataset contains about 25,000 stereo images with ground truth disparity. To make the data tractable, we select stereo frames whose largest disparity is less than 100 pixels, then we normalize the disparity to $0 \sim 1$.

Takeda *et al.* [104] demonstrate that in stereo setup, the disparity d and the diameter of the circle of confusion c have a linear relationship:

$$\frac{d}{c} = \frac{l}{D} \tag{5.1}$$

where l is the baseline length and D is the aperture size. Based on above observation, we adopt the *Virtual DSLR* approach from [123] to generate synthetic focal



(a)

(b)

Figure 5.2: (a) Same scene rendered with varying blur kernel. The top row shows the ground truth color and depth images for the stereo pair. The middle and bottom row are the rendered defocused image, where the bottom row is rendered with a larger kernel than the top row. Left and right columns show images with different focal plane. The insets show the close-up views. (b) Image with/without the Poisson noise. Best viewed in the electronic version by zooming in.

stacks. *Virtual DSLR* requires color and disparity image pair as inputs, and outputs refocused images with quality comparable to those captured from regular, expensive DSLR. The advantage of their algorithm is that it resembles light field synthesis and refocusing but does not require actual creation of the light field, hence reducing both memory and computational load. Further, the *Virtual DSLR* takes special care of occlusion boundaries, to avoid color bleeding and discontinuity commonly observed on brute-force blur-based defocus synthesis.

To better explain their approach, we list the formulation as below:

$$C_p = \frac{|s - s_p|}{s_p} D = sD|\frac{1}{z_p} - \frac{1}{z_s}|, \tag{5.2}$$

To simulate a scene point p with depth z_p projected to a circular region on the sensor, we assume the focal length f, an aperture size D, sensor to lens distance s,and the circular region diameter C_p. Here $z_s = (1/f - 1/s)^{-1}$ and $s_p = (1/f - 1/z_p)^{-1}$ according to the thin lens law. The diameter of the circular region C_p measures the size of the blur kernel and it is linear to the absolute difference of the inverse of the distances z_p and z_s. For the scope of this paper, we use only circular apertures, although more complex ones can easily be synthesized. To emulate the pupil of the eye in varying lighting conditions, we randomly select the size of the blur kernel for each stereo pair but limit the largest diameter of the blur kernel to 31 pixels. We also evenly separate the scene into 16 depth layers and render a refocused image for each layer. After generating the focal stacks, we add Poisson noise to the images to simulate the real image captured by a camera. This turns out to be critical in real scene experiments, as described in section 5.5.2. Finally, we split the generated dual focal stacks into 750 training data and 70 testing data. Figure 5.2 shows two slices from the dual focal and their corresponding color and depth image.

5.3 B-DfF Network Architecture

Convolutional neural networks are very efficient at learning the non-linear mapping between the input and the output. Therefore, we aim to take an end-to-end approach to predict a depth map. [94] shows that a deep network with small kernels is very effective in image recognition tasks. Although a small kernel has limited spatial support, a deep network by stacking multiple layers of such kernels could substantially enlarge the receptive field while reducing the number of parameters to avoid overfitting. Therefore, a general principle in designing our network is to use deep architecture with small convolutional kernels.

As already mentioned, the input to the neural network is binocular focal stacks. Therefore, we name our network binocular depth from focus net, or *BDfF-Net*. To extract depth from defocus and disparity respectively, *BDfF-Net* is composed of three individual networks. We start in section 5.3.1 by describing the *Focus-Net*, a multi-scale network that estimates depth from a single focal stack. Then in section 5.3.2 we show that the result can be further enhanced by the extended depth of field images from *EDoF-Net*. Finally we combine *Stereo-Net* and *Focus-Net* in 5.3.3 to infer high quality depth from binocular focal stacks.

5.3.1 FocusNet for DfF/DfD

Motivated by successes from multi-scale networks, we propose *Focus-Net*, a multiscale network to extract depth from a single focal stack. Specifically, *Focus-Net* consists of four branches of various scales. Except for the first branch, other branches subsample the image by using different strides in the convolutional layer, enabling aggregation of information over large areas. Therefore, both the high-level information from the coarse feature maps and the fine details could be preserved. At the end of the branch, a deconvolutional layer is introduced to upsample the image to its original resolution. Compared with the traditional bicubic upsampling, deconvolution layer automatically learns upsampling kernels that are better suited for the application. Finally, we stack the multi-scale features maps together, resulting in a concatenated per-pixel feature vector. The feature vectors are further fused by layers of convolutional networks to predict the final depth value.

An illustration of the network architecture is shown in Fig. 5.3. We use 3×3 kernels for most layers except those convolutional layers used for downsampling and upsampling, where a larger kernel is used to cover more pixels. The spatial padding is also applied for each convolution layer to preserve the resolution. Following [94],

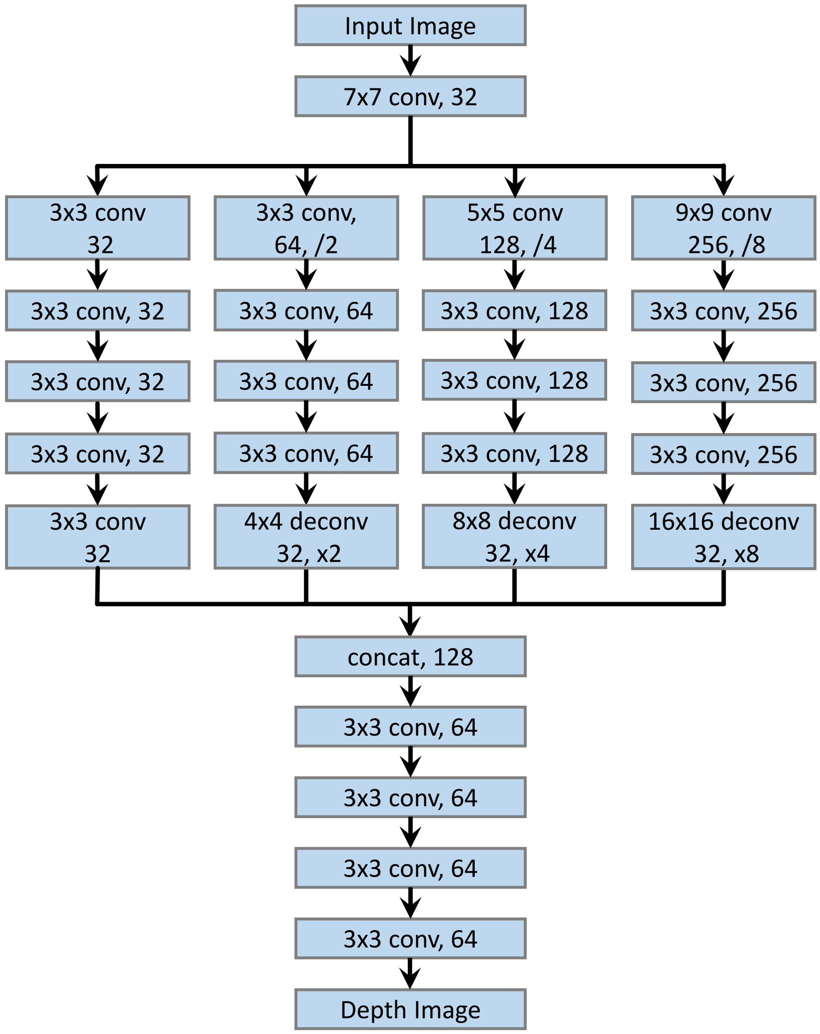


Figure 5.3: *Focus-Net* is a multi-scale network for conducting depth-from-focus.

the number of feature maps increases as the image resolution decreases. Between the convolutional layers we insert PReLU layer [40] to increase the network's nonlinearity. For the input of the network we simply stack the focal stack images together along the channel's dimension.

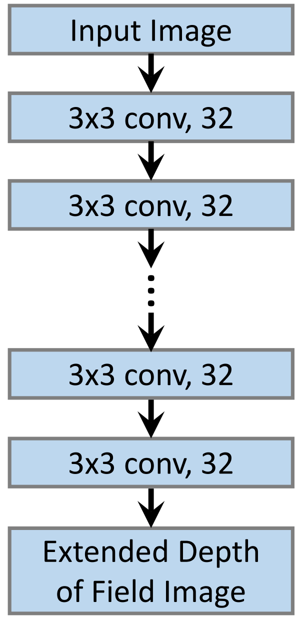


Figure 5.4: *EDoF-Net* consists of 20 layers of convolutional layers to form an extended depth-of-field (EDoF) image from focal stack.

5.3.2 Guided Depth Refinement by EDoF Image

There exist many approaches [29, 47] to refine/upsample depth image with the guidance of an intensity image. The observation is that homogeneous texture regions often correspond to homogeneous surface parts, while depth edges often occur at high intensity variations. With this in mind, we set out to first extract the EDoF image from the focal stack, then guide the refinement of the depth image. Several methods [59, 101] have been proposed to extract the EDoF image from the focal stack. However, the post processing is suboptimal in terms of computational efficiency and elegance. Thus, we seek to directly output an EDoF image from a separate network, which we termed *EDoF-Net*.

EDoF-Net is composed of 20 convolutional layers, with PRelu as its activation function. The input of the *EDoF-Net* is the focal stack, the same as the input of *Focus-Net*, and the output is the EDoF image. With the kernel size of 3×3, a 20 layer convolutional network will produce a receptive field of 41×41, which is larger than the

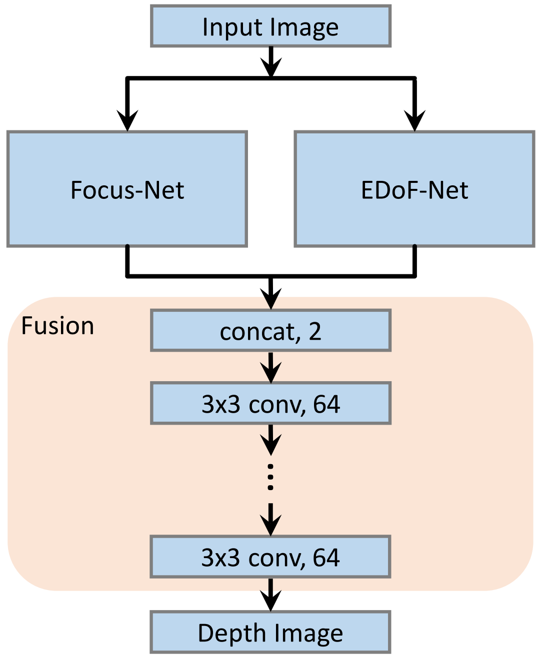


Figure 5.5: Our *Focus-Net-v2* combines *Focus-Net* and *EDoF-Net* by using the EDoF image to refine the depth estimation.

size of the largest blur kernel. Fig. 5.4 shows the architecture of *EDoF-Net*.

Finally, we concatenate the depth image from *Focus-Net* and the EDoF image from the *EDoF-Net*, and fuse them by using another 10 layer convolutional network. We call the new network *Focus-Net-v2*. The architecture of *Focus-Net-v2* is illustrated in Fig. 5.5.

5.3.3 StereoNet and BDfFNet for Depth from Binocular Focal Stack

Given the EDoF stereo pair from the *EDoF-Net*, we set out to estimate depth from stereo using another network, termed *Stereo-Net*. For stereo matching, it is critical to consolidate both local and global cues to generate precise pixel-wise disparity. To this end, we propose *Stereo-Net* by adopting the Hourglass(HG) network architecture [83], as shown in Fig. 6.2. HG network features a contractive part and an expanding part with skip layers between them. The contractive part is composed of convolution layers for feature extraction, and max pooling layers for aggregating high-level information

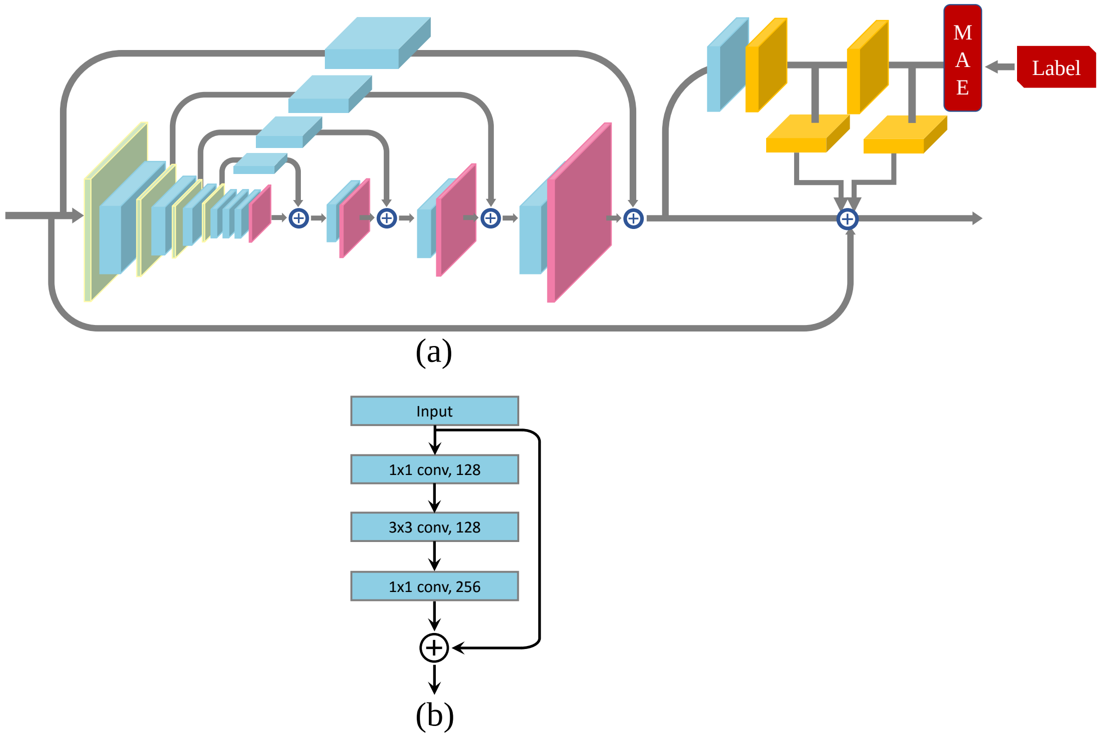


Figure 5.6: (a) *Stereo-Net* follows the Hourglass network architecture which consists of the max pooling layer (green), the nearest neighbor upsampling layer (pink), the residual module (blue), and convolution layer (yellow). The network includes intermediate supervision (red) to facilitate the training process. (b) shows the detailed residual module.

over large areas. Specifically, we perform several rounds of max pooling to dramatically reduce the resolution, allowing smaller convolutional filters to be applied to extract features that span across the entire space of image. The expanding part is a mirrored architecture of the contracting part, with max pooling replaced by nearest neighbor upsampling layer for upsampling. A skip layer that contains a residual module connects each pair of max pooling and upsampling layer so that the spatial information at each resolution will be preserved. Elementwise addition between the skip layer and the upsampled feature map follows to integrate the information across two adjacent resolutions. Both contractive and expanding part utilize a large amount of residual modules [41]. Figure 6.2 shows one HG structure.

One pair of contractive and expanding network can be viewed as one iteration of prediction. By stacking multiple HG networks together, we can further reevaluate and refine the initial prediction. In our experiment, we find a two-stack network is sufficient to provide satisfactory performance. Adding additional networks only marginally improves the results but at the expense of longer training time. Further, since our stacked HG network is very deep, we also insert auxiliary supervision after each HG network to facilitate the training process. Specifically, we first apply 1×1 convolution after each HG to generate an intermediate depth prediction. By comparing the prediction against the ground truth depth, we compute a loss. Finally, the intermediate prediction is remapped to the feature space by applying another 1×1 convolution, then added back to the features output from previous HG network. Our two-stack HG network has two intermediate loss, whose weight is equal to the weight of the final loss.

Different from [83], we do not downsample input images before the first downsampling part. This stems from the difference in problem settings: our solution aims for pixel-wise precision while [83] only requires structured understanding of images. After each pair of downsampling and upsampling parts, supervision is applied using the same ground truth disparity map. The final output is of the same resolution as the input images.

Finally, we construct *BDfF-Net* by concatenating the results from *Stereo-Net*, *Focus-Net-v2* and *EDoF-Net*, and adding more convolutional layers. The convolutional layers serve to find the optimal combination from focus cue and disparity cue. The overall structure of *BDfF-Net* is shown in figure 5.7.

5.4 Implementation

Optimization Given the focal stack as input and ground truth color/depth image as the label, we train all the networks end-to-end. In our implementation, we first train each network individually, then fine-tune the concatenated network with the pre-trained weights as initialization. Because *Focus-Net* and *Focus-Net-v2* contains multiple convolutional layers for downsampling, the input image needs to be cropped

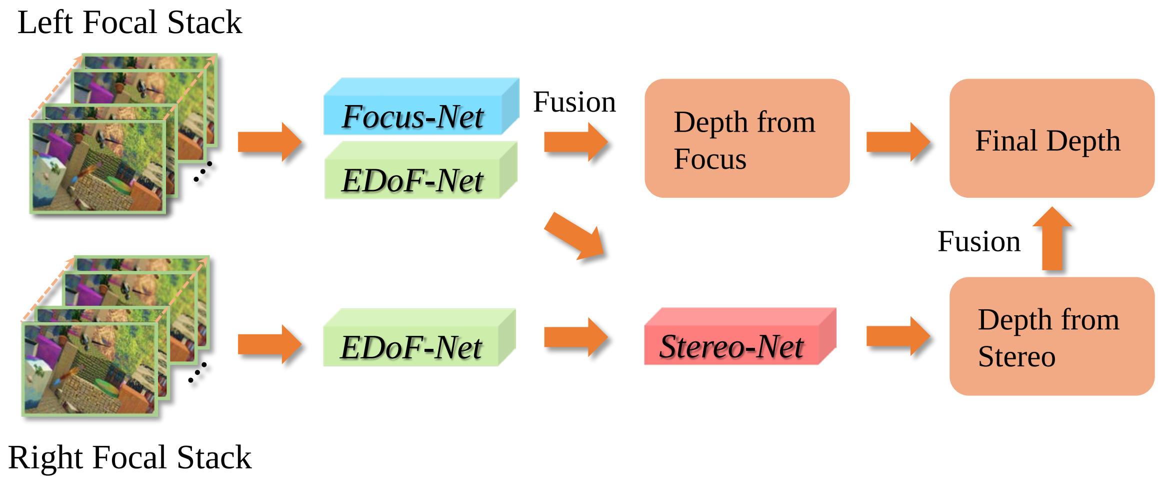


Figure 5.7: The overall structure of *BDfF-Net*.

to the nearest number that is multiple of 8 for both height and width. We use the mean square error (MSE) with l_2-norm regularization as the loss for all models, which leads to the following objective function

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \left\| F(S^i; \theta) - D^i \right\|_2^2 + \frac{\lambda}{2} \left\| \theta \right\|_2^2 \tag{5.3}$$

where S^i and D^i are the i-th focal stack and depth image, $F(S^i; \theta)$ is the function represented by the network and θ are the learned weights. Although there are works [131] suggesting the mean absolute error (MAE) might be a better loss function, our experiment shows that results from MAE are inferior to MSE.

Following [48], we apply batch normalization after the convolution layer and before PRelu layer. We initialize the weights using the technique from [40]. We employ MXNET [13] as the learning framework and train and test the networks on a NVIDIA K80 graphic card. We make use of the Adam optimizer [54] and set the weight decay $= 0.002$, $\beta 1 = 0.9$, $\beta 2 = 0.999$. The initial learning rate is set to be 0.001. All the networks are trained for 80 epochs.

Data augmentation and preprocessing For *Focus-Net* and *EDoF-Net*, the size of

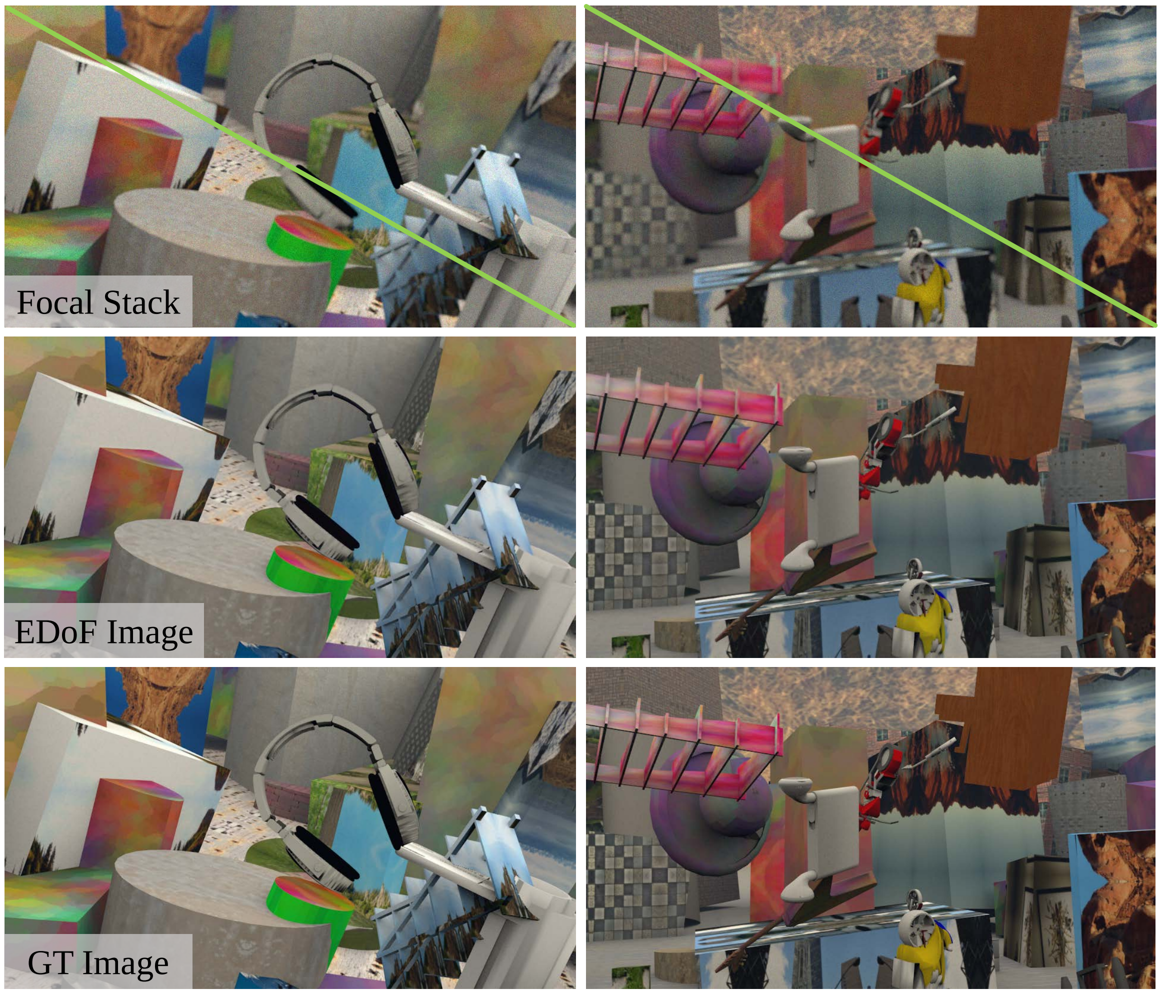


Figure 5.8: Results of our *EDoF-Net*. The upper and lower triangles on the first row show corresponding slices focusing at respective depths. Second and third row show the EDoF and ground truth image respectively.

the analyzed patches determines the largest sensible blur kernel size. Therefore, we randomly crop a patch of size 64×64 from the image, which contains enough contextual information to extract the depth and EDoF image. For *Stereo-Net*, a larger patch of size 256×256 is used to accommodate the large disparity between stereo images. To facilitate the generalization of the network, we augment the data by flipping the patches horizontally and vertically. All the data augmentations are performed on the fly at almost no extra cost. Finally, the range of all images is normalized to $0 \sim 1$.

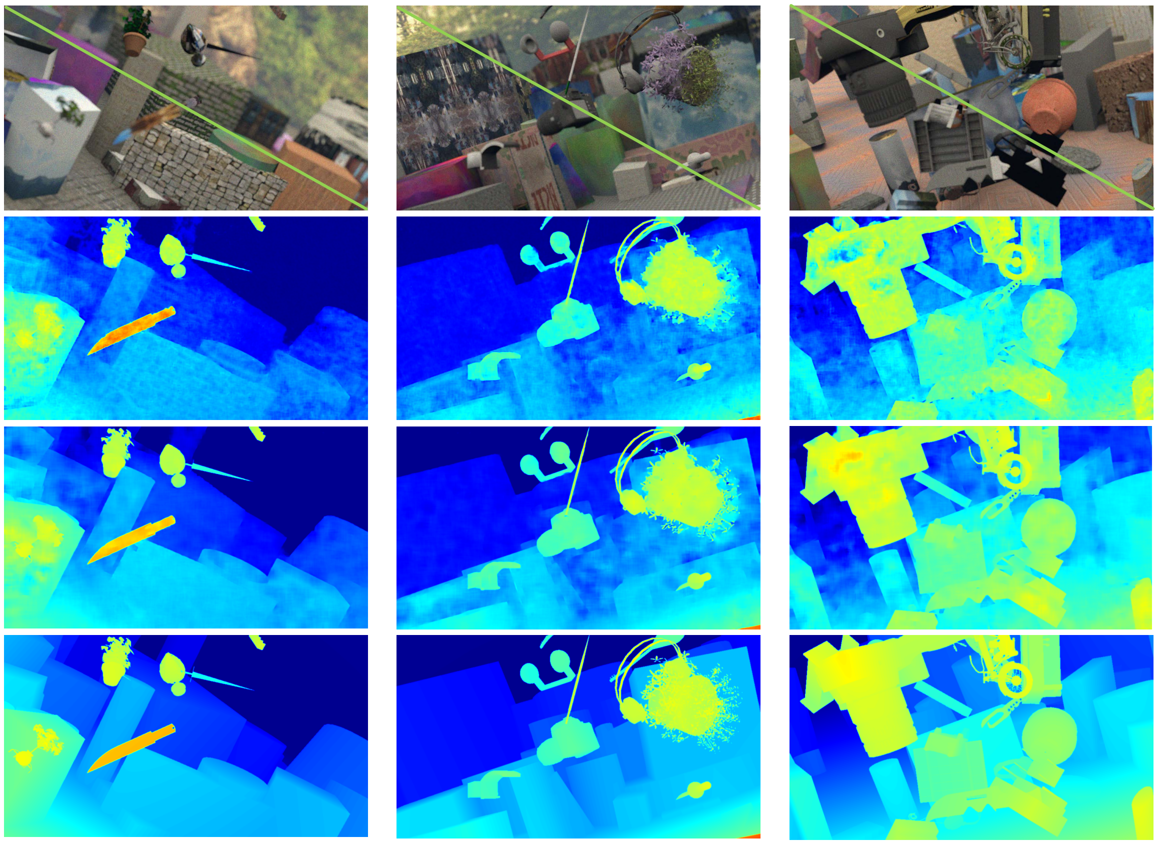


Figure 5.9: Comparisons on *Focus-Net*(second row), *Focus-Net-v2*(third row) and ground truth depth(fourth row), i.e., without and with the guide of an all-focus image.

5.5 Experiments

5.5.1 Extract the EDoF Image from Focal Stack

We train *EDoF-Net* on a single focal stack of 16 slices. Although the network has simple structure, the output EDoF image features high image quality. Our network also runs much faster than conventional methods based on global optimization: on the resolution of 960×540 it runs at 4 frames per second. Fig. 5.8 shows the result of *EDoF-Net*. Compared with ground truth image, the produced EDoF image is slightly blurry. However, given a very noisy focal stack as input, the resultant EDoF image gets rid of large part of the noise. Our experiments also show that it suffices to guide the refinement of the depth image and be used as the input of *Stereo-Net*.

5.5.2 Depth Estimation from Focal Stack

As mentioned in 5.3.2, to construct *Focus-Net-v2*, we first train *Focus-Net* and *EDoF-Net* respectively, then concatenate their output with more fusion layers and train the combination. Fig. 5.9 shows the result of both *Focus-Net* and *Focus-Net-v2*. We observe that *Focus-Net* produces results with splotchy artifacts, and depth bleeds across object's boundary. However, *Focus-Net-v2* utilizes the EDoF color image to assist depth refinement, alleviating the artifacts and leading to clearer depth boundary. It is worth noting that we also trained a network that has identical structure to *Focus-Net-v2* from scratch, but the result is of inferior quality. We suspect this is due to the good initialization provided by the pre-trained model.

We compare our results with [101] and [75] using the data provided by the authors of [101]. We select 16 images from their focal stack for DfF. Fig. 5.10 illustrates the results. Our *Focus-Net-v2* is capable of predicting disparity value with higher quality, while using significantly less time (0.9 second) than [101] (10 minutes) and [75] (4 seconds).

We also train the *Focus-Net-v2* on a clean dataset without Poisson noise. It performs better on synthetic data, but exhibits severe noise pattern on real images, as shown in Fig. 5.11. The experiment confirms the necessity to add noise to the dataset for simulating real images.

5.5.3 Depth Estimation from Stereo and Binocular Focal Stack

Figure 5.12 shows the results from *Stereo-Net* and *BDfF-Net*. Compared with *Focus-Net-v2*, *Stereo-Net* gives better depth estimation. This is expected since *Stereo-Net* requires binocular focal stacks as input, while *Focus-Net-v2* only use a single focal stack. However, *Stereo-Net* exhibits blocky artifacts and overly smoothed boundary. In contrast, depth prediction from *BDfF-Net* features sharp edges. The depth in flat surface region is also smoother compared to *Focus-Net-v2*.

Table 5.1 describes the mean absolute error (MAE) and running time of all models on 960×540 image.

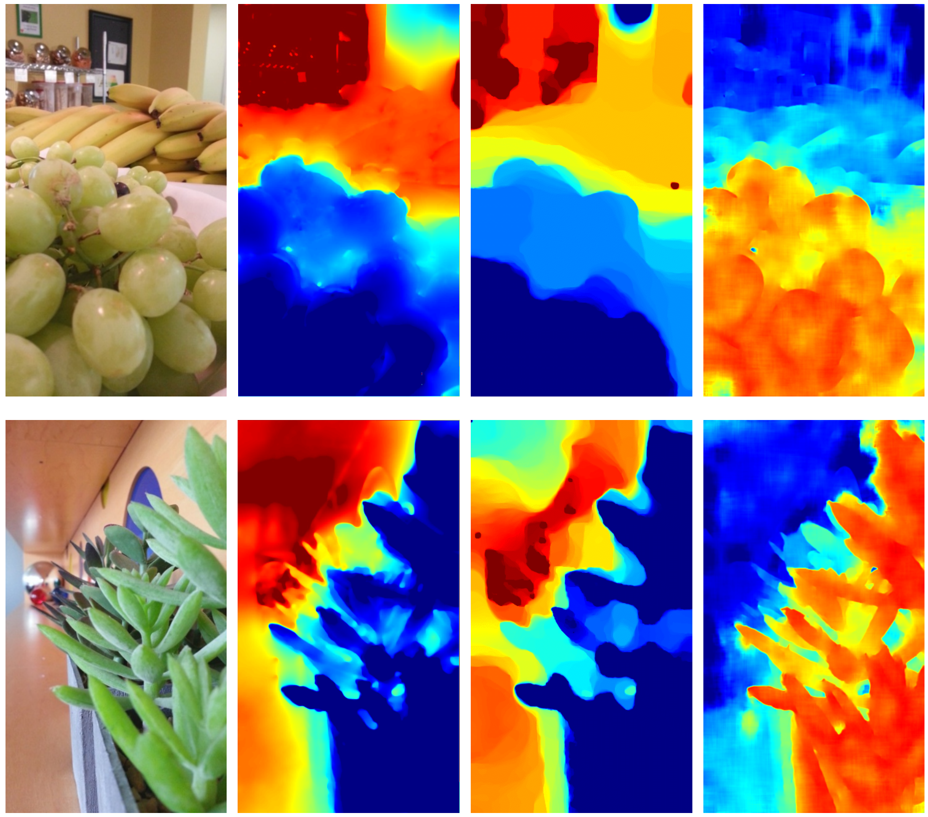


Figure 5.10: Comparisons on depth estimation from a single focal stack using our *Focus-Net-v2* (last column) vs. [101] (second column) and [75] (third column). *Focus-Net-v2* is able to maintain smoothness on flat regions while preserving sharp occlusion boundaries. Note that our approach produces disparity map while [101, 75] generate depth map, thus the colors are flipped.

	Focus-Net	*Focus-Net-v2*	*Stereo-Net*	*BDfF-Net*
MAE	0.045	0.031	0.024	0.021
Time(s)	0.6	0.9	2.8	9.7

Table 5.1: MAE and running time of models.

5.5.4 Real Scene Experiments

We further conduct tests on real scenes. To physically implement B-DfF, we construct a light field stereo pair by using two Lytro Illum cameras, as illustrated in

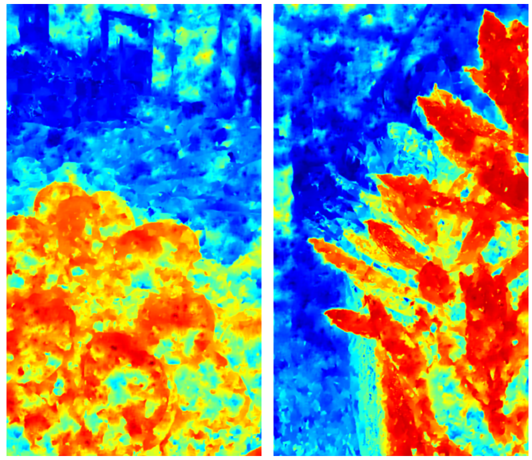


Figure 5.11: Results from *Focus-Net-v2* trained by the clean dataset without poisson noise.

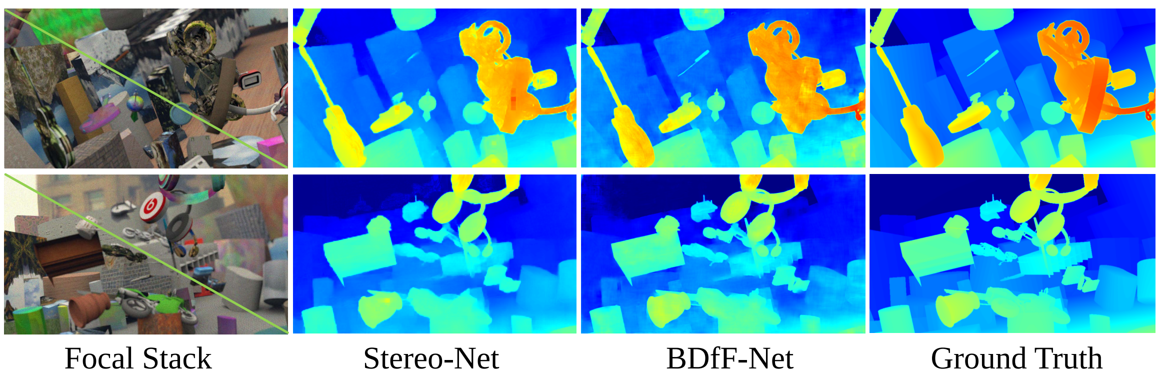


Figure 5.12: Comparisons on results only using *Stereo-Net* vs. the composed *BDfF-Net*. *BDfF-Net* produces much sharper boundaries while reducing blocky artifacts.

Fig. 5.13. Light field camera contains a microlens array to capture multiple views of the scene, allowing users to perform post-capture refocusing. In our experiment the two light field cameras share the same configuration including the zoom and focus settings. The raw images are preprocessed using Light Field Toolbox [19]. Finally we conduct refocusing using shift-and-add algorithm [84] to synthesize the focal stack.

Figure 5.14 shows the predicted depth from *Focus-Net-v2*, *Stereo-Net* and *BDfF-Net*. Results show that *BDfF-Net* benefits from both *Focus-Net-v2* and *Stereo-Net* to

Figure 5.13: To emulate our B-DfF setup, we combine a pair of Lytro Illum cameras into a stereo setup.

offer smoother depth with sharp edges. The experiments also demonstrate that models learned from our dataset could be transferred to predict real scene depth.

5.6 Discussions

Our *BDfF-Net* exploits efficient learning and computational light field imaging to infer depths from a focal stack pair. Our technique mimics human vision system that simultaneously employs binocular stereo matching and monocular depth-from-focus. Comprehensive experiments show that our technique is able to produce high quality depth estimation orders of magnitudes faster than the prior art. In addition, we have created a large dual focal stack database with ground truth disparity.

Our current implementation limits the input size of our network to be focal stacks of 16 layers. In our experiments, we have shown that it is able to produce high fidelity depth estimation under our setup. To handle denser focal stacks, one possibility is to concatenate all images in the stack as a 3D (XYS) focal cube or volume [135], where X and Y are the width and height and S is the index of a layer. We can then downsample the XS slice along S dimension to 16 slices using light field compression or simplification techniques such as tensor [112] and triangulation [126]. Another important future direction we plan to explore is to replace one of the two focal stacks to be an all-focus image. This would further reduce the computational cost for constructing the network but would require adjusting the architecture.

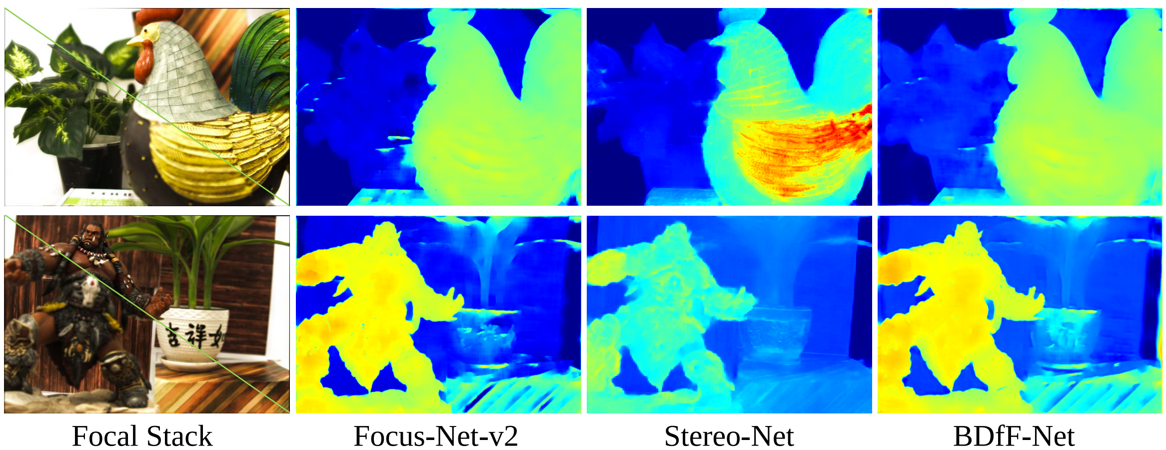


Figure 5.14: Comparisons of real scene results from *Focus-Net-v2*, *Stereo-Net* and *BDfF-Net*.

Chapter 6

HYBRID DEPTH FROM DEFOCUS AND STEREO IMAGING

6.1 Background

In this chapter, we investigate a similar but slightly different setup with chapter 5. Given an all-focus stereo pair and a defocused image of one of the stereo views, we propose a learning based approach to extract depth from the image triplets. While chapter 5 focuses on combining depth from focus (DfF) with stereo, this chapter exploits combining depth from defocus (DfD) with stereo.

It is important to note that DfD and stereo are complementary to each other: stereo provides accurate depth estimation even for distant objects whereas DfD can reliably handle repetitive texture patterns. In computational imaging, a number of hybrid sensors have been designed to combine the benefits of the two. In this chapter, we seek to leverage deep learning techniques to infer depths in such hybrid DfD and stereo setups. While recent advances in neural network have revolutionized both high-level and low-level vision, most existing solutions have exploited only stereo cues [69, 127, 128] and very little work addresses using deep learning for hybrid stereo and DfD or even DfD alone, mainly due to the lack of a fully annotated DfD dataset.

In our setup, we adopt a three images setting: an all-focus stereo pair and a defocused image of one of the stereo views, the left view in our case. We have physically constructed such a hybrid sensor by using Lytro Illum camera. We first generate a comprehensive training dataset for such an imaging setup. Similar to chapter 5, our dataset is based on FlyingThings3D from [73], which contains stereo color pairs and ground truth disparity maps. We then apply occlusion-aware light field rendering[123] to synthesize the defocused image. Next, we adopt the hourglass network [83] architecture to extract depth from stereo and defocus respectively. Hourglass network features

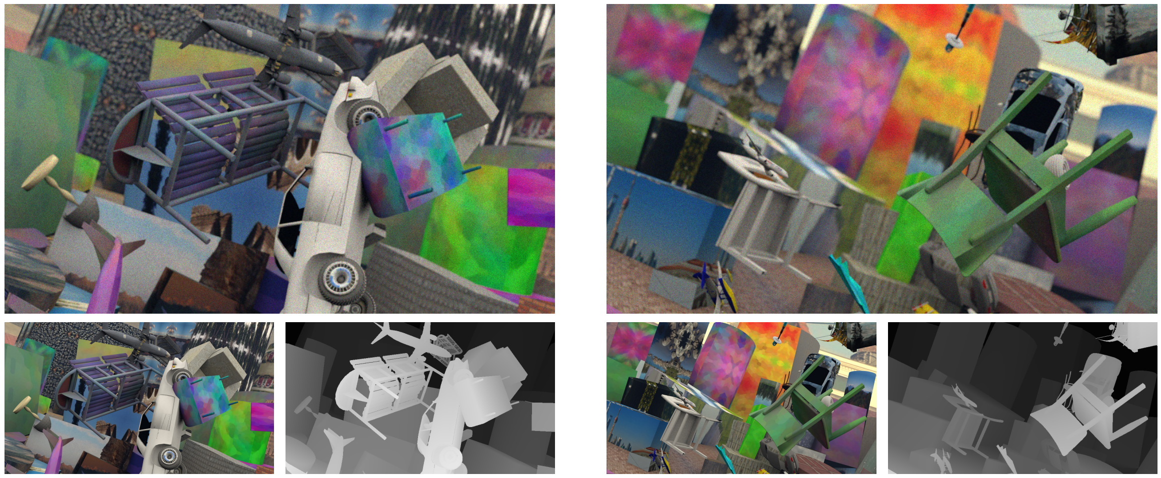


Figure 6.1: Top row shows the generated defocused image by using *Virtual DSLR* technique (best viewed in the electronic version by zooming in). The bottom row shows the ground truth color and depth images. We add Poisson noise to training data, a critical step for handling real scenes.

a multi-scale architecture that consolidates both local and global contextures to output per-pixel depth. We use stacked hourglass network to repeat the bottom-up, top-down depth inferences, allowing for refinement of the initial estimates. Finally, we exploit different connection methods between the two separate networks for integrating them into a unified solution to produce high fidelity 3D depth maps. Comprehensive experiments on real and synthetic data show that our new learning-based hybrid 3D sensing technique can significantly improve accuracy and robustness in 3D reconstruction.

6.2 Training Data

The key to any successful learning based depth inference scheme is a plausible training dataset. Numerous datasets have been proposed for stereo matching but very few are readily available for defocus based depth inference schemes. To address the issue, we set out to create a comprehensive DfD dataset. Our dataset generation similar to the dual focal stack dataset in chapter 5. The dataset is based on FlyingThing3D [73], a synthetic dataset consisting of 25,000 stereo images with ground truth disparities. We again adopt the *Virtual DSLR* approach from [123] to simulate the defocused

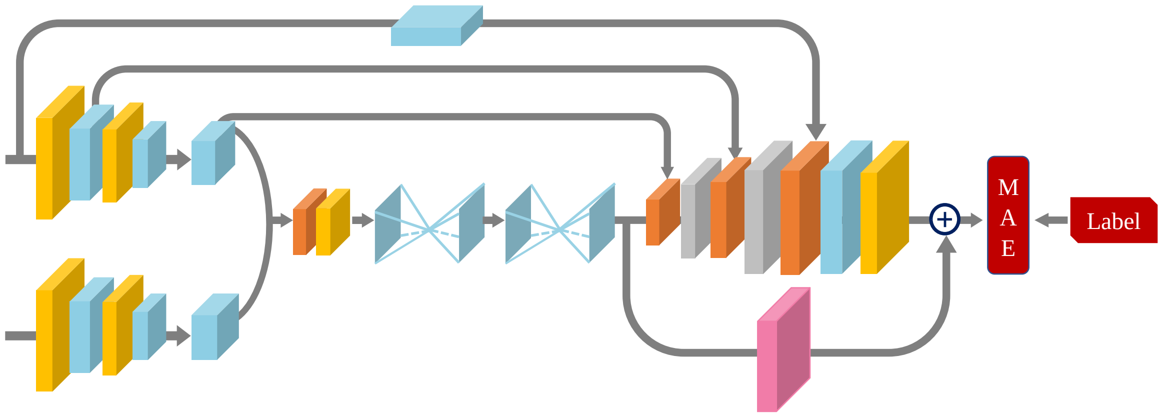


Figure 6.2: The overall architecture of HG-DfD-Net and HG-Stereo-Net. The hourglass structure in the middle represents the two stack HG network. The siamese network before the HG network aims to reduce the feature map size, while the deconvolution layers (gray) progressively recover the feature map to its original resolution. At each scale the upsampled low resolution features are fused with high resolution features by using the concatenating layer (orange).

image.

To emulate different focus settings of the camera, we randomly set the focal plane, and select the size of the blur kernel in the range of $7 \sim 23$ pixels. Finally, we add Poisson noise to both defocused image and the stereo pair to simulate the noise contained in real images. Our final training dataset contains 750 training samples and 160 testing samples, with each sample containing one stereo pair and the defocused image of the left view. The resolution of the generated images is 960×540, the same as the ones in FlyingThings3D. Figure 6.1 shows two samples of our training set.

6.3 DfD-Stereo Network Architecture

Depth inference requires integration of both fine- and large-scale structures. For DfD and stereo, the depth cues could be distributed at various scales in an image. For instance, textureless background requires understanding of a large region, while objects with complex shapes need attentive evaluation of fine details. To capture the contextual information across different scales, a number of recent approaches adopt multi-scale networks and the corresponding solutions have shown plausible results [22, 47]. In addition, recent studies [94] have shown that a deep network with small kernels are very effective in image recognition tasks. In comparison to large kernels, multiple layers of small kernels maintain a large receptive field while reducing the number of parameters to avoid overfitting. Therefore, we design our network with small kernels in a deep multi-scale architecture.

6.3.1 Hourglass Network for DfD and Stereo

Based on the observations above, we construct multi-scale networks that follow the hourglass (HG) architecture [83] for both DfD and stereo.

We have introduced the HG network in Chapter 5 and shown its structure. Here we will briefly reiterate its architecture. The advantage of this network is that it can attentively evaluate the coherence of features across scales by utilizing large amount of residual modules [41]. The network composes of downsampling part and upsampling part. The downsampling part consists of a series of max pooling interleaved with residual modules while the upsampling part is a mirrored architecture of the downsampling part, with max pooling replaced by nearest neighbor upsampling layer for upsampling. Between any pair of corresponding max pooling and upsampling, there is a skip layer comprising of a residual module. Elementwise addition follows to add processed lower-level features to higher-level features. In this way, the network learns a more holistic representation of input images. Prediction is generated at the end of the upsampling part. One round of downsampling and upsampling part can be viewed as one iteration of predicting, whereas additional rounds can be stacked to refine initial estimates. For *StereoNet*, we use two rounds of downsampling and upsampling parts as they already give good performance, while further rounds improve marginally at the cost of more training time. Note that the weights are not shared in the two rounds. After each round of downsampling and upsampling, we add intermediate supervision since the overall network is deep.

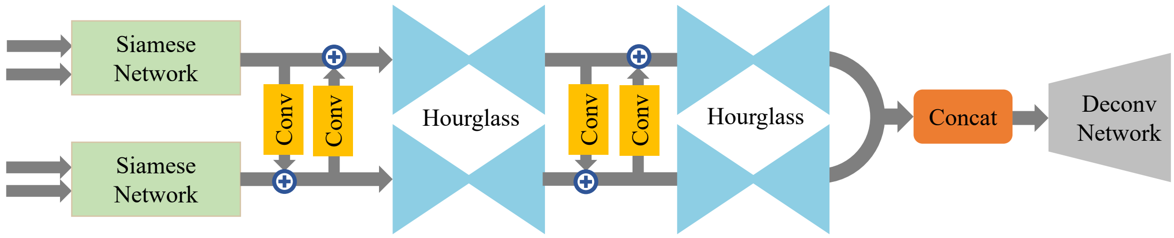


Figure 6.3: Architecture of HG-Fusion-Net. The convolution layers exchange information between networks at various stages, allowing the fusion of defocus and disparity cues.

Before the two-stack HG network, we add a siamese network, whose two network branches share the same architecture and weights. By using convolution layers that have a stride of 2, the siamese network serves to shrink the size of the feature map, thus reducing the memory usage and computational cost of the HG network. After the HG network, we apply deconvolution layers to progressively recover the image to its original size. At each scale the upsampled low resolution features are fused with high-resolution features from siamese network. This upsampling process with multi-scale guidance allows structures to be resolved at both fine- and large-scale. Note that based on our experiment, the downsample/upsample process largely facilitates the training and produces results that are very close to those obtained from full resolution patches. Finally, the network produces pixel-wise disparity prediction at the end. For DfD and stereo, we utilize the same HG architecture, which we call HG-DfD-Net and HG-Stereo-Net. Figure 6.2 shows the overall structure of both networks.

6.3.2 Network Fusion

The most brute-force approach to integrate DfD and stereo is to directly concatenate the output disparity maps from the two branches and apply more convolutions. However, such an approach does not make use of the features readily presented in the branches and hence neglects cues for deriving the appropriate combination of the predicted maps. Consequently, such naive approaches tend to average the results of two branches rather than making further improvement, as shown in Table 6.1.

Instead, we propose HG-Fusion-Net to fuse DfD and stereo, as illustrated in figure 6.3. HG-Fusion-Net consists of two HG networks, with extra connections between them. Each connection applies an 1×1 convolution on the features of one network and adds to the other one. In doing so, the two sub-networks can exchange information at various stages, which is critical for different cues from the two networks to interact with each other. The 1×1 convolution kernel serves as a transformation of feature space, consolidating new cues into the other branch.

In our network, we set up pairs of interconnections at two spots, one at the beginning of each hourglass. At the cost of only four 1×1 convolutions, the interconnections largely proliferate the paths of the network. The HG-Fusion-Net can be regarded as an ensemble of original HG networks with different lengths that enables much stronger representation power. In addition, the fused network avoids solving the whole problem all at once, but first collaboratively solves the stereo and DfD sub-problems, then merges into one coherent solution.

In addition to the above proposal, we also explore multiple variants of the HG-Fusion-Net. With no interconnection, the HG-Fusion-Net simply degrades to the brute-force approach. A compromise between our HG-Fusion-Net and the brute-force approach would be using only one pair of interconnections. We choose to keep the first pair, the one before the first hourglass, since it would enable the network to exchange information early. Apart from the number of interconnections, we also investigate the identity interconnections, which directly adds features to the other branch without going through 1×1 convolution. We present the quantitative results of all the models on Table 6.1.

6.4 Implementation

Optimization

The input of HG-DfD-Net, HG-Stereo-Net, HG-Fusion-Net are defocused/focus image pair, stereo pair and stereo pair plus the defocused image of the left view, respectively. All networks are trained in an end-to-end fashion. For the loss we use the mean absolute error (MAE) with l_2-norm regularization. We adopt MXNET [13] deep learning framework to implement and train our models. Our implementation applies batch normalization [48] after each convolution layer, and use PRelu layer [40] to add nonlinearity to the network while avoiding "dead" filters. We also use the technique from [40] to initialize the weights. For the network solver we choose the Adam optimizer [54] and set the initial learning rate to 0.001, weight decay = 0.002, $\beta 1 = 0.9$, $\beta 2 = 0.999$. We train and test all the models on a NVIDIA Tesla K80 graphic card.

Data Preparation and Augmentation To prepare the data, we first stack the stere/defocus pair along the channel's direction, then extract patches from the stacked image with a stride of 64 to increase the number of training samples. Recall that the HG network contains multiple max pooling layers for downsampling, the patch needs to be cropped to the nearest number that is multiple of 64 for both height and width. In the training phase, we use patches of size 512×256 as input. The large patch contains enough contextual information to recover depth from both defocus and stereo. To increase the generalization of the network, we also augment the data by flipping the patches horizontally and vertically. We perform the data augmentation on the fly at almost no additional cost.

6.5 Experiments

6.5.1 Synthetic Data

We train the HG-DfD-Net, HG-Stereo-Net and HG-Fusion-Net separately, and then conduct experiments on test samples from the synthetic data. Figure 6.4(a) compares the results of three networks. We observe that results from HG-DfD-Net show clearer depth edge, but also exhibit noise on flat regions. On the contrary, HG-Stereo-Net provides smooth depth. However, there is depth bleeding across boundaries, especially when there are holes, such as the tire of the motorcycle on the first row. We suspect that the depth bleeding is due to the occlusion, by which DfD is less affected. Finally, HG-Fusion-Net finds the optimal combination of the two, producing smooth depth while keeping sharp depth boundaries. Table 6.1 also quantitatively describes the performance of different models on our synthetic dataset. Results from Table 6.1 confirm that HG-Fusion-Net achieves the best result for almost all metrics, with notable margin ahead of using stereo or defocus cues alone. The brute-force fusion approach without interconnection only averages results from HG-DfD-Net and HG-Stereo-Net, making no further improvement. The network with fewer or identity interconnection performs slightly worse than the HG-Fusion-Net, but still a lot better than the network without interconnection. This demonstrates that interconnections can efficiently broadcast information across branches and largely facilitate mutual optimization.

We also conduct another experiment on a scene with a staircase textured by horizontal stripes, as illustrated in figure 6.4(b). The scene is rendered from the front view, making it extremely challenging for stereo since all the edges are parallel to the epipolar line. On the contrary, DfD will be able to extract the depth due to its 2D aperture. Figure 6.4(b) shows the resultant depths enclosed in the red box of the front view, proving the effectiveness of our learning-based DfD on such difficult scene. Note that the inferred depth is not perfect. This is mainly due to the fact that our training data lacks objects with stripe texture. We can improve the result by adding similar textures to the training set.

6.5.2 Real Scene

To conduct experiments on the real scene, we use light field (LF) camera to capture the LF and generate the defocused image. LF camera captures a rich set of rays to describe the visual appearance of the scene. In free space, LF is commonly represented by two-plane parameterizations $L(u, v, s, t)$, where st is the camera plane and uv is the image plane [62]. To conduct digital refocusing, we can move the synthetic image plane that leads to the following photography equation [84]:

$$E(s,t) = \iint L(u, v, u + \frac{s-u}{\alpha}, v + \frac{t-v}{\alpha}) dudv \tag{6.1}$$

By varying α, we can refocus the image at different depth. Note that by fixing st, we obtain the sub-aperture image $L_{(s^\star t^\star)}(u, v)$ that is amount to the image captured

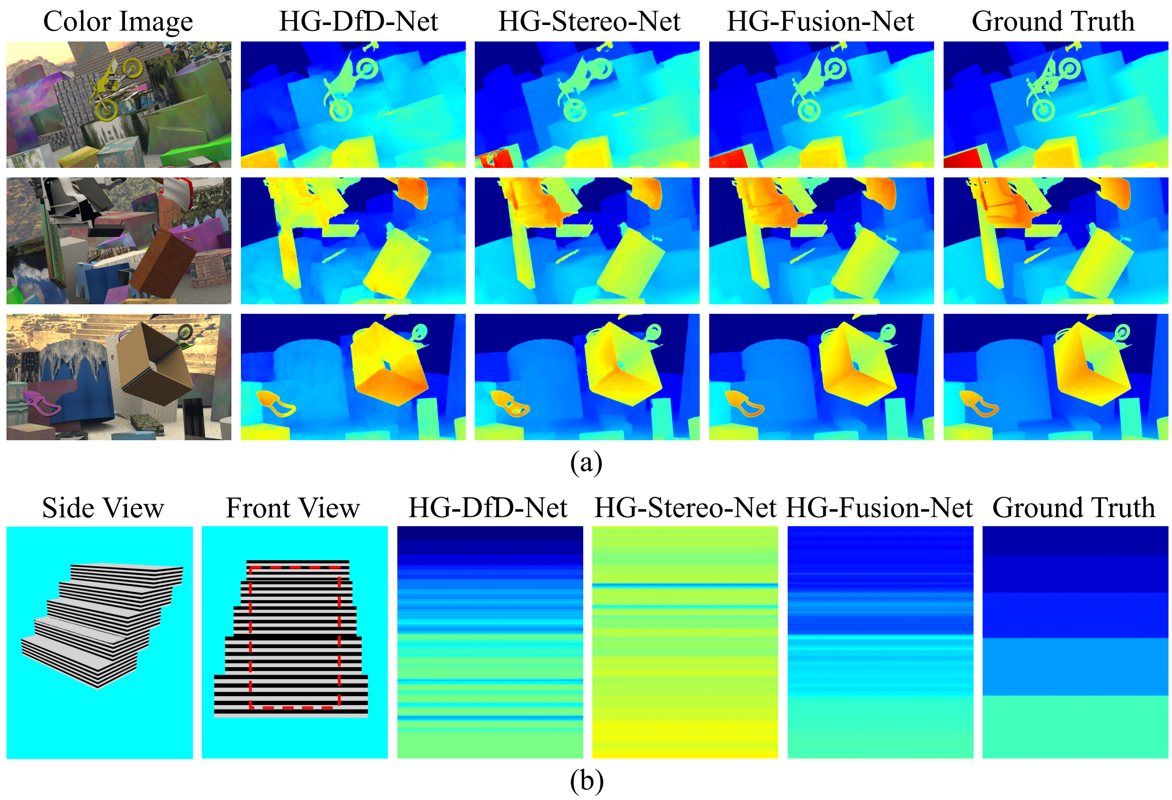


Figure 6.4: Results of HG-DfD-Net, HG-Stereo-Net and HG-Fusion-Net on (a) our dataset (b) staircase scene textured with horizontal stripes. HG-Fusion-Net produces smooth depth at flat regions while maintaining sharp depth boundaries. Best viewed in the electronic version by zooming in.

using a sub-region of the main lens aperture. Therefore, Eqn. 6.1 corresponds to shift-and-add the sub-aperture images [84].

In our experiment we use Lytro Illum camera as our capturing device. We first mount the camera on a translation stage and move the LF camera horizontally to capture two LFs. Then we extract the sub-aperture images from each LF using Light Field Toolbox [19]. The two central sub-aperture images are used to form a stereo pair. We also use the central sub-aperture image in the left view as the all-focused image due to its small aperture size. Finally, we apply the shift-and-add algorithm to generate the defocused image. Both the defocused and sub-aperture image has the size of 625×433.

	> 1 px	> 3 px	> 5 px	MAE (px)	Time (s)
HG-DfD-Net	70.07%	38.60%	20.38%	3.26	0.24
HG-Stereo-Net	28.10%	6.12%	2.91%	1.05	0.24
HG-Fusion-Net	**20.79%**	5.50%	**2.54%**	**0.87**	0.383
No Interconnection	45.46%	10.89%	5.08%	1.57	0.379
Less Interconnection	21.85%	**5.23%**	2.55%	0.91	0.382
Identity Interconnection	21.37%	6.00%	2.96%	0.94	0.382

Table 6.1: Quantitative results of proposed models. Upper half compares results from different input combinations: defocus pair, stereo pair and stereo pair + defocused image. Lower half compares various fusion scheme, mainly differentiating by the number and type of interconnection: *No interconnection* is the brute-force approach that only concatenates feature maps after the HG network, before the deconvolution layers. *Less Interconnection* only uses one interconnection before the first hourglass; *Identity Interconnection* directly adds features to the other branch, without applying the 1×1 convolution.

The result of real scene is shown in Fig.6.5. We have conducted tests on both indoor and outdoor scenes. In general, both HG-DfD-Net and HG-Stereo-Net preserve depth edges well, but results from HG-DfD-Net are noisier. HG-Fusion-Net produces the best results with smooth depth and sharp depth boundaries. The plant in the first row of Fig.6.5 presents challenges for both stereo and DfD methods due to the heavy occlusion of branches and leaves. But HG-Fusion-Net manages to identify the fine structure of leaves and generate correct depth value. We have also trained HG-Fusion-Net on a clean dataset without Poisson noise, and show the results in the last column of Fig.6.5. The inferred depths exhibit severe noise pattern on real data, confirming the necessity to add noise to dataset for simulating real images.

6.6 Discussion

We have presented a learning based solution for a hybrid DfD and stereo depth sensing scheme. We have adopted the hourglass network architecture to separately extract depth from defocus and stereo. We have then studied and explored multiple neural network architectures for linking both networks to improve depth inference.

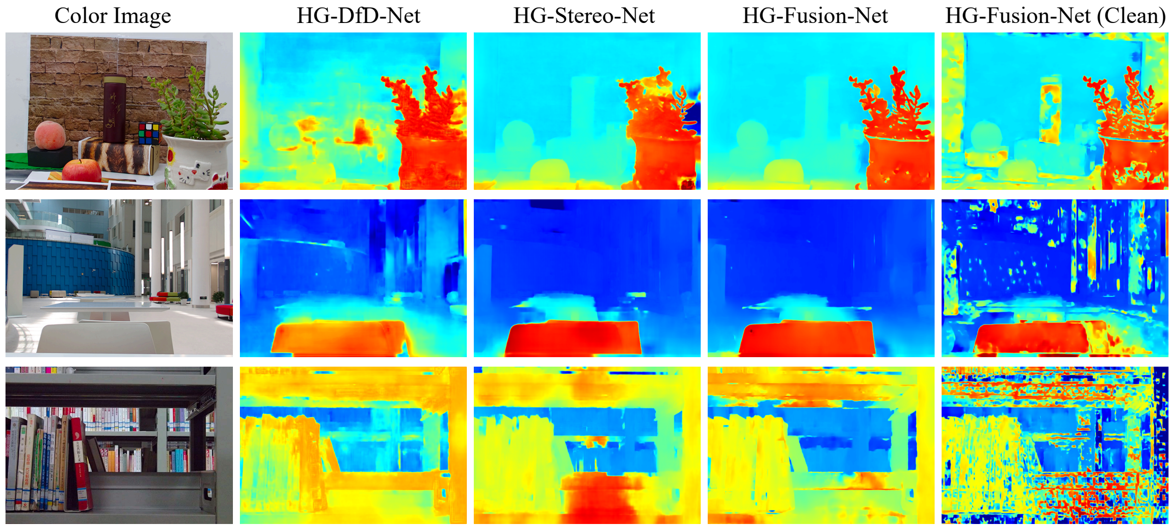


Figure 6.5: Comparisons of real scene results from HG-DfD-Net, HG-Stereo-Net and HG-Fusion-Net. The last column shows the results from HG-Fusion-Net trained by the clean dataset without Poisson noise. Best viewed in color.

Comprehensive experiments show that our proposed approach preserves the strength of DfD and stereo while effectively suppressing their weaknesses. In addition, we have created a large synthetic dataset for our setup that includes image triplets of a stereo pair and a defocused image along with the corresponding ground truth disparity.

Our immediate future work is to explore different DfD inputs and their interaction with stereo. For instance, instead of using a single defocused image, we can vary the aperture size to produce a stack of images where objects at the same depth exhibit different blur profiles. Learning based approaches can be directly applied to the profile for depth inference or can be combined with our current framework for conducting hybrid depth inference. We have presented one DfD-Stereo setup. Another minimal design was shown in [104], where a stereo pair with different focus distance is used as input. In the future, we will study the cons and pros of different hybrid DfD-stereo setups and tailor suitable learning-based solutions for fully exploiting the advantages of such setups.

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Conclusions

In this dissertation, I have presented several computational imaging algorithms and systems to infer geometry of the scene.

Mobile Multi-flash System To obtain a qualitative depth map of the scene on a mobile platform, we have presented a new mobile multi-flash camera that uses the mobile device's own flash as a pseudo synchronization unit. Our mobile MF camera is compact, light-weight, inexpensive and can be mounted on most smart phones and tablets as a hand-held imaging system. The corresponding algorithm is tailored to mobile platform and is able to extract depth map, depth edge of the scene, as well as produce non-photorealistic effects.

A portable Immersive System using RGB-D Sensor We have also developed a system that is based on structured light technique to recover large scale structure in real time. Specifically, we use the Microsoft Kinect sensor as the acquisition device and develop a class of multi-view 3D fusion techniques to faithfully reconstruct the event. We have conducted preliminary tests of the system fidelity for cholecystectomy (gallbladder surgery) training and have developed a space-time visualization system to display the acquired data. Furthermore, we integrate our system with 3D stereoscopic displays to enhance the user experience.

Depth from a Single Light Field To extract depth in a time sensitive scenario, we have developed a fast depth extraction algorithm that is tailored to barcode. Our algorithm first localize the barcode region in the raw light field image, then jointly analyze the size and the statistical characteristics of the barcode region to infer depth.

Finally, with the depth information we only need to render one focal slice that focuses on the barcode plane, which dramatically reduce the amount of time needed to produce a decodable image.

Depth from Dual Light Fields Human vision system perceives depth with both disparity cue and focus cue. Therefore, we have presented a learning based technique mimics human vision system that simultaneously employs binocular stereo matching and monocular depth-from-focus. Given a binocular focal stack as input, we propose BDfDNet to extract depth. We decompose BDfFNet into sub-networks and first train each sub-network separately before combining them to further finetune the result. This allows us to infer depth from either a single focal stack or the dual focal stack. Comprehensive experiments show that our technique is able to produce high quality depth estimation orders of magnitudes faster than the prior art. In addition, we have created a large dual focal stack database with ground truth disparity.

Hybrid Depth from Defocus and Stereo Imaging We have also investigated combining the disparity cue with the defocus cue. Given an all-focus stereo pair and a defocused image of one of the stereo views, we propose a learning based approach to extract depth from the image triplets. We have adopted the hourglass network architecture to separately extract depth from defocus and stereo. We have then studied and explored multiple neural network architectures for linking both networks to improve depth inference. Comprehensive experiments show that our proposed approach preserves the strength of DfD and stereo while effectively suppressing their weaknesses.

7.2 Future Work

There are several directions for future research.

Mobile Multi-flash System The mobile multi-flash system will enable several new applications. On the computer vision front, we can rely on the inferred depth to aid object detection, tracking and recognition on mobile devices. On the graphics front, we can explore a broader range of image manipulation applications such as depth edge guided image retargeting, and distracting regions de-emphasis. We will also investigate using the mobile multi-flash technique for enhancing hand gesture and head pose based human-computer interaction.

A portable Immersive System using RGB-D Sensor Based on structured light technique, the RGB-D sensor has shown great potential. However, our current implementation only recovers point clouds. In the future we will focus on generating a mesh from the depth maps and color images, making the virtual navigation more realistic.

Depth from a Single Light Field For light field barcode scanner, our current algorithm only support the 1D barcode. Our immediate next step will extend our system to 2D barcode scanning. We will also explore other application that could benefit from our statistical analysis on the raw light field image.

Depth from Dual Light Fields Our current implementation of the BDfDNet limits the input size of our network to be focal stacks of 16 layers. Our experiments have shown that it is able to produce high fidelity depth estimation under our setup. To handle denser focal stacks, one possibility is to concatenate all images in the stack as a 3D (XYS) focal cube or volume [135], where X and Y are the width and height and S is the index of a layer. We can then downsample the XS slice along S dimension to 16 slices using light field compression or simplification techniques such as tensor [112] and triangulation [126]. Another important future direction we plan to explore is to replace one of the two focal stacks to be an all-focus image. This would further reduce the computational cost for constructing the network but would require adjusting the architecture.

Hybrid Depth from Defocus and Stereo Imaging Exploring different depth from defocus inputs and their interaction with stereo will be our immediate next step. For instance, instead of using a single defocused image, we can vary the aperture size to produce a stack of images where objects at the same depth exhibit different blur profiles. Learning based approaches can be directly applied to the profile for depth inference or can be combined with our current framework for conducting hybrid depth inference. We have presented one DfD-Stereo setup. Another minimal design was shown in [104], where a stereo pair with different focus distance is used as input. In the future, we will study the cons and pros of different hybrid DfD-stereo setups and tailor suitable learning-based solutions for fully exploiting the advantages of such setups.

BIBLIOGRAPHY

[1] Edward H Adelson and James R Bergen. The plenoptic function and the elements of early vision. 1991.

[2] Gerald J Agin and Thomas O Binford. Computer description of curved objects. In *Proceedings of the 3rd international joint conference on Artificial intelligence*, pages 629–640. Morgan Kaufmann Publishers Inc., 1973.

[3] Motilal Agrawal and Larry S Davis. A probabilistic framework for surface reconstruction from multiple images. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages 470–476, 2001.

[4] Emma Alexander, Qi Guo, Sanjeev Koppal, Steven Gortler, and Todd Zickler. Focal flow: Measuring distance and velocity with defocus and differential motion. In *European Conference on Computer Vision*, pages 667–682. Springer, 2016.

[5] Joan Batlle, E Mouaddib, and Joaquim Salvi. Recent progress in coded structured light as a technique to solve the correspondence problem: a survey. *Pattern recognition*, 31(7):963–982, 1998.

[6] Marcelo Bertalmio, Andrea L Bertozzi, and Guillermo Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[7] Michael J Black, Guillermo Sapiro, David H Marimont, and David Heeger. Robust anisotropic diffusion. *IEEE Transactions on image processing*, 7(3):421–432, 1998.

[8] J-Y Bouguet and Pietro Perona. 3d photography on your desk. In *Computer Vision, 1998. Sixth International Conference on*, pages 43–50. IEEE, 1998.

[9] V Michael Bove. Entropy-based depth from focus. *JOSA A*, 10(4):561–566, 1993.

[10] Myron Z Brown, Darius Burschka, and Gregory D Hager. Advances in computational stereo. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):993–1008, 2003.

[11] Douglas Chai and Florian Hock. Locating and decoding ean-13 barcodes from images captured by digital cameras. In *Information, Communications and Signal Processing, 2005 Fifth International Conference on*, pages 1595–1599. IEEE, 2005.

[12] Jiawen Chen, Dennis Bautembach, and Shahram Izadi. Scalable real-time volumetric surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(4):113, 2013.

[13] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *CoRR*, abs/1512.01274, 2015.

[14] Zhuoyuan Chen, Xun Sun, Liang Wang, Yinan Yu, and Chang Huang. A deep visual correspondence embedding model for stereo matching costs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 972–980, 2015.

[15] Clement Creusot and Asim Munawar. Real-time barcode detection in the wild. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 239–245. IEEE, 2015.

[16] Brian Curless and Marc Levoy. Better optical triangulation through spacetime analysis. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 987–994, 1995.

[17] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996.

[18] Don Dansereau and Len Bruton. Gradient-based depth estimation from 4d light fields. In *Circuits and Systems, 2004. ISCAS'04. Proceedings of the 2004 International Symposium on*, volume 3, pages III–549. IEEE, 2004.

[19] Donald G Dansereau, Oscar Pizarro, and Stefan B Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1027–1034, 2013.

[20] Michael Daum and Gregory Dudek. On 3-d surface reconstruction using shape from shadows. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 461–468. IEEE, 1998.

[21] Abe Davis, Marc Levoy, and Fredo Durand. Unstructured light fields. In *Computer Graphics Forum*, volume 31, pages 305–314. Wiley Online Library, 2012.

[22] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[23] Elmar Eisemann and Frédo Durand. Flash photography enhancement via intrinsic relighting. *ACM transactions on graphics (TOG)*, 23(3):673–678, 2004.

[24] John Ens and Peter Lawrence. A matrix based method for determining depth from focus. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 600–606. IEEE, 1991.

[25] Paolo Favaro and Stefano Soatto. A geometric approach to shape from defocus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):406–417, 2005.

[26] Paolo Favaro, Stefano Soatto, Martin Burger, and Stanley J Osher. Shape from defocus via diffusion. *IEEE transactions on pattern analysis and machine intelligence*, 30(3):518–531, 2008.

[27] Rogerio Feris, Ramesh Raskar, Longbin Chen, Karhan Tan, and Matthew Turk. Multiflash stereopsis: Depth-edge-preserving stereo with small baseline illumination. *IEEE transactions on pattern analysis and machine intelligence*, 30(1):147–159, 2008.

[28] Rogerio Feris, Matthew Turk, and Ramesh Raskar. Dealing with multi-scale depth changes and motion in depth edge detection. In *Computer Graphics and Image Processing, 2006. SIBGRAPI'06. 19th Brazilian Symposium on*, pages 3–10. IEEE, 2006.

[29] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rüther, and Horst Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 993–1000, 2013.

[30] Juliet Fiss, Brian Curless, and Richard Szeliski. Refocusing plenoptic images using depth-adaptive splatting. In *Computational Photography (ICCP), 2014 IEEE International Conference on*, pages 1–9. IEEE, 2014.

[31] H Fuchs and U Neumann. A vision of telepresence for medical consultation and other applications. In *Sixth International Symposium of Robotics Research*, pages 555–571, 1993.

[32] Henry Fuchs, Gary Bishop, Kevin Arthur, Leonard McMillan, Ruzena Bajcsy, Sang Lee, Hany Farid, and Takeo Kanade. Virtual space teleconferencing using a sea of cameras. In *Proc. First International Conference on Medical Robotics and Computer Assisted Surgery*, volume 26, 1994.

[33] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

[34] Orazio Gallo and Roberto Manduchi. Reading 1d barcodes with mobile phones using deformable templates. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1834–1843, 2011.

[35] Todor Georgiev, Zhan Yu, Andrew Lumsdaine, and Sergio Goma. Lytro camera technology: theory, algorithms, performance analysis. In *Proc. SPIE*, 2013.

[36] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM, 1996.

[37] Mohit Gupta, Shree K Nayar, Matthias B Hullin, and Jaime Martin. Phasor imaging: A generalization of correlation-based time-of-flight imaging. *ACM Transactions on Graphics (ToG)*, 34(5):156, 2015.

[38] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3279–3286, 2015.

[39] Samuel W Hasinoff and Kiriakos N Kutulakos. Confocal stereo. *International journal of computer vision*, 81(1):82, 2009.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[42] Stefan Heber and Thomas Pock. Shape from light field meets robust pca. In *European Conference on Computer Vision*, pages 751–767. Springer, 2014.

[43] Stefan Heber and Thomas Pock. Convolutional networks for shape from light field. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3746–3754, 2016.

[44] Felix Heide, Matthias B Hullin, James Gregson, and Wolfgang Heidrich. Low-budget transient imaging using photonic mixer devices. *ACM Transactions on Graphics (ToG)*, 32(4):45, 2013.

[45] Robert T Held, Emily A Cooper, and Martin S Banks. Blur and disparity are complementary cues to depth. *Current biology*, 22(5):426–431, 2012.

[46] Daniel Herrera, Juho Kannala, and Janne Heikkilä. Joint depth and color camera calibration with distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):2058–2064, 2012.

[47] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. Depth map super-resolution by deep multi-scale guidance. In *European Conference on Computer Vision*, pages 353–369. Springer, 2016.

[48] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[49] Aaron Isaksen, Leonard McMillan, and Steven J Gortler. Dynamically reparameterized light fields. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 297–306. ACM Press/Addison-Wesley Publishing Co., 2000.

[50] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 559–568, 2011.

[51] Hae-Gon Jeon, Jaesik Park, Gyeongmin Choe, Jinsun Park, Yunsu Bok, Yu-Wing Tai, and In So Kweon. Accurate depth map estimation from a lenslet light field camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1547–1555, 2015.

[52] Changil Kim, Henning Zimmer, Yael Pritch, Alexander Sorkine-Hornung, and Markus Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.*, 32(4):73:1–73:12, 2013.

[53] Yongjin Kim, Jingyi Yu, Xuan Yu, and Seungyong Lee. Line-art illustration of dynamic and specular surfaces. In *ACM Transactions on Graphics (TOG)*, volume 27, page 156. ACM, 2008.

[54] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[55] William N Klarquist, Wilson S Geisler, and Alan C Bovik. Maximum-likelihood depth-from-defocus for active vision. In *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 3, pages 374–379. IEEE, 1995.

[56] Patrick Knöbelreiter, Christian Reinbacher, Alexander Shekhovtsov, and Thomas Pock. End-to-end training of hybrid cnn-crf models for stereo. *arXiv preprint arXiv:1611.10229*, 2016.

[57] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. *Computer VisionECCV 2002*, pages 8–40, 2002.

[58] Dilip Krishnan and Rob Fergus. Dark flash photography. *ACM Trans. Graph.*, 28(3):96:1–96:11, 2009.

[59] Sujit Kuthirummal, Hajime Nagahara, Changyin Zhou, and Shree K Nayar. Flexible depth of field photography. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):58–71, 2011.

[60] Douglas Lanman and Gabriel Taubin. Build your own 3d scanner: 3d photography for beginners. In *ACM SIGGRAPH 2009 Courses*, page 8. ACM, 2009.

[61] Anat Levin, Rob Fergus, Frédo Durand, and William T Freeman. Image and depth from a conventional camera with a coded aperture. *ACM transactions on graphics (TOG)*, 26(3):70, 2007.

[62] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996.

[63] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, et al. The digital michelangelo project: 3d scanning of large statues. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 131–144. ACM Press/Addison-Wesley Publishing Co., 2000.

[64] Nestk Library. https://github.com/nburrus/nestk.

[65] Kok lim Low, Adrian Ilie, Greg Welch, and Anselmo Lastra. Combining head-mounted and projector-based displays for surgical training. In *in Proceedings of IEEE Virtual Reality 2003*, pages 110–117, 2003.

[66] Ming-Yu Liu, Oncel Tuzel, Ashok Veeraraghavan, Rama Chellappa, Amit K. Agrawal, and Haruhisa Okuda. Pose estimation in heavy clutter using a multi-flash camera. In *ICRA*, pages 2028–2035, 2010.

[67] Zishun Liu, Zhenxi Li, Juyong Zhang, and Ligang Liu. Euclidean and hamming embedding for image patch description with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 72–78, 2016.

[68] Andrew Lumsdaine and Todor Georgiev. The focused plenoptic camera. In *Computational Photography (ICCP), 2009 IEEE International Conference on*, pages 1–8. IEEE, 2009.

[69] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5695–5703, 2016.

[70] Q-T Luong and Olivier D Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of computer vision*, 22(3):261–289, 1997.

[71] Aamir Saeed Malik, Seong-O Shim, and Tae-Sun Choi. Depth map estimation using a robust focus measure. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 6, pages VI–564. IEEE, 2007.

[72] Luca Marchesotti, Claudio Cifarelli, and Gabriela Csurka. A framework for visual saliency detection with applications to image thumbnailing. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2232–2239. IEEE, 2009.

[73] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[74] Daniel Miau, Oliver Cossairt, and Shree K Nayar. Focal sweep videography with deformable optics. In *Computational Photography (ICCP), 2013 IEEE International Conference on*, pages 1–8. IEEE, 2013.

[75] Michael Moeller, Martin Benning, Carola Schönlieb, and Daniel Cremers. Variational depth from focus reconstruction. *IEEE Transactions on Image Processing*, 24(12):5369–5378, 2015.

[76] Francesc Moreno-Noguer, Peter N Belhumeur, and Shree K Nayar. Active refocusing of images and videos. *ACM Transactions On Graphics (TOG)*, 26(3):67, 2007.

[77] Christian Mostegel, Markus Rumpler, Friedrich Fraundorfer, and Horst Bischof. Using self-contradiction to learn confidence measures in stereo vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4067–4076, 2016.

[78] Ruben Muniz, Luis Junco, and Adolfo Otero. A robust software barcode reader using the hough transform. In *Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on*, pages 313–319. IEEE, 1999.

[79] PJ Narayanan, Peter W Rander, and Takeo Kanade. Constructing virtual worlds using dense stereo. In *Computer Vision, 1998. Sixth International Conference on*, pages 3–10. IEEE, 1998.

[80] Shree K Nayar. Shape from focus system. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 302–308. IEEE, 1992.

[81] Shree K Nayar and Yasuo Nakagawa. Shape from focus. *IEEE Transactions on Pattern analysis and machine intelligence*, 16(8):824–831, 1994.

[82] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[83] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[84] Ren Ng, Marc Levoy, Mathieu Bredif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. *Stanford University Computer Science Tech Report*, 2:1–11, 2005.

[85] Haesol Park and Kyoung Mu Lee. Look wider to match image patches with convolutional neural networks. *IEEE Signal Processing Letters*, 2016.

[86] Alex Paul Pentland. A new sense for depth of field. *IEEE transactions on pattern analysis and machine intelligence*, (4):523–531, 1987.

[87] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. *ACM transactions on graphics (TOG)*, 23(3):664–672, 2004.

[88] AN Rajagopalan and Subhasis Chaudhuri. Optimal selection of camera parameters for recovery of depth from defocused images. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 219–224. IEEE, 1997.

[89] AN Rajagopalan, Subhasis Chaudhuri, and Uma Mudenagudi. Depth estimation and image restoration using defocused stereo pairs. *IEEE transactions on pattern analysis and machine intelligence*, 26(11):1521–1525, 2004.

[90] Ramesh Raskar, Kar-Han Tan, Rogerio Feris, Jingyi Yu, and Matthew Turk. Non-photorealistic camera: depth edge detection and stylized rendering using

multi-flash imaging. In *ACM transactions on graphics (TOG)*, volume 23, pages 679–688. ACM, 2004.

[91] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 179–188. ACM, 1998.

[92] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[93] Akihito Seki and Marc Pollefeys. Patch based confidence prediction for dense disparity map. In *BMVC*, volume 10, 2016.

[94] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[95] Oliver G. Staadt, Markus H. Gross, Andreas Kunz, Markus H. Gross Andreas Kunz, and Markus Meier. The Blue-C: Integrating real humans into a networked immersive environment. In *In Proceedings of ACM Collaborative Virtual Environments 2000*, pages 201–202, 2000.

[96] Murali Subbarao and Gopal Surya. Depth from defocus: a spatial domain approach. *International Journal of Computer Vision*, 13(3):271–294, 1994.

[97] Murali Subbarao, Ta Yuan, and Jenn-Kwei Tyan. Integration of defocus and focus analysis with stereo for 3d shape recovery. In *Proceedings of SPIE*, volume 3204, pages 11–23, 1997.

[98] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on pattern analysis and machine intelligence*, 25(7):787–800, 2003.

[99] Jin Sun, Christopher Thorpe, Nianhua Xie, Jingyi Yu, and Haibin Ling. Object category classification using occluding contours. *Advances in Visual Computing*, pages 296–305, 2010.

[100] Gopal Surya and Murali Subbarao. Depth from defocus by changing camera aperture: A spatial domain approach. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*, pages 61–67. IEEE, 1993.

[101] Supasorn Suwajanakorn, Carlos Hernandez, and Steven M Seitz. Depth from focus with your mobile phone. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3497–3506, 2015.

[102] Immersive Surgery Training System. https://www.eecis.udel.edu/ xinqing/inbre.

[103] Yuichi Taguchi. Rainbow flash camera: Depth edge extraction using complementary colors. *International journal of computer vision*, 110(2):156–171, 2014.

[104] Yuichi Takeda, Shinsaku Hiura, and Kosuke Sato. Fusing depth from defocus and stereo with coded apertures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 209–216, 2013.

[105] Michael W Tao, Sunil Hadap, Jitendra Malik, and Ravi Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 673–680, 2013.

[106] Jonas Unger, Andreas Wenger, Tim Hawkins, Andrew Gardner, and Paul Debevec. Capturing and rendering with incident light fields. Technical report, UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY CA INST FOR CREATIVE TECHNOLOGIES, 2003.

[107] Vaibhav Vaish, Marc Levoy, Richard Szeliski, C Lawrence Zitnick, and Sing Bing Kang. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2331–2338. IEEE, 2006.

[108] Vaibhav Vaish, Bennett Wilburn, Neel Joshi, and Marc Levoy. Using plane+ parallax for calibrating dense camera arrays. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1. IEEE, 2004.

[109] Daniel A Vaquero, Rogerio S Feris, Matthew Turk, and Ramesh Raskar. Characterizing the shadow space of camera-light pairs. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[110] Ashok Veeraraghavan, Ramesh Raskar, Amit Agrawal, Ankit Mohan, and Jack Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Trans. Graph.*, 26(3):69, 2007.

[111] Ting-Chun Wang, Manohar Srikanth, and Ravi Ramamoorthi. Depth from semi-calibrated stereo and defocus. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3717–3726, 2016.

[112] Sven Wanner and Bastian Goldluecke. Globally consistent depth labeling of 4d light fields. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 41–48. IEEE, 2012.

[113] Sven Wanner and Bastian Goldluecke. Variational light field analysis for disparity estimation and super-resolution. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):606–619, 2014.

[114] Masahiro Watanabe and Shree K Nayar. Telecentric optics for focus analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1360–1365, 1997.

[115] Masahiro Watanabe and Shree K Nayar. Rational filters for passive depth from defocus. *International Journal of Computer Vision*, 27(3):203–225, 1998.

[116] G. Welch, A. State, A. Ilie, Kok-Lim Low, A. Lastra, B. Cairns, H. Towles, H. Fuchs, Ruigang Yang, S. Becker, D. Russo, J. Funaro, and A. van Dam. Immersive electronic books for surgical training. volume 12, pages 22–35, 2005.

[117] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Marc Levoy, and Mark Horowitz. High-speed videography using a dense camera array. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2004.

[118] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High performance imaging using large camera arrays. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 765–776. ACM, 2005.

[119] Oliver Woodford, Philip Torr, Ian Reid, and Andrew Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2115–2128, 2009.

[120] L-Q Xu, B Lei, and E Hendriks. Computer vision for a 3-d visualisation and telepresence collaborative working environment. *BT Technology Journal*, 20(1):64–74, 2002.

[121] Wei Xu and Scott McCloskey. 2d barcode localization and motion deblurring using a flutter shutter camera. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 159–165. IEEE, 2011.

[122] Jason C Yang, Matthew Everett, Chris Buehler, and Leonard McMillan. A real-time distributed light field camera. *Rendering Techniques*, 2002:77–86, 2002.

[123] Yang Yang, Haiting Lin, Zhan Yu, Sylvain Paris, and Jingyi Yu. Virtual DSLR: high quality dynamic depth-of-field synthesis on mobile platforms. In *Digital Photography and Mobile Imaging XII*, pages 1–9, 2016.

[124] Ryunosuke Yokoya and Shree K Nayar. Extended depth of field catadioptric imaging using focal sweep. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3505–3513, 2015.

[125] Xuan Yu, Rui Wang, and Jingyi Yu. Real-time depth of field rendering via dynamic light field generation and filtering. In *Computer Graphics Forum*, volume 29, pages 2099–2107. Wiley Online Library, 2010.

[126] Zhan Yu, Xinqing Guo, Haibing Lin, Andrew Lumsdaine, and Jingyi Yu. Line assisted light field triangulation and stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2792–2799, 2013.

[127] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4353–4361, 2015.

[128] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1592–1599, 2015.

[129] Chunhui Zhang, Jian Wang, Shi Han, Mo Yi, and Zhengyou Zhang. Automatic real-time barcode localization in complex scenes. In *Image Processing, 2006 IEEE International Conference on*, pages 497–500. IEEE, 2006.

[130] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.

[131] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, 2017.

[132] Yuanjie Zheng, Chandra Kambhamettu, Jingyi Yu, Thomas Bauer, and Karl Steiner. Fuzzymatte: A computationally efficient scheme for interactive matting. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[133] Yuanjie Zheng, Jingyi Yu, Chandra Kambhamettu, Sarah Englander, Mitchell D Schnall, and Dinggang Shen. De-enhancing the dynamic contrast-enhanced breast mri for robust registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 933–941. Springer, 2007.

[134] Changyin Zhou, Stephen Lin, and Shree Nayar. Coded aperture pairs for depth from defocus. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 325–332. IEEE, 2009.

[135] Changyin Zhou, Daniel Miau, and Shree K Nayar. Focal sweep camera for space-time refocusing. *Technical Report, Department of Computer Science*, 2012.