Epsilon Focus Photography: A Study of Focus, Defocus and Depth-of-field

Thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy in Computer Science and Engineering

by

Parikshit Vishwas Sakurikar
Roll No. 201199650
Email: parikshit.sakurikar@research.iiit.ac.in

International Institute of Information Technology
(Deemed University)
Hyderabad - 500 032, INDIA
August 2021

Copyright © Parikshit Sakurikar, 2021
All Rights Reserved

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Epsilon Focus Photography - A Study of Focus, Defocus and Depth-of-Field " by Parikshit Vishwas Sakurikar, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Advisor: Prof. P. J. Narayanan

Acknowledgments

I would like to thank my advisor, Prof P. J. Narayanan for always providing the much needed guidance that a young student needs to become a researcher. Throughout the work presented in this dissertation, Prof PJN has been very supportive and has always given me the freedom to explore ideas that I was passionate about, despite those ideas being off-beat compared to standard pursuits. Prof PJN has a unique and pragmatic approach as an advisor and I know that I speak for many of my colleagues in saying that it is a wonderful approach that really allows students to excel. I will always cherish our long discussions about life, research, philosophy and so many things in general. I am also thankful to Prof PJN for allowing me to supervise other post graduate students and work on diverse problems that have helped me broaden my horizons. I am immensely grateful to him for having shaped my early career into something that I am very proud of. It is his guidance that pulled me into pursuing a Ph.D in the first place and I have cherished every moment of this journey.

There have been so many seniors and colleagues over the years who have accompanied and assisted me and I am thankful to all of them for their support. I must specifically thank Pawan Harish who has been an inspiration to me since the day I found out about him. I am thankful to Manan Nayak for turning me into a undergraduate researcher and for helping start this journey. I owe a special thanks to Rajvi Shah, Saurabh Saini and Revanth N.R who have been consistent pillars of support through this journey and without whom I would have certainly handed over this thesis much earlier. I also cherish the work I have done with remarkable students such as Ishit Mehta and K. T. Aakash. This journey has been a humbling experience and having a supportive group of colleagues has truly been a blessing. In addition to my colleagues, I owe a great deal of thanks to the IIIT family in general. I have known the university and everyone in it for so many years and I would do it all over again, given the chance. I am very grateful to Prof C. V. Jawahar, Prof Anoop Namboodiri, Prof Vineet Gandhi, Prof Jayanthi Sivaswamy, Prof Michael Brown and Prof Vineeth Balasubramanian for guiding me through this journey at different times.

In the words of Gerard Manly Hopkins - ”No wonder of it, sheer plod makes plough down sillion shine...”. Nothing that was achieved in this dissertation would have been possible if it weren't for my family and the unconditional support I have received from them. My mother, Shubhada, has been the single most insipirational person in my life for having lived every single moment of the last 30 years trying to make a living for her two children and excelling at it, if I may add. My sister, Nandini, is a

self-made woman whose passion for science and strength in adversity is something that I don't think I can ever emulate. My childhood was shaped by these wonderful women, and I am eternally grateful to them. My wife, Chandni, has been the kindest and most considerate person I have known. She has been a pillar of strength in every sense of the phrase. I often find myself doubting whether there is anyone else who was at the receiving end of such a wonderful and supportive environment. I am thankful to Brent and little Cora for being awesome and for jointly having more qualifications than I do at the moment. I owe a great deal of thanks to Chandni's parents and my grandparents for always being encouraging and supportive of my endeavors. I am extremely grateful to my friend and brother Krishna Chaitanya. There are no words to describe how much he has done for me. I am thankful for the love and support I have received from Karthik Kotha, Abhiram Kotha and Sajjad Hussain, my true friends who were always together in spirit. A final note of thanks to Xylein, Stud and Shadowfax for keeping me sane.

Abstract

Focus, defocus and depth-of-field are integral aspects of a photograph captured using a wide-aperture camera. Focus and defocus blur provide critical cues for estimation of scene depth and structure which helps in scene understanding or post-capture image manipulation. Focus and defocus blur are also used creatively by photographers to produce remarkable compositional effects such as emphasis on the foreground subject with aesthetic bokeh in the background. Epsilon Focus Photography is a branch of computational photography that deals with the capture and processing of multi-focus imagery - where multiple wide-aperture images are captured with a small change in focus position. In this thesis, we provide a comprehensive study of various problems in epsilon focus photography along with a detailed analysis of the related work in the area. We provide useful constructs for the understanding and manipulation of focus, defocus blur and the depth-of-field of an image.

The work in this thesis can be divided into four broad categories of measurement, representation, manipulation and applications of focus. Measuring focus is a long studied and challenging problem in computer vision. We study various methods to measure focus and propose a composite measure of focus that combines the strengths of well-known focus measures. We the study the task of post-capture focus manipulation at each pixel in an image and formulate a novel representation of focus that can find much use in image editing toolkits. Our representation can faithfully encode the fine characteristics of a wide-aperture image even at complex interaction locations such as depth-edges and over-saturated background regions, while optimizing the memory footprint of multi-focus imagery. Apart from precise geometric constructs for scene refocusing, we also propose an data-driven approach for post-capture scene refocusing using deep adversarial learning. We show how the tasks of deblurring an image, magnification of the defocused content and overall comprehensive focus manipulation can be efficiently modeled using conditional adversarial networks. We study several applications of focus in computer vision such as view interpolation and depth-from-focus. We provide a tool that can interpolate different views of a scene based on focus texture segmentation and propose a novel solution for depth-from-focus using the proposed composite focus measure. In summary, this thesis consists of a comprehensive study of epsilon focus photography and its applications in the context of computer vision and computational photography.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The sense of sight is popularly regarded as the most crucial one of all human senses. Human vision along with perception of the world based on vision forms the primary source of interaction between human beings and their environment. It is therefore quite natural that photography - the process of 'recording' one's visual experience - is practiced and relished by mankind at large. With the advent of faster, smaller and more efficient optics and the development of versatile programmable devices such as smartphones, almost every individual is nowadays equipped with a powerful pocket camera.

Computational photography is a research area devoted to getting more out of the imaging process by the judicious use of computation at different stages. Methods in computational photography can be broadly categorized into two groups - methods based on novel optics and methods based on novel algorithms. Methods based on novel optics capture novel picture(s) of the scene (which may not make sense to a human) that may enable the estimation of additional properties of the scene such as 3D structure, illumination conditions, albedo, shading etc. Methods based on novel algorithms use standard picture(s) captured by existing cameras but process them differently to improve them or estimate more information about the scene.

The conventional pipeline of photography is described in Figure 1.1. The crucial elements in the pipeline are the illumination sources in the environment, the scene, the camera(s) and the processing involved in converting the light captured by the camera into a digital image. Computational photography relies on modifications to the illumination, constraints on the scene, modifications to the camera(s) or novel methods of processing the captured images in order to gather more information about the world. Computational photographic techniques result in great gains in the performance-to-complexity ratio for computer vision applications. Higher performance for vision applications generally requires higher complexity in camera design, and computational photography aims to reduce this requirement by adding computation at different stages of the photographic pipeline.

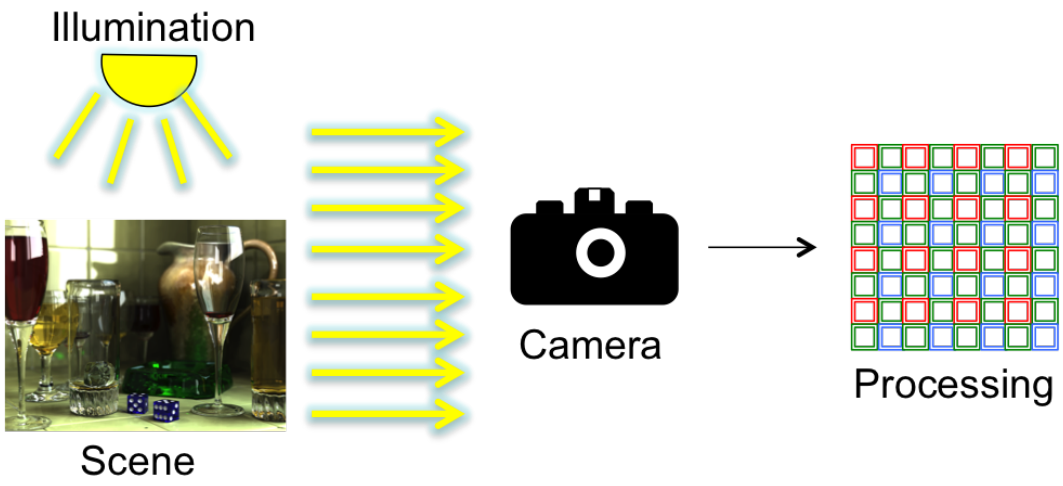

Figure 1.1 An image captured in the photographic pipeline is dependent on four elements - The source(s) of illumination in the scene, the objects in the scene, the imaging device and the processing pipeline that converts photo-intensities to a digital image.

1.1 Computational Photography with Novel Optics

Computational photography with novel optics - also referred to as coded optics - includes techniques in which the camera or the illumination source is modified. Methods such as structured illumination, object side coding, focal plane coding and sensor-side coding belong to this category. Figure 1.2 provides an illustration of the differences between the types of optical coding.

Illumination Coding is a technique in which the illumination sources in the scene are controlled synchronously with the camera. Illumination in the scene may either be provided in the form of single or multiple flash units or in the form of active projections using a digital projector. Illumination coding and structured illumination have been popular methods for depth and structure estimation of the scene. Some of the well-known techniques for scene structure estimation are Izadi et al. [45] - where the deformation in structured projections is used to estimate the structure of the scene and Zickler et al. [149] where the reciprocity between the camera and the projector is utilized for surface reconstruction. Illumination coding is also useful for several other applications in computational photography. Petschnigg et al. [99] describe a flash-no-flash technique for removal of specular artifacts along with denoising, detail transfer under low light conditions, white-balancing and red-eye removal. Raskar et al. [102] describe a method in which multiple separated flash units are used synchronously with a camera in order to provide accurate depth-edge measurements of the elements in the scene. Sun et al. 2007 [122] propose FlashCut, a foreground extraction method based on flash-no-flash image pairs based on the fact that the intensity of the illumination provided by a flash unit falls off exponentially with distance.

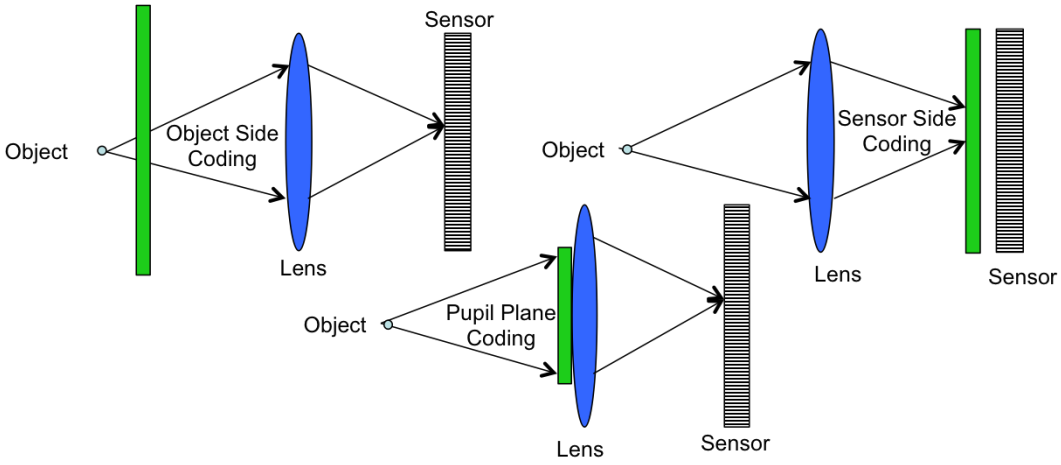

Figure 1.2 Three types of optical coding techniques. The position of the coding element (in green) with respect to the imaging pipeline defines the type of coding. In object side coding, all light rays falling on the lens are equally affected, in pupil plane coding, it is possible to accumulate differently at different points on the lens, while in sensor side coding, light is modulated just before it is collected on the sensor.

Structured illumination has also been explored using active projections from digital projectors. Several techniques simultaneously predict the 3D structure and the material properties of objects in the scene by estimating the light transport matrix between a projector and the camera [27, 94, 110, 111]. View synthesis from the point of view of the projector is also possible once the light transport matrix has been estimated. Fast and accurate 3D reconstruction using structured light has also been demonstrated in the past. Narasimhan et al. [86] describe solutions to vision tasks ranging from stereo to illumination-decomposition using an active temporal dithering framework built over a DLP projector. Gong et al. [30] show a method to estimate 3D structure from Fourier analysis of the fringe patterns generated by a modulated binary structured projection from a DLP projector. Large apertures of projectors lead to projection defocus, which has also been studied for better estimation of 3D structure. Zhang et al. [140] describe a depth estimation pipeline which works well at depth-discontinuities by explicitly modeling the projection defocus kernel. Gupta et al. [34] show the superiority in performance of depth estimation when projection defocus and global illumination of the projector is modeled appropriately.

Object-side coding, shown in Figure 1.2, deals with the introduction of optical elements in front of the camera such that the light falling on the surface of the lens is coded. Object side coding has been useful for a variety of computer vision applications such as deblurring, 3D reconstruction, multi-spectral imaging, etc. Raskar et al. [101] describe a shutter-modulation technique which leads to tractable models for motion deblurring. Georgeiv et al. [28] propose an optical element consisting of multiple lens-prism pairs to define an integral imaging framework for light-field photography. Du et al. [22] desribe a prism based method to greatly improve the spectral resolution of videos. Kuthirummal et al. [59] describe a radial imaging system by placing a curved mirror in front of the lens which is capable

of simultaneously estimating the geometry, texture and reflectance of the scene. Zhou et al [143] use an optical diffuser as the coding element for accurate 3D structure estimation while Cossairt et al. [20] extend this idea for the purpose of improving the depth-of-field of images.

Pupil plane coding, shown in Figure 1.2, is a technique in which an optical element(s) or modulation is performed at the surface of the lens. A spatially varying modulation across the lens surface is thereby possible. Coded aperture photography is the ideal example for pupil plane coding. Aggarwal et al. [4] describe a simple split aperture which can increase the dynamic range of the camera sensor. Levin et al. [64] describe a method for wavefront coding using a lattice-focal lens to improve the depth-of-field of captured images. Ashok and Neifeld [6] employ a pseudorandom phase mask in front of the aperture to remove the pixel-limited resolution of digital cameras dependent limited by impulse PSFs of light. Llull et al. [72] show how coded apertures can be used for temporal and spatial compression of optical information being captured by the camera. Zhou et al. [146] describe optimal coded aperture for removal of defocus blurs. Levin et al. [63] describe a coded aperture technique for simultaneous super-resolution and depth estimation.

Sensor-side coding, shown in Figure 1.2, is a form of optical coding where light modulation is applied right before the light falls onto the camera sensor. Narasimhan et al. [87] demonstrate sensor-side coding using assorted pixel-filters to modulate an imaging sensor for HDR imaging, while Yasuma et al. [139] extend the idea to propose post-capture control over spatial/spectral resolution and dynamic range using assorted pixel arrays. Light-field imaging involving groups of lenslets placed in front of the camera sensor can be classified into this category [91, 132], although this would not be the most typical sense of sensor-side coding. Independent pixel-wise exposure control for HDR imaging has also been described in the past [71, 88]. Sensor translation for extended depth-of-field imaging and motion blur removal has been described in [67, 82].

1.2 Computational Photography with Novel Processing

Computational photography with conventional cameras relies on the use of novel processing techniques to extract more information from standard images. Novel processing usually makes use of multiple images from the same or different cameras. Standard problems in computer vision such as stereo, structure-from-motion, panorama stitching etc. can be classified into this category. Image enhancement techniques based on color-tone adjustments/filtering, cartoonization, artistic renderings etc are also classified as novel processing techniques.

1.2.1 Epsilon Photography

Epsilon photography is the branch of computational photography which deals with novel processing of multiple input images that are closely related to each other. Epsilon photography involves the capture

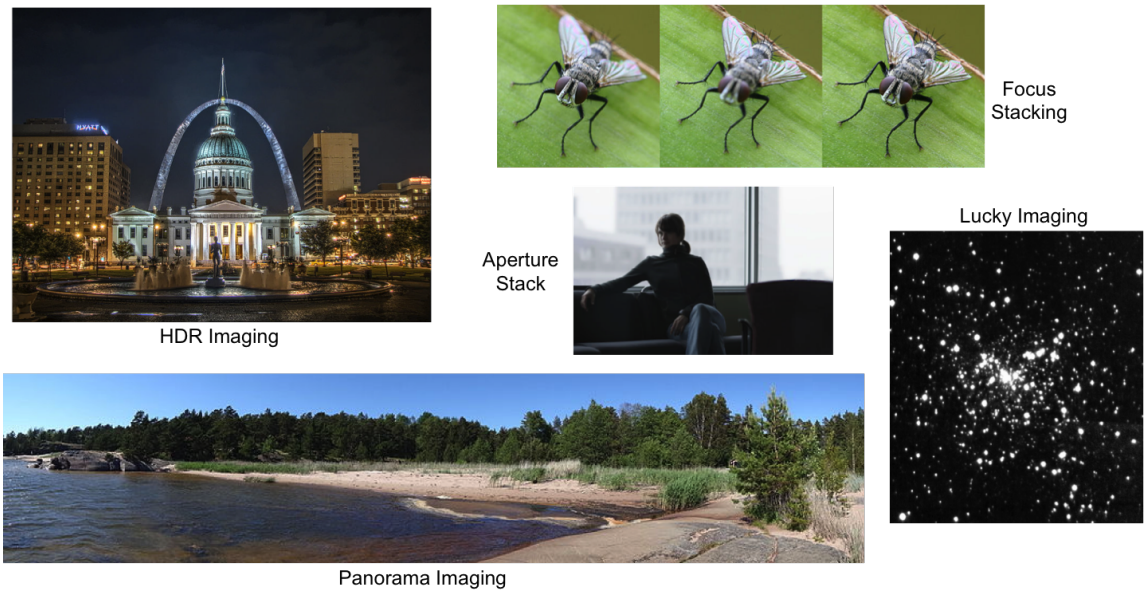

Figure 1.3 An illustration of different epsilon photography techniques that are popular in computer vision and photography. (Images from Wikipedia)

of a series of images of the same scene with an incremental small change 'ϵ' between consequent capture steps. The change is usually applied to a single camera parameter such as exposure time, aperture size, camera location, focus distance, etc. A composite image can be created from the individual slices. Composite images usually have better characteristics and reduced noise compared to a single slice and they are better equipped for inferring useful details about the scene.

The popular parameters for ϵ variation are focus, exposure, aperture and viewpoint. Figure 1.3 shows the different applications of epsilon photography. The epsilon variation in camera parameters may be applied over time, across different sensors or even across the pixels on the sensor having variable spectral characteristics. Exposure stacking deals with the capture of two or more images of the same scene with different exposure times, thereby capturing different scene details in different images. On fusing the images, a rich color image of the scene with a high dynamic range can be composited. Exposure stacking is also used for high quality capture of astronomical images, in a technique known as lucky imaging. Several tiny exposure bursts are captured for a significant time duration and after manually sifting through the captured images (astronomical images may be affected by atmosphere, pollution, stellar interference etc) the better ones are selected, shifted (to negate Earth's rotation between shots) and fused together into a composite image. Viewpoint stacking involves capturing multiple images with a small translation/rotation between adjacent shots. Camera rotation enables the miniaturization effect of tilt-shift photography as different parts of the scene come into focus at different columns of the sensor. Camera translation enables the creation of high resolution composite panoramas of the scene.

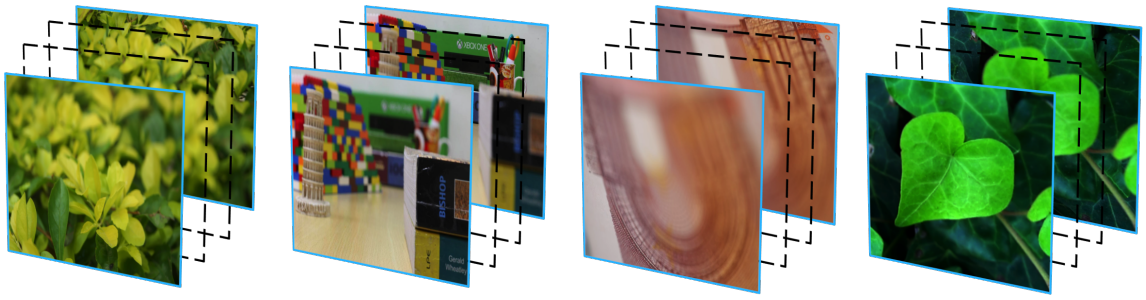

Figure 1.4 A few examples of focal stacks where the focus distance progressively varies from the near-end to the far-end of focus.

Focus and aperture stacking are closely related to each other and both have an impact on the depth-of-field of the scene. Aperture stacking can be used for improved light-throughput and post-capture selection of appropriate depth-of field. Focus stacking finds use for a variety of tasks such as view interpolation, depth estimation, post-capture focus control etc. In this thesis, the primary topic of study is focus stacking and its applications in computer vision and computational photography.

1.3 Epsilon Focus Photography

Epsilon focus photography or focal stacking involves the capture of multiple images of the scene with incrementally varying focus distance between consequent shots. The focal length, aperture size, exposure time, ISO and all other tertiary camera parameters are held constant across all the slices. The only change occurs in the focus distance of the shot, which is effectively the distance between the pole of the lens and the center of the camera sensor. The size of the aperture determines the depth-of-field for each slice and is also relevant for estimating the number of focal slices required to sweep from the near-end to the far-end of the scene. Figure 1.4 illustrates a few examples of focal stacks.

There are three main benefits of using focal stacks in place of standalone narrow-aperture images. Focal stacks, consisting of wide-aperture focal slices, have much better per-pixel focus information than a single narrow-aperture image as more light-rays contribute to the image of the scene object as shown in Figure 1.5. Focal stacks implicitly contain some proxy depth information about the scene since different scene objects come into focus at different focus distances. Focal stacks enable post-capture control of crucial parameters such as the defocus quality of lenses and size of depth-of-field, which are important aesthetic elements in digital photography.

We provide a broad overview of the different research areas related to these characteristics of focal stacks. The work in this thesis based on the listed themes and is defined formally in the chapters that follow.

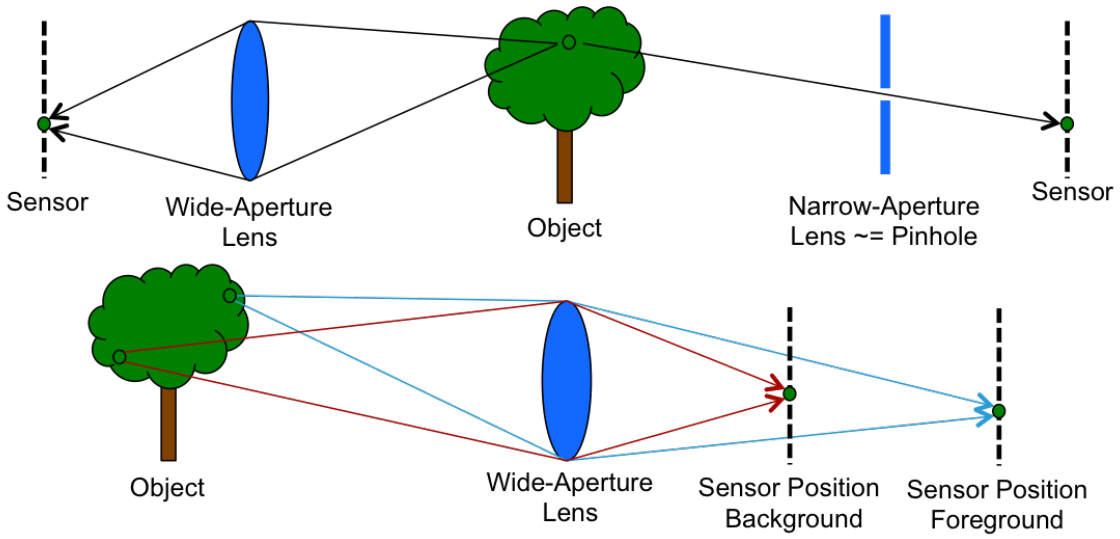

Figure 1.5 The benefits of focal stacks over single images. Top: Using a wide aperture camera allows more light rays to contribute to the image of a scene point and the image has better per-pixel focus information. A focal stack thereby captures all scene points with better clarity. Bottom: The selective focusing and defocusing of different scene points based on geometry provides useful cues for depth estimation. A focal stack thus provides proxy depth information.

1.3.1 Focal Stack Capture

Capturing the individual slices in a focal stack is a non-trivial problem. There are two distinct challenges while capturing a focal stack. The first is the time taken between successive focal steps. Traditional DSLR cameras are typically not programmable which creates a challenge for automatic control of the focus distance. A focal stack can therefore be captured only by manually changing the focus distance between shots. This can lead to inconsistencies in the conjunction of depths-of-field and also unneccesary camera motion between shots. Moreover, the time taken to capture a focal stack in this manner is in the order of minutes. A focal stack typically results in a loss of temporal resolution due to this finite amount of time taken to capture each slice and inference is thereby limited to mostly static scenes.

The second aspect is that a focal stack is expected to contain all scene points in focus in one and only one focal slice. An algorithm or protocol for focal-stack capture in which the consequent depths-of-field are connected to each other without any intersection is therefore a suitable choice. However, since the depths-of-field at macro distances are extremely small for wide apertures, such protocols might result in a large number of focal slices. In the absence of any scene points in macro range, it is not necessary for those slices to be captured at all. Therefore, optimal focal stack capture requires that the depth distribution of the scene is analyzed first and only the necessary focal slices are captured afterwards. Several early works in epsilon photography provide a computational solution to focal-stack capture.

Nowadays, programmable devices are more commonly available and capturing a focal stack is slowly becoming an accessible tool at large.

1.3.2 Measuring Focus and Defocus Blur

Understanding the scene using a focal stack requires that the degree of focus or the degree of defocus blur is measured accurately at all the pixels across the focal stack. The focus at each pixel can be modeled using a measure of focus. Many measures of focus have been proposed in computer vision literature. A focus measure basically estimates the amount of sharpness in the neighborhood of the pixel to determine the amount of focus. There is no consistently best focus measure, as the performance of a measure is usually scene dependent. Focus can also be studied by modeling the degree of variation of defocus blur across the stack. Focus and defocus modeling has been used extensively in the past for estimating the depth map of the scene. Modeling the defocus blur using two or a few slices is known as depth-from-defocus. Modeling the depth using a focus measure across many focal slices is referred to as depth-from-focus.

Measuring spatially-varying defocus kernels involves an additional layer of complexity where the nature and overlap of multiple kernels from different depth results in complex pixel compositions. The challenges in measuring defocus at a pixel are centered around the fact that defocus kernels merge with each other in complicated arrangements based on the scene-depth of pixels that are adjacent to each other in a two-dimensional view. The size and shape of defocus kernels as well as intensity saturated scene regions need to be measured for appropriate analysis and rendering.

1.3.3 Scene Refocusing

Reconstructing the in-focus scene content from focal slices is one of the the popular application areas for focal stacks. An in-focus image of the scene is free of any ambiguity caused by defocus blur for computer vision tasks such as segmentation, recognition and retrieval. Focus and defocus blur are used creatively by photographers for bringing out the emphasis and contrast in an image. The shape of defocus blur, also known as bokeh, is of importance to the creative process of image conceptualization and realization. Post-capture control of these aspects is undoubtedly a very useful tool for both professional and amateur photographers alike. Such a tool can significantly reduce the compositional burden during capture. However, for aesthetic acceptability, post-capture focus manipulation must be capable of simulating all scenarios that can be captured by a real camera. In other words, the quality of bokeh and the transitions between focused and defocused regions should be as photorealistic as possible. Moreover, the performance of such an approach needs to be close to real-time and thereby a compact representation of focal stacks for post-capture manipulation may be useful. This is also relevant because storing a full focal stack on the camera can result in large storage overheads for portable devices.

1.3.4 Focal Stacks for Vision Applications

Focal stacks contain proxy depth information and more informative pixels with reduced noise in dimly light areas. Using focal stacks in place of active depth cameras has been proved useful for several vision problems where scene depth is required. Using scene depth, interpolation of novel views adjacent to the captured view is also possible. Other methods such as learning based denoising, learning based deblurring, intrinsic image decomposition etc. can all benefit from the cues provided by focal stacks.

1.4 Contributions of this thesis

In this thesis we provide a comprehensive analysis of focus, defocus blur and depth-of-field in multi-focus imagery captured by wide-aperture cameras. From capture to measurement and from measurement to manipulation, we study several aspects of epsilon focus photography in the context of scene understanding and post-capture photo-manipulation. As a primary contribution, we analyse the large corpus of focus measures proposed in the literature and devise a novel composite measure of focus that combines the strengths of existing focus measures. We show that using a composite measure not only resolves the ambiguities caused by a single measure but also helps in generalization of the process of focus measurement for diverse scenes. The composite focus measure is the central theme of our enquiry.

Using this composite measure of focus, we build a representation for focal stacks. Focal stacks by definition are volumes of images that store multiple captures of the same scene under different focus settings. If the relevant information contained within a focal stack can be estimated and stored within a compact representation, there may be no need to store the entire focal stack. We identify the critical characteristics of wide-aperture images - including some fine aspects that have never been considered before in computer vision - and build a compact yet efficient representation for multi-focus imagery. We show that our representation is suitable for precise reconstruction as well as photo-realistic rendering of novel focused images of the scene, using a geometry-preserving algorithm for refocusing.

With data-driven methods gaining more and more popularity in the recent days, we also explore the task of comprehensive scene refocusing using deep learning. We study conditional adversarial learning for the task of photo-realistic scene refocusing and defocus magnification. We find that with the right training paradigm and appropriate priors, data-driven techniques provide powerful tools for complex tasks such as single-image scene refocusing. Motivated from a large corpus of focal stacks created from light-field images, we build a comprehensive dataset of focal stacks that help us train robust models for data-driven focus manipulation.

We also study the impact of focal stacks for computer vision applications such as depth-from-focus. We show that the composite focus measure captures of a rich but sparse representation of the focus content within the scene and devise a strategy to propagate depth information from sparse estimates to

the entire image. We also study the problem of geometric view interpolation using sparse multi-focus imagery captured at quantized intervals in the scene.

In the upcoming chapters, we first discuss the related work in computational photography that has motivated the efforts in this thesis and then provide a detailed account of our broad contributions to epsilon focus photography.

Chapter 2

Background & Related Work

The measurement and manipulation of focus and defocus blur are well-studied topics in computer vision and are relevant because many real-world images are captured from wide-aperture cameras. Focus and defocus blur provide crucial clues for understanding the depth distribution of the objects in any scene. Semantic scene understanding however requires the deblurring of defocus scene regions for optimal performance. Focus and defocus blur are also studied for the creative applications in photography such as portraiture - or adding bokeh - which is a common feature in today's smartphones. In this chapter we provide a detailed account of the contemporary work in computational photography that has inspired and preceded the work presented in this thesis.

2.1 Epsilon Focus Photography

Capturing a scene with a *epsilon* change in focus position between consequent shots results in a focal stack of the scene. This sequence of images provides several cues that can aid scene depth estimation and free-form scene refocusing. A dense model of the focus profile for every scene point can be computed from a focus stack. These focus profiles can be used for deblurring, refocusing and depth estimation. Hasinoff et al. [38, 37, 60] show the overall capture time for a target depth-of-field is significantly reduced by capturing that region in a piecewise manner using multiple wide-aperture images as a focal stack (Figure 2.2). Focal stacks are also shown to exhibit reduced noise characteristics than a single image capturing the full depth-of-field.

Capturing a focal stack, however, is a challenging task and often requires specialized hardware or access to firmware-level APIs for the camera in question. Early efforts in focal stacking often involved tedious computational photographic setups to move the sensor plane between consequent shots such as the focal sweep camera described by Zhou et al. [144] and shown in Figure 2.1. Over the years, several tools have become available for easier capture. Nowadays, it is possible to program automatic focal stack capture on certain DSLRs and iOS or Android mobile devices. This significantly improves capture time and consistency between focal slices. Mobile devices can be programmed to capture multiple focused images sequentially or using region-based focus stacking [106]. MagicLantern [75] provides control on

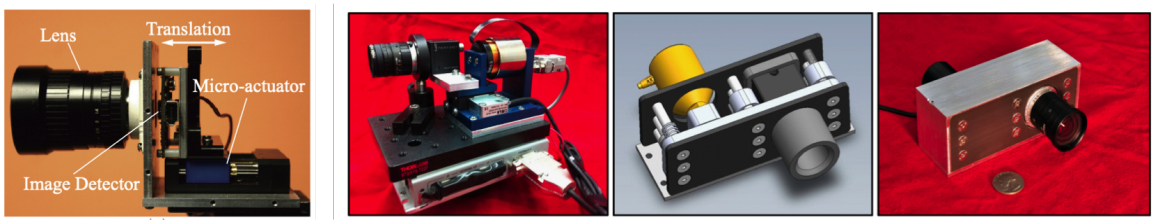

Figure 2.1 Left: The focal sweep camera from Nagahara et al. [81] consisting of an arrangement where the sensor translates during capture. Right: The focal sweep camera from Zhou et al. [144] for capturing *breathing* pictures.

Canon DSLRs to configure focus limits and precisely control the movement of the focus ring between consecutive slices.

2.2 Depth-of-Field Extension

Extending the depth-of-field of an image to generate an in-focus image where all pixels are within an acceptable level of sharpness is one of the primary applications of epsilon focus photography. Kubota et al. [53] generate in-focus images using linear filtering of focal slices using a focal-texture model. Kodama and Kubota [51] extend this to a 3D filtering method for generating dense pin-hole views from novel viewpoints. Iterative computation of focal textures can be used to produce in-focus images for shallow focal stacks [106]. Nagahara et al. [81] use a special camera to capture a focal sweep of the scene by aggregating the light on the sensor during relative translation between the sensor and the lens. Deconvolution of the blurred focal sweep image yields an all-in-focus image as shown in Figure 2.3. Kuthirummal et al. [58] use the same principle and average the captured focal slices to generate the integrated image from a focal stack. Agarwala et al. [3] generate an in-focus image from a focal stack using a global contrast maximization approach. Other works such as [56] are also targeted towards estimating the sharpest representation of each pixel across multiple focal slices.

2.3 Measuring Focus and Defocus Blur

Measuring focus and defocus blur remains the central and most challenging problem in epsilon focus photography. Measures of focus based on different image properties have been proposed in computer vision literature [89, 42, 80, 112, 138] in the past. The response of these measures not only depends on their parameters but also on the textural content around the pixel in the image. Measuring per-pixel focus based solely on a single pixel's response is usually noisy and unreliable [144]. Smooth focus maps that consider neighborhood consistency can be generated using optimization methods such as Cost-Volume Filtering [103] or MRF labeling [14]. Again, iterative optimization methods can also be used to estimate smooth focus maps [106, 55]. Pertuz et al. [98] analyze and compare several focus measures independently for the task of depth-from-focus. They conclude that Laplacian based operators are best suited

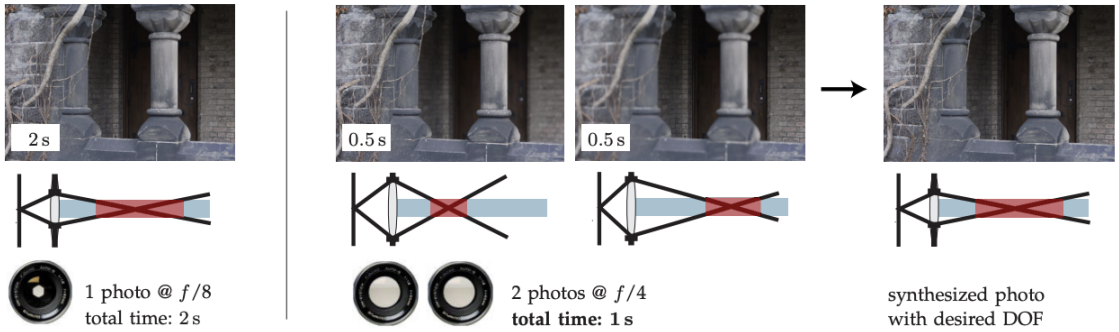

Figure 2.2 The overall capture time and signal-to-noise ratio for each pixel is improved on capturing a scene in a piecewise focal stack rather than capturing a single long-exposure narrow-aperture image as illustrated in this figure and discussed in [38, 37, 60].

under normal imaging conditions. In [79], the Laplacian focus measure is used to compare classical DfF energy minimization with a variational model. A Ring Difference Filter measure is proposed in [123], with a filter shape designed to encode the sharpness around a pixel using both local and non-local terms. Mahmood et al. [76] combine three well known focus measures (Tenengrad, Variance and Laplacian Energy) in a genetic programming framework. Boshtayeva et al. [13] describe the benefit of using multiple focus measures together in an anisotropic smoothing framework to compute scene depth. Measuring the focus information of a scene is in fact analogous to computing relative scene depth from multiple focused images [89, 18, 26, 97, 11, 124].

Modeling the spatially varying defocus kernel in wide-aperture images, or measuring defocus, is relevant in the context of image de-blurring and scene refocusing. Kee et al. [49] remove spatially varying optical blur by estimating dense non-parametric blur kernels across the image. They describe calibration and kernel fitting methods for blur estimation. Shih et al. [116] show a calibration technique to predict the lens point-spread-function (PSF) at arbitrary depths using a calibrated PSF at a known depth. Tang and Kutulakos [127] describe an analytical approach for blind image de-blurring and PSF calibration at other pixels from known PSFs at some pixels. Hach et al. [35] show a dense calibration method for a high quality lens with depth-aware rendering to produce synthetic bokeh.

2.4 Scene Refocusing

Controlling the focus position of an image after capture has also received attention in computer vision. Nagahara et al. [81] demonstrate flexible depth-of-field rendering by capturing disjoint focal sweeps through the scene and deconvolving the aggregate image by an integration kernel. Half-sweep imaging [78] has also been used on similar lines. Jacobs et al. [46] describe a precise composition model to generate composite images from the available focal slices using a defocus map that specifies the amount of target defocus at each pixel. They use geometric integration of rays for free-form depth-

of-field control. The Lytro lightfield camera [92, 73] enables post-capture refocusing and interpolation by capturing a 4D lightfield at a limited resolution.

Jacobs et al. [46] propose a geometric approach to refocusing and create refocused images by appropriately blending pixels from different focal slices, while correctly handling halo artifacts that result from double-counting of rays. Hach et al. [36] model real point-spread-functions between several pairs of focus positions, using a high quality RGBD camera and dense kernel calibration. They are thereby able to generate production-quality refocusing with accurate bokeh effects. Suwajanakorn et al. [124] compute the depth-map of the scene from a focal stack and then demonstrate scene refocusing using the computed depth values for each pixel. Several methods have been proposed in the past to compute in-focus images and depth maps from focal stacks [8, 79, 81, 107, 124]. Most of these methods enable post-capture control of focus but use all the images in the focal stack. Zhang and Cham [142] change the focus position of a single image by estimating the amount of focus at each pixel and use a blind deconvolution framework for refocusing. Methods based on Bae and Durand [105] also estimate the per-pixel focus map but for the task of defocus magnification. These methods are usually limited by the quality of the focus estimation algorithm as the task becomes much more challenging with increasing amounts of blur.

Deep neural networks have been used in the past for refocusing light-field images. Wang et al. [134] upsample the temporal resolution of a light-field video using another aligned 30 fps 2D video. The light-field at intermediate frames is interpolated using both the adjacent light-field frames as well as the 2D video frames using deep convolutional neural networks. Any frame can then be refocused freely as the light-field image at each temporal position is available. Full light-fields can themselves be generated using deep convolutional neural networks using only the four corner images as shown in [48]. A full 4D RGBD light-field can also be generated from a single image using deep neural networks trained over specific scene types as shown in [118]. Srinivasan et al. [117] implicitly estimate the depth-map of a scene by training a neural network to generate a wide-aperture image from an in-focus radiance image. These methods suggest that it is possible to generate light-fields using temporal and spatial interpolation. However, these methods have not been applied for focus interpolation.

Deep neural networks have been used for deblurring an input image to generate an in-focus image. Schuler et al. [109] describe a layered deep neural network architecture to estimate the blur kernel for blind image deblurring. Nimisha et al. [93] propose an end-to-end solution to blind deblurring using an autoencoder and adversarial training. Xu et al. [137] propose a convolutional neural network for deblurring based on separable kernels. Nah et al. [83] propose a multi-scale convolutional neural network with multi-scale loss for generating high-quality deblurring of dynamic scenes. Orest et al. [57] show state-of-the-art deblurring for dynamic scenes using a conditional adversarial network and use perceptual loss as an additional cue to train the deblurring network.

Defocus magnification of portrait photos is a well studied problem in computer vision. With the abundance of mobile devices equipped with high quality narrow-aperture cameras, blur magnification or portraiture has become a popular tool in photo-sharing and social media applications. Several hardware

Figure 2.3 Deconvolved extended depth-of-field image from Nagahara et al. [81] where all pixels are reconstructed to be in-focus.

modifications have also been implemented in mobile cameras to estimate per-pixel depth specifically for blur magnification. Bae and Durand [7] proposed defocus magnification as a two-stage task of defocus estimation followed by amplification. The bluriness (defocus map) is estimated at intensity edges by fitting the best Gaussian blur kernel to pixel intensities. The blur map is then refined using bilateral filtering and homogeneously propagated to all other pixels in the image. Tai and Brown [125] propose a local contrast prior for defocus estimation which models defocus blur as the ratio between the gradient and the contrast of intensities surrounding the pixel. This is based on the observation that with increasing blur, local gradients become smaller than the local contrast due to smoothing. The estimated defocus blur is propagated through the image using an MRF optimization. Methods that are based on defocus estimation and amplification share the benefit that the depth-map and the texture-less regions in the scene need not be estimated explicitly.

Zhou and Sim [148] estimate the per-pixel defocus in an image by re-blurring the image with known blur radii and compute the unknown blur using the gradient ratio of the re-blurred images. They estimate sparse blur radii at intensity edges and blur values are propagated through the image using the edge-aware matting Laplacian approach of [65]. Zhu et al. [147] use blur-spectrum fitting to estimate a probability distribution over the scale of the point-spread-function at all pixels which is optimized using gradient-aware smoothness terms. Chen et al. [19] magnify the defocus in portrait images with foreground enhancement using a face prior for foreground detection. Recent work has also focused on estimating small blurs in images captured by point-and-shoot cameras. Shi et al. [113] estimate small

Figure 2.4 Scene refocusing from a single image is typically achieved in a two-stage manner with deep neural networks used to first estimate scene depth distribution and then render a novel depth-of-field. Depth-of-field control over a single image from Wang et al. [70] is illustrated here.

scale defocus blurs using non-parametric matching and edgelet primitives which model intensity edges with varying directions, curvatures and scales. Shi et al. [115] estimate just-noticeable-blur using sparse reconstruction statistics based on the observation that clear and slightly blurred areas are composed of visually different dictionary patches.

Deep neural networks have also been used for estimating the per-pixel defocus kernel in a single image. Ma et al. [74] use a fully convolutional neural network trained on the discriminative blur detection dataset of Shi et al. [114]. They show state-of-the-art blur detection and estimation for several applications including defocus magnification. Park et al. [95] use a combination of defocus measures and deep features in order to estimate the defocus at all pixels in an image. The sparse defocus map obtained using a combined defocus measure is propagated to the full image using the matting Laplacian [65] approach.

Defocus magnification has also been shown in conjunction with methods that compute depth-maps from single images. Namboodiri and Chaudhuri [85] solve a stabilized reverse heat equation to estimate the depth-map of the scene characterized by the amount of defocus at each pixel. Barron et al. [10] propose a stereo based approach with disparity inference in bilateral space for high quality depth maps and consequently render shallow depth-of-field images. Srinivasan et al. [117] create a large dataset of aperture stacks to train a monocular depth-estimation network on images of flowers and plants. Defocus magnification is the ideally suited application for their technique. Several mobile devices are now equipped with dual cameras and primarily use stereo-based depth information to simulate a shallow depth-of-field. The Pixel 2 cameras [100, 133] use dual pixels on the sensor that generate stereo images at small baselines of $1mm$ which are useful for depth-aware defocus magnification and a variety of other applications. Wang et al. [70] use deep-neural networks for post-capture control over the depth-of-field. They estimate scene depth from a single image and then apply a lens-blur network to simulate the target depth-of-field.

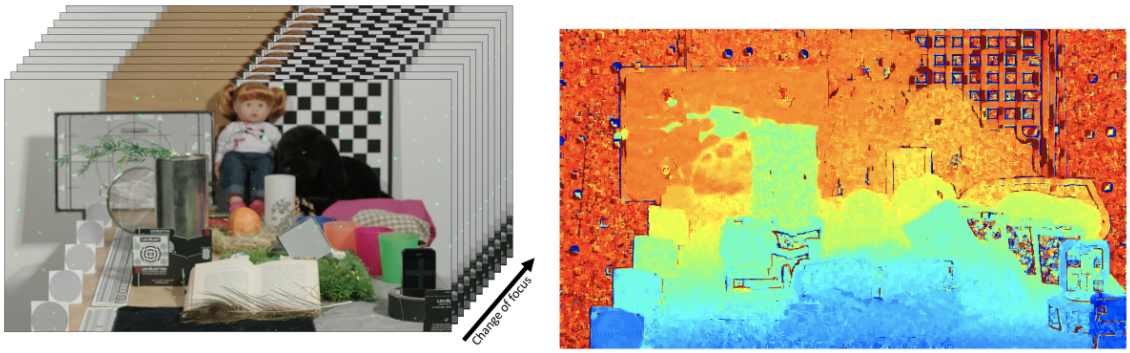

Figure 2.5 Focal stacks contain useful proxy cues for dense depth estimation. A dense focal stack and its corresponding scene-depth computed in [79] is illustrated here.

2.5 Focal Stacks in Computer Vision

The computation of depth from multiple focused images is a popular application of focal stacks [9, 12, 79, 124]. Defocus cues have also been used [11, 18, 26, 78, 89, 97, 126] to estimate scene depth. In most methods, depth is estimated from the peak focus slice computed using per-pixel focus measures. In [79], the Laplacian focus measure is used to compare classical DfF energy minimization with a variational model. A dense depth representation is computed from the focal stack as shown in Figure 2.5. The RDF focus measure proposed in [123] has a filter shape designed to encode the sharpness around a pixel using both local and non-local terms for computing depth. Mahmood et al. [77] combined three well known focus measures (Tenengrad, Variance and Laplacian Energy) in a genetic programming framework for depth-from-focus. Boshtayeva et al. [12] described anisotropic smoothing over a coarse depth map computed from focal stacks. Suwajanakorn et al. [124] proposed a joint optimization method to solve the full set of unknowns in the focal stack imaging model.

2.6 Contributions of this Thesis

The work presented in this thesis spans a broad range of contributions related to the different applications of epsilon focus photography discussed in this chapter. The work is related to focal stack capture on mobile devices, measuring and estimating the focus profile across a focal stack in a robust manner, representing a focal stack in an efficient yet compact representation, a learning based approach for post-capture scene refocusing and several applications of focal stacks such as depth-from-focus and dense view interpolation using image based rendering.

Chapter 3

Measuring Focus

Measurement of focus or defocus blur is one of the most challenging problems in epsilon focus photography. Multiple wide-aperture images of the scene - in the form of a focal stack - consist of the same scene points appearing with a different degree of focus or defocus blur across the focal slices. In order to study the appearance of these scene points for tasks such as estimating their depth, the true color or for re-rendering the scene, the first requirement is to accurately estimate of the amount of focus at each pixel across the stack. The sharpness of the image content across the two-dimensional neighborhood of a pixel is a reliable indicator of its amount of focus. However, identifying the appropriate measure of sharpness in a scene- independent manner is a non-trivial problem. Figure 3.1 shows the response of different measures of focus/sharpness at two pixels in a focal stack. It can be seen that although all these measures encode the sharpness around a pixel, they exhibit remarkable variability. Furthermore, highly blurred bokeh circles typically have sharp gradient edges which end up complicating the measurement problem to another degree. In this chapter, we seek to identify a composite focus measure (cFM) as a weighted combination of existing measures that estimates the amount of focus at a pixel more accurately than any single measure.

There are two that factors critically affect focus estimation - the quality of a focus measure and its region of support. No single focus measure works well in all situations, irrespective of whether it uses statistical, spectral, gradient, or other properties of the pixel neighborhood. The response of a focus measure depends significantly on the underlying scene structure and intensities. For most focus measures, the size of the region of support plays an important role in the identification of the focus peak. Smaller regions usually have high specificity, but noisy estimates. Larger neighborhoods provide stable estimates but cause dilation across depth edges.

Pertuz et al. [98] analyzed 36 different focus measures individually to characterize their sensitivity with respect to support window size, image contrast, noise and saturation. Their analysis provided no definitive recommendation about the best focus measure as different measure exploit different properties and perform well on varying scene types. This conclusions suggests that a combination of FMs can work well for more varied and general situations. A composite or combined focus measure, is the central theme of our enquiry in this chapter.

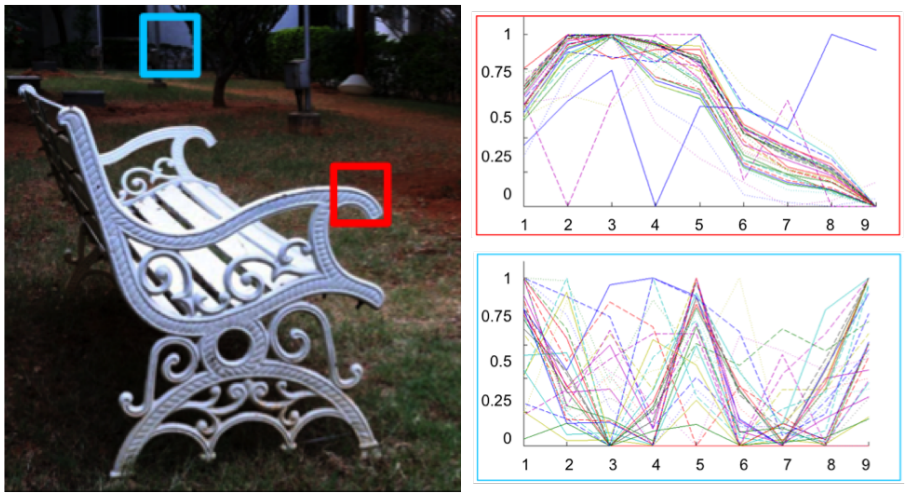

Figure 3.1 The focus profiles of thirty different focus measures evaluated at two pixels across a focal stack. Severe variability in in-focus estimation across different measures can observed based on the textural content around challenging pixels.

3.1 Composite Focus Measure

We study the performance of 39 focus measures - all measures from Pertuz et al. [98], two additional measures from Boshtayeva et al. [13] and the Ring Difference Filter from [123] - in the context of per-pixel focus estimation over a large dataset of focal stacks. We represent the focus measures using a similar convention to [98]; the additional measures are labeled as HFN (Frobenius Norm of the Hessian), DST (Determinant of Structure Tensor) and RDF (Ring Difference Filter). We compute focus profiles shown in Figure 3.1 for all the pixels in a large focal stack dataset across all focus measures. We seek to select the best subset of focus measures using these focus profiles.

Selecting the best subset of focus measures from a large number of them is a challenging problem. Supervised approaches with principled learning of weights for a composite focus measure are not feasible, due to the lack of ground truth data defining the focus peak for every pixel in a focal stack. Capturing large number of aligned focal stacks and depth maps can enable supervised learning of FM weights or the use of deep learning methods to directly come up with a robust composite measure.

In the absence of ground truth depth, unsupervised feature selection is the natural candidate for FM selection. Unsupervised methods use unified learning frameworks that simultaneously estimate the structure of the data and the best set of features that describe the data [16, 23]. However, selecting the best combination of focus measures is different from the feature selection problem. In feature selection, the goal is to identify the best subset of representative features which define the data well, and each selected feature usually encodes diverse information about the underlying data. The selection process thereby maximizes diversity between individual features. For the selection of focus measures, all the features represent the same information - the amount of focus at a pixel. Therefore, the agreement of different focus measures is important.

Traditional methods for unsupervised feature selection of focus measures [16, 23] perform poorly for DfF (Figures 3.7, 3.8), as expected. The top-ranked measures according to [16] exhibit different focus peaks at most pixels, since FMs with diverse responses are selected. For accurate focus measurement, it is important to select those FMs which agree with one another. The agreement of different focus measures about the peak location of focus is of importance because it suggests a consensus in identification of sharp content around the pixel. However, since we use a diverse collection of FMs, some FMs may give near identical responses to others. These have redundant information and may not be very useful together, even though they agree about the focus peak. Measures that agree on the focus peak but not at other slices should ideally be part of the composite focus measure. Thus, we seek to identify focus measures that demonstrate *high consensus* but *low correlation*.

In the upcoming sections of the chapter we describe our strategy to compute the composite focus measure by looking for high-consensus FMs which do not have high correlation. We also first describe the dataset that is used for the computation of the composite focus measure.

3.1.1 Focal Stack Dataset

To compute the composite focus measure, we use a large corpus of 320 focal stacks curated into a single focal stack dataset. This dataset is composed of 100 focal stacks from the light-field saliency dataset [69], 80 sparsely sampled stationary focal stacks from the Autofocus dataset [1], 20 focal stacks that are available publicly across different works [13] and around 120 focal stacks that we captured using DSLR and mobile cameras. A snapshot of the different images in this dataset is provided in Figure 3.2. Our DSLR focal stacks are captured using Magic Lantern on a Canon EOS 70D and a Canon EOS 1100D. Our mobile focal stacks are captured on an iPhone 6 and an HTC One X. The dataset includes a variety of scenes with differing illumination, variable scene-texture and a broad range of magnitudes of defocus blur. Each focal stack is rescaled to the order of 1 million pixels while preserving the aspect ratio. All experiments are performed on RAW images (or on images rescaled by the appropriate gamma factor). In multi-focus imagery captured in the form of focal stacks, image magnification due to the translation of the sensor relative to the lens results in misalignment of pixels across the stack. To eliminate these pixel misalignments due to magnification, we use the enhanced correlation coefficient maximization approach [24]. This approach simulates a zoom-and-crop behaviour after estimating an affine transformation across consequent slices. The result of using such an approach is a pixel aligned focal volume. To compute the composite focus measure, the response of each focus measure is averaged across three different region of support sizes 3×3, 7×7 and 11×11. The cumulative scores for each focus measure across thereby computed across a corpus of about 1.2 billion pixels across the 320 focal stacks.

3.1.2 Consensus of Focus Measures

To measure the consensus among focus measures we first estimate the mean peak focus position at each pixel. The measures that peak within a small neighborhood of this mean position are then

considered to be in consensus with each other. For the mean focus position, we use all measures to build a coarse focus map using MRF based energy minimization [14]. The data cost $D_L(p)$ of labeling the mean focus position of a pixel p to focal slice index L is computed as the sum of normalized measure responses at the pixel:

$$D_L(p) = exp\left(-\sum_{j=1}^{N_F} \frac{F_j(p,L)}{\sum_l F_j(p,l)}\right), \tag{3.1}$$

where $F_j(p,L)$ is the j^{th} focus measure applied at pixel p for the L^{th} focal slice and N_F is the total number of measures. A multi-label Potts [14] term is used to assign smoothness costs.

The result of MRF optimization for every focal stack is a neighborhood-aware mean focus position for all its pixels. The consensus for each focus measure is now recorded as the agreement of the measure with this mean focus position within a small margin. The consensus C is computed as

$$C(F_j;p) = \left\{ \begin{array}{ll} 1 & \text{if } \arg\max_l F_j(p,l) \\ & \in [i(p)-w, i(p)+w] \\ 0 & \text{otherwise} \end{array} \right\}, \tag{3.2}$$

where $i(p)$ is the label assigned after MRF optimization at pixel p and w denotes a small neighborhood around $i(p)$. We select w as 10% of the number of focal slices in the stack. This corresponds to a small depth neighborhood as the focus steps in the focal stack dataset are mostly uniform. w can also be parameterized based on the blur difference between two slices in case of non-uniform focus steps. The cumulative consensus score for a measure is the sum of its consensus at pixels across all focal stacks used in our analysis:

$$\hat{C}(F_j) = \sum_{FS} \sum_{p} C(F_j;p), \tag{3.3}$$

where FS represents all the focal stacks in the dataset and p represents the pixel in each stack. Figure 3.3 lists all measures used in our analysis, ranked according to their normalized cumulative consensus score $\hat{C}$.

Alternatively, we also compute max-consensus, which measures whether the peak location agrees with focal slice at which most focus measures peak. The focus measures that peak within a small neighborhood of this slice are assumed to be in consensus. The C_{max} function computes max consensus as:

$$C_{max}(F_j;p) = \left\{ \begin{array}{ll} 1 & \text{if } \arg\max_l F_j(p,l) \\ & \in [m(p)-w, m(p)+w] \\ 0 & \text{otherwise} \end{array} \right\}. \tag{3.4}$$

Here $m(p)$ is the focal slice at which maximum number of measures peak for pixel p, $F_j(p,l)$ the j^{th} focus measure response at pixel p of slice l and w denotes a small neighborhood around $m(p)$. We find that MRF based consensus is a more robust indicator of consensus as it considers the pixel neighborhood in a relatively global manner.

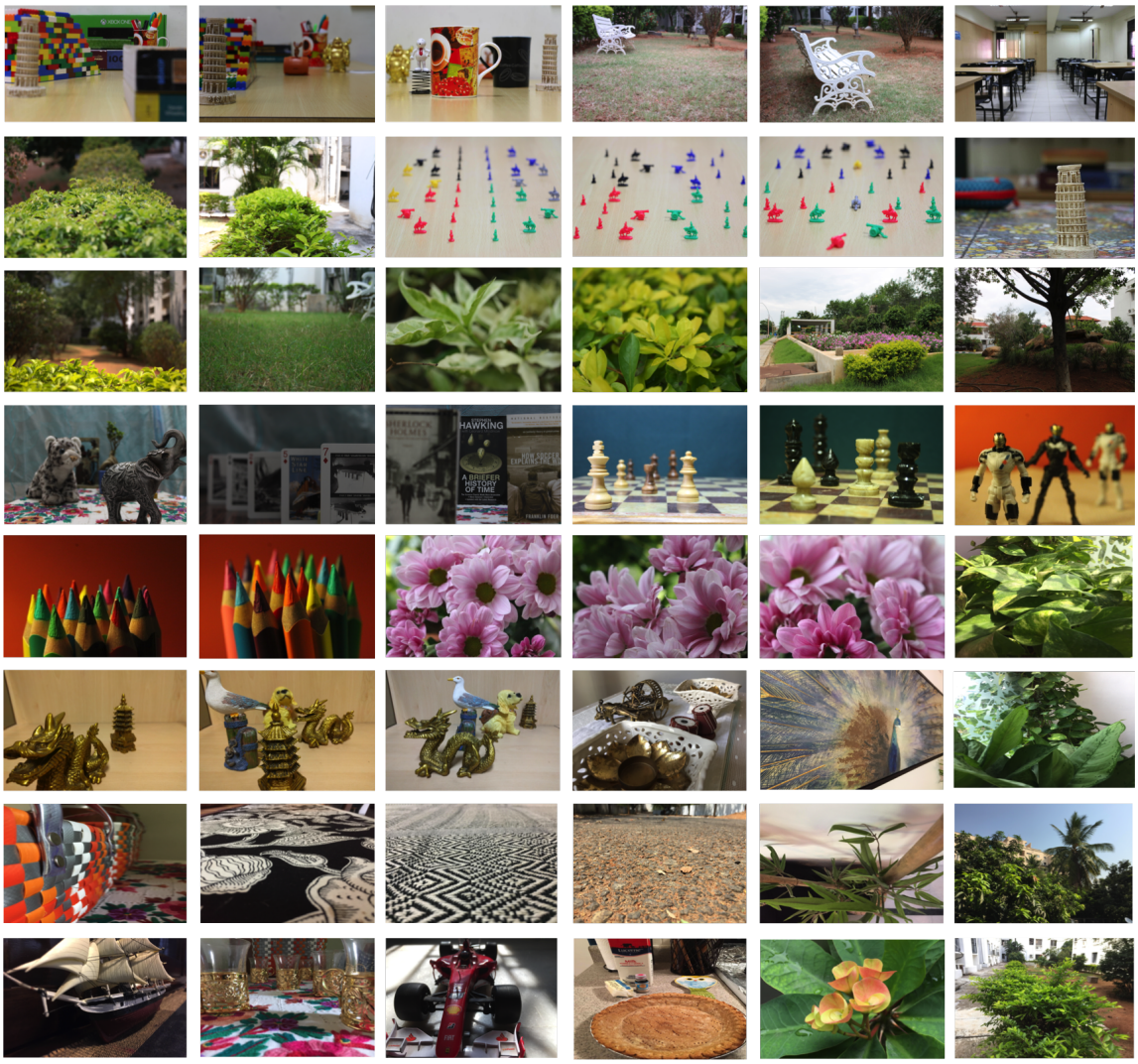

Figure 3.2 A snapshot of the different scenes in our focal stack dataset that are used to compute the composite focus measure.

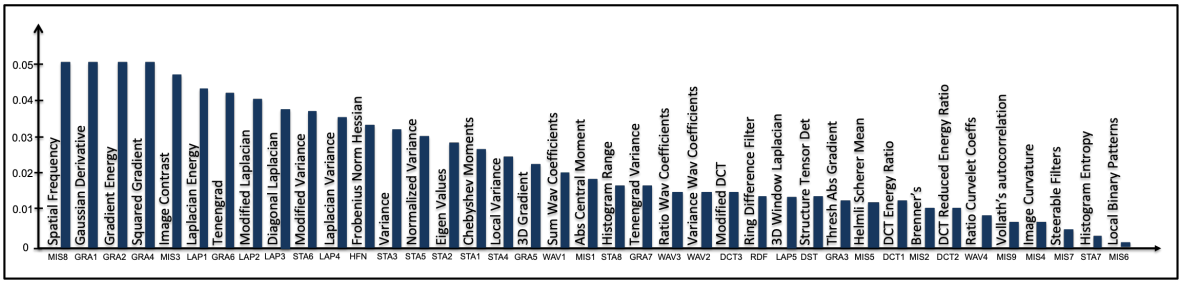

Figure 3.3 A ranked list of all focus measures used in our analysis. The list is sorted in descending order of the normalized cumulative consensus score $\hat{C}$ computed over our focal stack dataset.

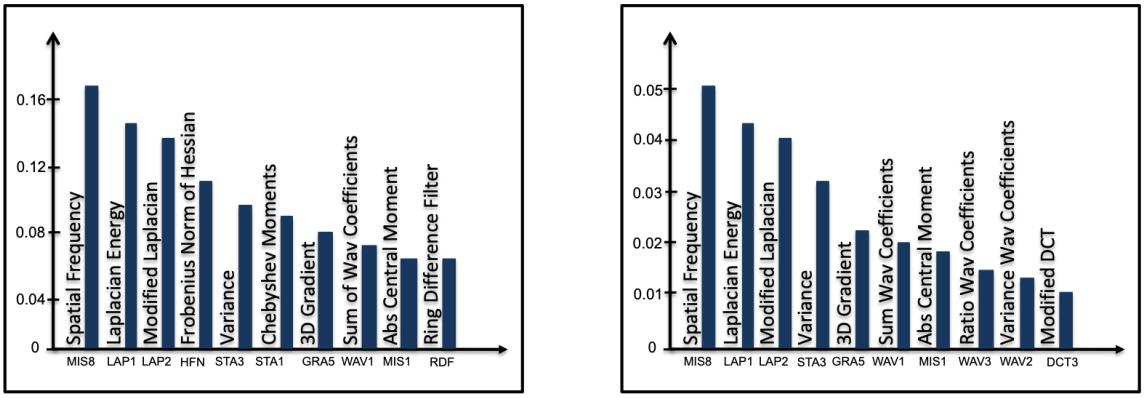

Figure 3.4 Top 10 focus measures with a high degree of consensus but not high correlation. The normalized consensus score is shown on the Y-axis. This score is used as the weight for creating the composite focus measure. On the left is the cFM is computed using the heuristic thresholding based approach while the cFM computed using Hierarchical Agglomerative Clustering is on the right side.

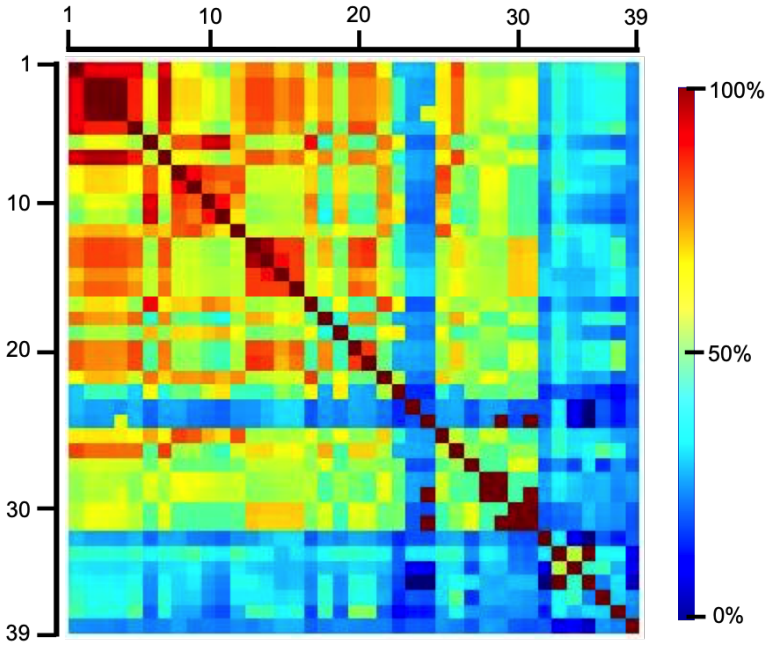

Figure 3.5 Percentage pairwise correlation between pairs of focus measures (best viewed in color). The FMs from left-to-right and top-to-bottom follow the same ordering shown in the ranked consensus list of Fig. 3.3. FMs in dark shades of red indicate high correlation. It is evident that the first five measures in descending order of consensus are all highly correlated with each other. We would like to retain only one measure from these.

3.1.3 Correlation of Focus Measures

The focus measures used in our analysis contain near-identical or highly correlated measures. These will naturally be in consensus with each other but add little additional value. Only one of each highly correlated set of measures is a useful choice for the composite focus measure. We compute pairwise correlation between focus measures across pixels of all focal stacks. This is a cumulative correlation score and is computed as:

$$\hat{C}_r(F_i, F_j) = \sum_{FS} \sum_{p} \sum_{l} \sqrt{(F_i(p,l) - F_j(p,l))^2}, \tag{3.5}$$

where FS indicates all focal stacks, p indicates all the pixels in a focal slice and l indicates the slices in the stack.

Figure 3.5 shows the pairwise correlation between all pairs of measures across the focal stack dataset. From left-to-right and top-to-bottom, the measures are arranged in descending order of consensus $\hat{C}$. We seek to cluster these measures and only select one representative from each cluster. We experiment with two clustering strategies to compute the composite focus measure from the consensus and correlation statistics.

In the first strategy, we isolate all pairs of FMs which show a correlation greater than 80%. From each of these FM pairs, the FM with the higher consensus score is retained and the other is removed. This process is applied transitively, i.e. if the correlation between A:B and B:C is greater than 80%, then the measure with the highest consensus score is retained (say A) and the other measures (B and C) are removed. On iteratively parsing through all pairs of highly correlated FMs, we arrive at the list in Figure 3.4, which shows the top ten FMs with high consensus but not high correlation using the transitive pairwise criterion.

In the second strategy, we use the similarity between focus measures encoded in the correlation matrix of Figure 3.5 to cluster the focus measures. We apply hierarchical agglomerative clustering on the distance matrix (reciprocal of the similarity matrix) to cluster correlated focus measures.

We invert the correlation score between all pairs of measures to compute a distance $d_{F_i,F_j} = \hat{C}_r(F_i, F_j)^{-1}$. Agglomerative clustering is then applied with the single-linkage criterion, which specifies the distance D between clusters c_i and c_j as the minimum distance between all the measures belonging to the clusters:

$$D(c_i, c_j) = \min_{F_i \in c_i, F_j \in c_j} d_{F_i,F_j} \tag{3.6}$$

The optimal number of clusters is computed using the gap statistic [129] that uses the cumulative within-cluster distance W_m for all candidate number of clusters m as

$$W_m = \sum_{n=1}^{m} \sum_{F_i,F_j \in c_n} d_{F_i,F_j}, \tag{3.7}$$

which is the sum of distances between measures belonging to the same cluster. The optimal cluster count is set to the point at which the logarithmic plot of the within-cluster distance W_m exhibits an elbow or

begins to flatten out. This is the point at which the clusters have achieved an optimal distribution in terms of the inter-cluster and intra-cluster distances.

We now arrive at an optimal number of 23 clusters for the $N_F = 39$ measures used in our analysis. The dendrogram for this clustering is shown in Figure 3.6. We use the measure with the highest $\hat{C}$ as the representative measure for each cluster. We further sort all the representative measures based on $\hat{C}$. The top ten focus measures that exhibit high consensus but not high correlation using agglomerative clustering, are shown in the list in Figure. 3.4.

A weighted combination of the top five focus measures of Figure 3.4 forms our composite focus measure (cFM). The weight for each measure is its normalized cumulative consensus score. More measures from this list can be used for focus measurement at the cost of more computation but the benefit of adding measures keeps decreasing as the list is traversed, as elaborated in [107]. Using more than five FMs in the cFM results in minor improvements in focus estimation quality at the cost of increased computation, while using lesser FMs results in loss of quality. We also note that the qualitative gain beyond five measures is practically negligible. In our practical implementations of the composite focus measure, we reuse common computations across different measures wherever possible in order to achieve a minimal computation budget.

3.2 Evaluation of the cFM

To verify whether the computed cFM is universal, we re-compute the consensus for all focus measures using Equation 3.1, but the data term for each pixel is computed using only the ten measures from the composite measure of Figure 3.4. The correlation scores between the measures remain unchanged. We observe that there is no re-ordering in the composite measure on re-computing consensus scores, confirming the robustness of the computed composite measure.

To confirm the generalization of the cFM, we evaluate the consensus and correlation of focus measures for different subsets of our focal stack dataset. We isolate four subsets containing 50 focal stacks each. The subsets are selected based on scene attributes such as texture complexity, amount of blur, type of illumination and random selection. Table 3.1 shows the top-five composite measure for the four subsets of (1) scenes with high texture, (2) scenes with high degree of blur, (3) outdoor scenes with bright illumination, (4) random subset of 50 focal stacks. Such a categorization has little impact on the composite measure, the only difference being that the HFN measure does not cluster within the LAP2 family for densely textured scenes. Overall, the top five measures are consistent and similar, suggesting that the composite measure is general.

We further note that our second strategy is a more structured approach to cluster correlated focus measures across a large dataset of focal stacks. Moreover, the overall similarity in the composite measure between the two strategies goes to suggest that the composite focus measure is quite general.

0	Full Dataset	MIS8	LAP1	LAP2	STA3	GRA5
1	Dense Textures	MIS8	LAP1	LAP2	HFN	STA3
2	High Defocus	MIS8	LAP1	LAP2	STA3	GRA5
3	Outdoor Illumination	MIS8	LAP1	LAP2	STA3	GRA5
4	Random subset (50)	MIS8	LAP1	LAP2	STA3	GRA5

Table 3.1 The top-five measures in the cfm show very little change on using different subsets of the focal stack dataset.

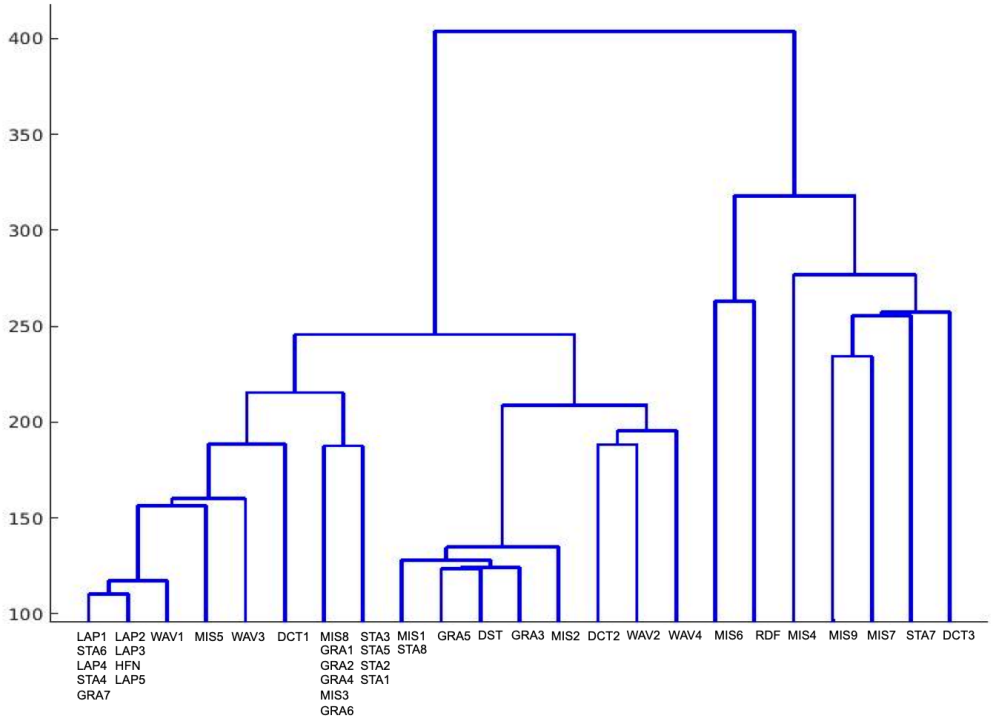

Figure 3.6 23 Optimal clusters selected using the GAP statistic [129] applied with hierarchical agglomerative clustering on the correlation matrix of Figure 3.5.

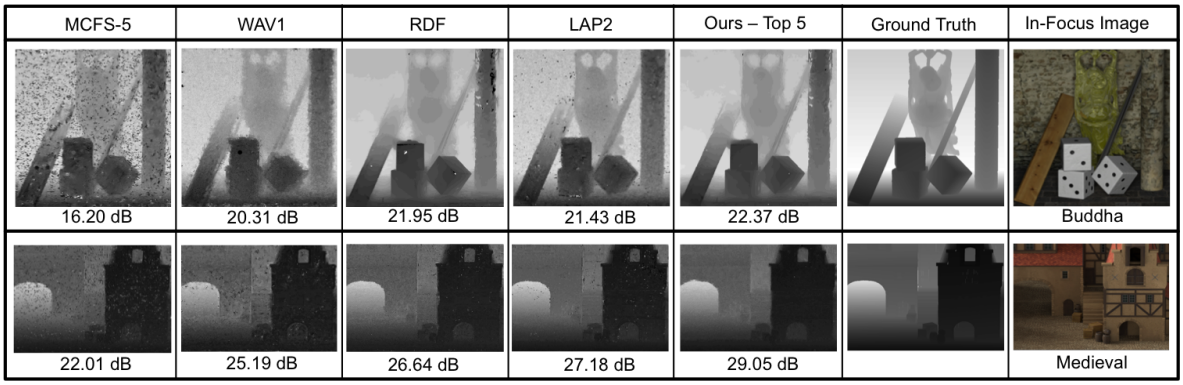

Figure 3.7 Quantitative Evaluation on two synthetic datasets from [135]. We generate 25 focal slices using the ground truth depth map and a standard depth-from-focus pipeline to compute depth using different FMs. Our composite focus measure performs better than the top single measures from [98, 123], which is visible in the images and reflected in the PSNR (in dB) reported below each depth-map. MCFS-5 denotes selecting top five measures using the unsupervised feature selection approach of [16]. We report PSNR to indicate the comparison between 8-bit grayscale ground truth depth maps and high resolution 8-bit depths computed using our method.

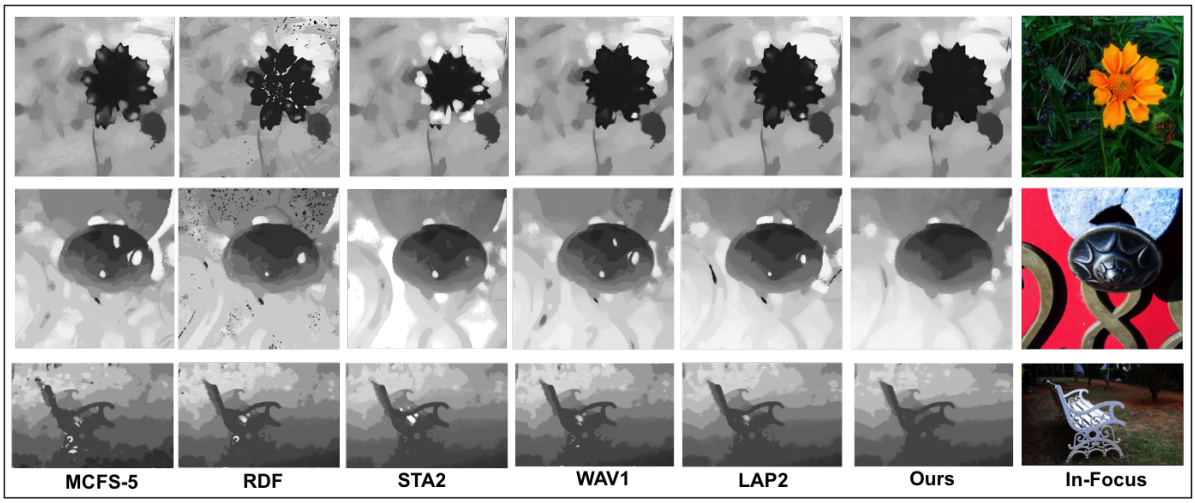

Figure 3.8 Qualitative comparison of the top individual focus measures from [98], our implementation of [123] and our composite focus measure. A standard two-stage depth-from-focus pipeline is used in all cases. The composite focus measure captures the true focus profile even at difficult scene locations. MCFS-5 denotes using the top five focus measures selected using the unsupervised approach of [16].

3.3 Summary

In this chapter, we propose a composite focus measure that combines the strengths of multiple focus measures for accurate focus peak estimation. Our composite focus measure is better suited for focus estimation (and therefore depth estimation) from a set of focal slices of the scene than existing single focus measures. We show qualitative and quantitative performance of depth/focus estimation using the cFM in Figure 3.7. This is computed on a synthetic focal stack created from a 3D scene and its depth map. Additionally we also we show the performance of depth/focus estimation using our composite focus measure on real world focal stacks in Figure 3.8. Figures 3.9 and 3.10 provide more examples of dense depth estimation using our composite focus measure. The approach for computing scene depth is described in detail in Chapter 6. Our composite measure shows superior estimation of depth and smooth focus estimates compared to well-performing single focus measures. In the next chapter, we discuss how this measure can be used to build a compact representation of the focus content in the scene for post-capture focus control.

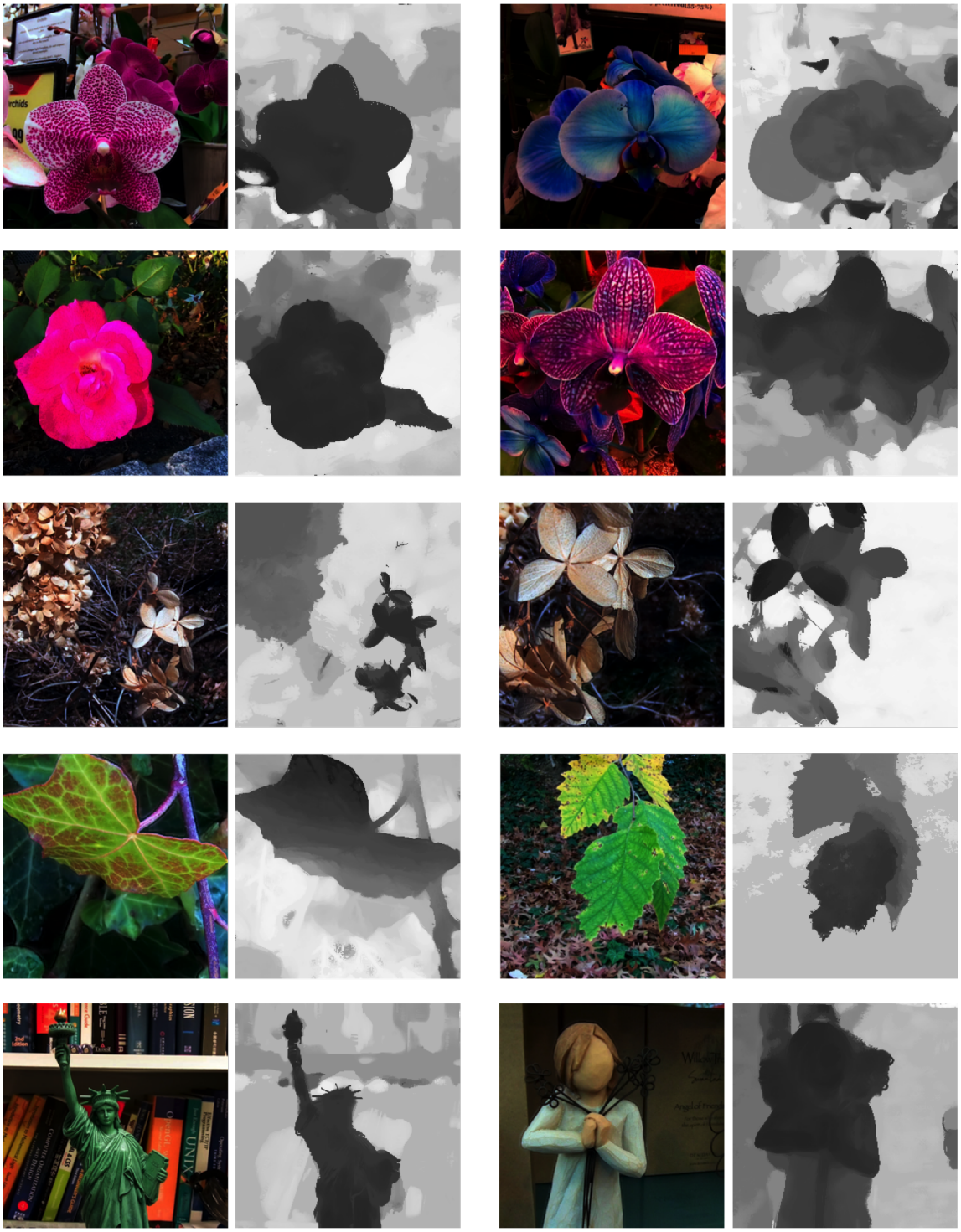

Figure 3.9 Dense depth estimation using our composite focus measure on several focal stacks.

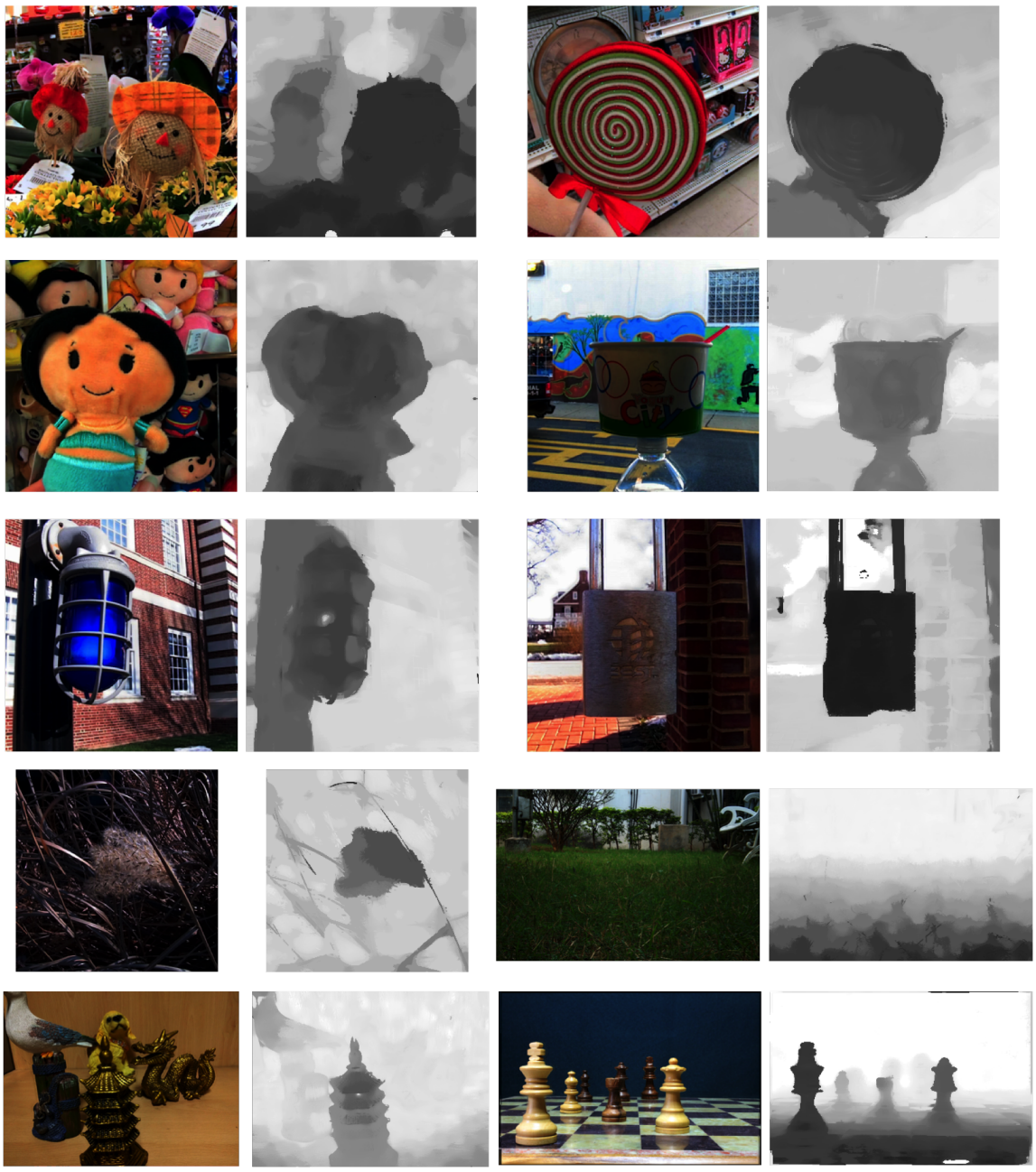

Figure 3.10 Dense depth estimation using our composite focus measure on several focal stacks.

Chapter 4

Focus Representation & Manipulation

Sharp and soft focus are important attributes of a good photograph. Focus and defocus blur are used creatively by photographers to produce remarkable compositional effects. An image captured using a wide-aperture camera is a collection of differently focused scene points. The relative geometry of the sensor and the lens at capture-time governs the points in the scene that appear focused in the image. The light arriving from these points contributes to a single or very few pixels on the sensor. Other regions of the scene appear defocused by an amount proportionate to their distance from the in-focus region. The luminosity of defocused scene points is distributed across a set of proximate pixels. The spread of these pixels is referred to as a defocus kernel. The size and shape of this kernel depends on the distance of the scene point from the in-focus region and its 2-dimensional pixel location in the image. Defocus kernels of proximate pixels may overlap with each other, leading to complex pixel interactions. An accurate model for focus and defocus blur is relevant to computational photography as it can enable measurement of focus for tasks such as de-blurring, depth-of-field extension, refocusing and depth-from-focus. Post-capture modeling of focus and defocus blur from a single image is an ill-constrained problem. Accurate focus modeling usually requires multi-focus imagery and a robust method to estimate in-focus pixels and defocus kernels.

In-focus pixels can be estimated by measuring the sharpness across each pixel's neighborhood. As discussed in the previous chapter, measuring the focus at any pixel is a complex task. In-focus pixels are expected to be sharp while defocused pixels exhibit low contrast. Sharpness alone is an unreliable estimate however as it is dependent on the texture and arrangement of scene points. Furthermore, the intensity of a pixel is quantized by the dynamic range of the sensor and may not represent the original luminosity of its scene point. The intensity might also be mixed with the defocused intensity of other scene points. These issues complicate the task of identifying whether a pixel is in focus and by what amount. Furthermore, the image-processing tools available for estimating sharpness are not robust, as discussed in the previous chapter.

Modeling the defocus kernel at a pixel is also a challenging task. The size and shape of a defocus kernel depends on the camera and scene geometry. The kernel shape is also affected by vignetting (or kernel-shortening) close to the boundaries of the aperture. The defocus spread from a farther scene point may be partially occluded by the presence of closer objects. Such interaction between defocus kernels

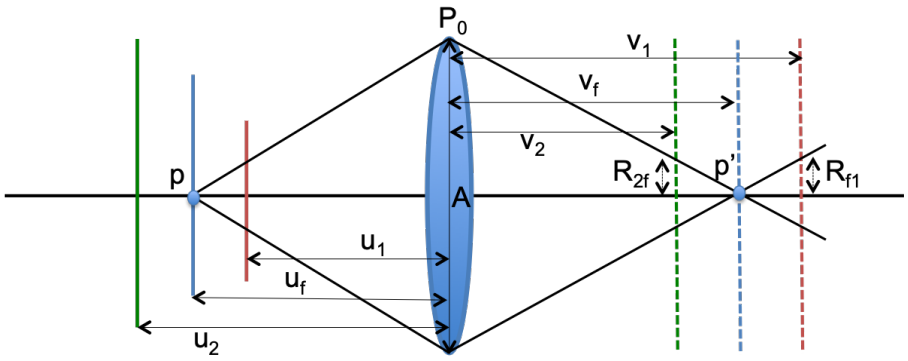

Figure 4.1 The rays from scene point p at a depth u_f converge at an in-focus pixel p' when the sensor is positioned at v_f. At other sensor positions such as v_1 and v_2, the rays from p spread across a defocus kernel of radius R_{f1} and R_{2f} respectively.

from different depths requires accurate geometric modeling. In this chapter, we propose a robust method to manipulate focus and defocus blur:

1. We build a compact representation for multi-focus input imagery using the composite measure and a calibration method that estimates the size and shape of defocus kernels.

2. We propose a novel algorithm for geometric depth-of-field manipulation. Our algorithm correctly accounts for complex pixel interactions at depth edges using an occlusion coefficient.

This chapter provides a comprehensive study of focus and defocus and is applicable to a variety of scenes captured as a focal stack. Careful analysis of focus and defocus blur along with simple models that help in synthesis are the primary contributions in this chapter. We expect our model to be used by image editing tools in conjunction with focal stacks or RGBD images to enable post-capture manipulation of focus.

4.1 Wide-Aperture Imaging

An image captured through a finite aperture opening consists of a combination of focused and defocused scene points. Unlike a pinhole camera that captures one (or very few) rays at each pixel, a wide aperture camera records the combination of several rays at a pixel. If the rays at a pixel arrive from the same scene point, the point manifests as an in-focus pixel in the image, while it appears as a defocused pixel if multiple scene points contribute to it as illustrated in Figure 4.1.

The characteristics of a wide-aperture image such as field-of-view, depth-of-field, focus distance, amount of defocus at each pixel, image brightness and sensitivity depend on the nature of the lens and the capture-time camera parameters. Image formation in wide-aperture cameras is typically modeled as a two stage process. The first stage deals with the optical traversal of light through the lens assembly

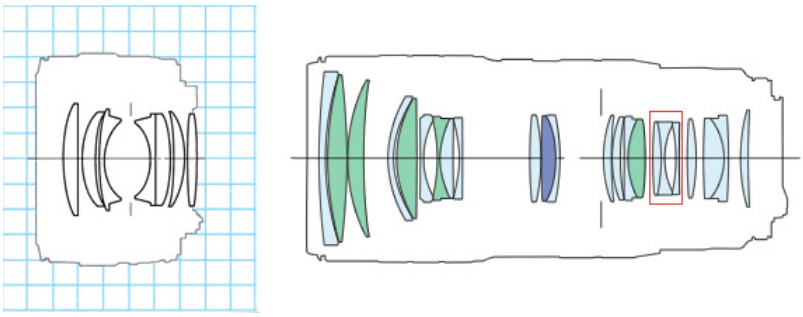

Figure 4.2 A schematic illustration of a simple lens (left) and complex zoom lens (right). A typical lens unit is made up of several lens-elements. Image credits: [62]

and its collection onto the discrete photo-electric sensor. The second stage models the conversion of sensor voltages to discrete pixel intensities in the image [15].

4.1.1 Light Traversal - Lens Optics

A wide-aperture lens is made up of a variety of lens elements arranged in conjunction with each other within the body of the lens [62]. Most lenses available today are a result of several incremental innovations in photographic-lens design ranging as far back as the early 18th century [62]. The nature of real lenses is complex as is illustrated in Figure 4.2. The figure shows a compound lens consisting of several lens elements arranged along a common principal axis. Each element is a part of a larger element group and each group is designed to correct optical aberrations. These aberrations are either geometric or chromatic in nature and manifest as artifacts in the captured image. Geometric aberrations are a result of the structure of the lens elements. Coma, astigmatism, spherical aberration, field curvature, shape distortions, bokeh etc. are the different geometric aberrations. Chromatic aberrations are caused due to the difference in reflection/refraction coefficients of the lens elements based on the wavelength of incident light and commonly occur as axial or lateral aberrations. Zoom-lenses are capable of achieving telephoto as well as wide-angle focal lengths within the same lens unit, by moving the parts of the lens relative to each other. Housing the entire setup within a relatively small lens body usually requires highly complex element groups. Other corrections such as minimization of lens flare and variable coating over different lens elements add an additional dimension to the complexity of lens design. Digital pixel sensors also complicate lens design as most pixels perform best when light falls on them in a square manner rather than at slanted angles.

The current state of incredibly complex lens design can be attributed to the requirement for aberration-free images, wide range of field-of-view, wide range of apertures and maximum and uninterrupted light throughput across the lens body. Computational control over image parameters after it has been captured would benefit from a ray-specific traversal map for each ray passing through the lens. However, it is very difficult to geometrically trace image formation through a real lens, mainly due to lens com-

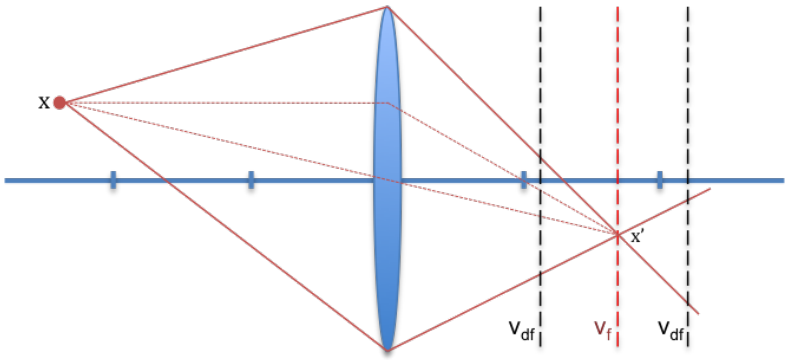

Figure 4.3 In-focus and defocused sensor positions for a source point x

plexity and even more so because exact lens specifications and design are proprietary information that are rarely shared by manufacturers. Moreover, even if such information is available, tracing a real lens would involve wavelength-specific and non-ideal simulation of optical processes. The process of image formation is thereby abstracted into one of two fairly basic lens approximations:

4.1.1.1 Thin-Lens Approximation

The thin-lens approximation for an aperture camera assumes that the overall contribution of the lens is equivalent to an ideal, aberration-free double convex lens. Several rays from a source point x in the scene traverse through the bi-convex lens and converge at a fixed distance behind the lens to form its image x', as described by the thin-lens equation 4.1.

$$\frac{1}{F} = \frac{1}{v} + \frac{1}{u} \tag{4.1}$$

If the lens-to-sensor distance coincides with the focus distance v_f for the source point x, then it appears to be in focus in the image. This set of converging rays spreads out to form a finite circle of confusion (CoC) for all other lens-to-sensor distances as shown in Figure 4.3. This results in the source point spreading its intensity across a defocused disk [80]. The distribution of intensity across this disk on the image plane is typically modeled in the image-space as a convolution of the focused intensity or radiance of the source point with an appropriate Gaussian point-spread-function [96].

4.1.1.2 Thick-Lens Approximation

The structure of a real lens is far from the commonly assumed abstract model of a thin-lens. A more accurate but yet simplistic geometric model for lenses is the thick-lens or the double-gauss approximation. This model assumes that the image distance v and object distance u are measured from two hypothetical principal planes of the lens assembly [52, 128]. This is illustrated in Figure 4.4, where

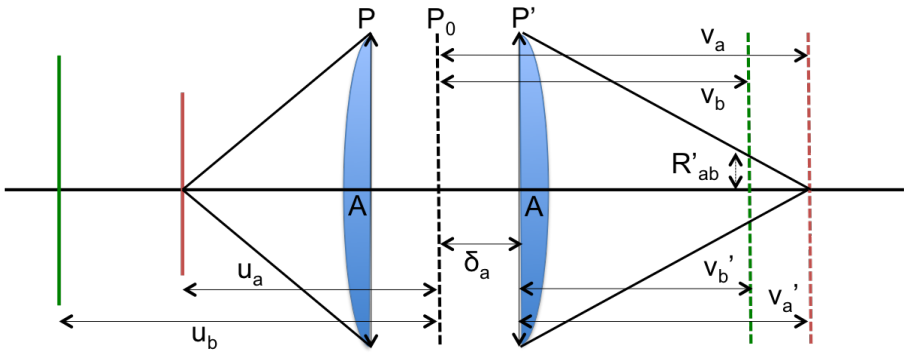

Figure 4.4 Radius of kernel for defocused source point in an aperture camera lens

the distances to the image and the object are measured from the principal planes. v'_a, v'_b refer to the distances between the image planes and the sensor-side principal plane P', while v_a, v_b refer to corresponding distances to the assumed single principal plane P_0 of Figure 4.1.

The double-gauss lens model can be related to the thin lens model based on the separation δ_a of the principal plane P' with the single principal plane P_0 of Figure 4.1. This separation δ_a is constant for all other points imaged using this configuration. For scene points focused at other distances b, the separation changes as the location of the hypothetical principal planes changes with focus distance, even if the focal length remains constant [128]. Thus the separation δ_a between the sensor side principal plane and the assumed ideal principal plane depends on the focus distance a. The lens model for the thick lens can be described as:

$$\frac{1}{F} = \frac{1}{v_a - \delta_{va}} + \frac{1}{u_a - \delta_{ua}} \tag{4.2}$$

The in-focus contribution and the size and the shape of the defocus kernel can be computed using the distances v'_a, v'_b computed from the sensor-side principal plane.

4.1.2 Intensity Conversion - Sensor Processing

Light rays are collected on a photoelectric sensor after travelling through the lens. The light rays converging to very small kernels contribute to focused pixels and others contribute to defocused pixels. The conversion of light falling on the sensor to a pixel intensity occurs over a series of steps [15] that are briefly discussed below.

1. Analog Front-End: The voltage range collected on the sensor is first mapped to the desired signal range in this first stage. This is known as gain control. Also, precalibrated or shielded pixels are masked in a black-light subtraction step and dead pixels are compensated using a factory-calibrated look-up table. Vignetting and sensor cross-talk correction is also applied at this stage in some cameras.

2. White Balancing: White balancing is a technique intended to mimic chromatic adaptation of the human eye. This step usually involves the application of a transformation matrix specific to the camera-settings onto the input intensity matrix from the previous stage.

3. Color-Filter-Array Demosaicing: This step interpolates the color-filtered light falling on the individial photoreceptors into RGB triplets. Most cameras use proprietary and camera-specific demosaicing algorithms.

4. Noise Reduction: Demosaicing is usually followed by a noise reduction step where despeckle filters are used along with high-pass filtering for image sharpening.

5. Color-Space Manipulation: The computed colors are now mapped from the sensor-specific demosaiced RGB triplets to a standard color space such as sRGB using appropriate transformations.

6. Tone-mapping: The pixel intensities are so far linear based on the light collected at each pixel. These are now transformed using a non-linear tone-mapping step to convert them into particular tonal reproduction. Gamma correction to mimic more sensitive human perception at low intensities is also applied at this stage.

7. Compression and Output Image: The image intensities are finally compressed based on the desired output format and written to a file along with any metadata.

The transformations described above convert the luminosity of light rays falling onto the sensor into pixel intensities. The scaling of colors and intensities is a non-linear process and differs for different cameras and camera parameters. When the true luminosity of scene points is too high or too low for the dynamic range of the sensor, the captured pixel color is either over-saturated or not captured at all respectively. Over-saturated intensities lead to bokeh artifacts in wide-aperture images especially when they contribute to large kernels. In order to manipulate an image post-capture, it is it is important to consider and account for all the transformations from scene point luminosity to pixel intensities that occur in the operations described above.

Modeling the image formation pipeline in wide-aperture cameras is important for accurate estimation and processing of focus and defocus blur. In this chapter, we present a method to identify crucial capture parameters such as in-focus pixels, dual focus pixels, defocus radii and intensity saturation from multiple wide-aperture images captured as an ordered focal stack.

The formal definition of a focal stack $\mathcal{G}$ is a sequence of k images (called focal slices) $\mathcal{G}_i, 1 \leq i \leq k$. Each slice is captured with a progressively varying focal distance but the same aperture opening. A focal slice $\mathcal{G}_i$ is the wide-aperture image corresponding to the focus position i and can be defined as:

$$\mathcal{G}_i = \int \int h^i(x, y, d_{(x,y)}) \hat{\mathcal{G}}_r(x, y)\, dx\, dy\,, \tag{4.3}$$

where h^i is a spatially varying defocus kernel whose size and shape depends on the spatial location of the pixel and the depth $d_{(x,y)}$ of its corresponding scene point and $\hat{\mathcal{G}}_r$ is the radiance or luminosity of the scene point. An ideal focal stack captures each scene point in sharp focus in one and only one focal slice. Focal slices exhibit magnification differences as the focus position varies. This can be corrected using image registration or image alignment. Once registered, a focal stack is a volume of pixels, with each node in the volume representing the pixel's intensity in the corresponding slice. The variation of pixel intensities across this volume can be used to estimate in-focus scene points and defocus kernels as demonstrated in the forthcoming sections.

4.2 Focus Representation

The composite focus measure that we proposed in the previous chapter can be used to build a compact model for focal stacks. We propose a representation for focal stacks that consists of the in-focus intensity for each pixel, a secondary dual focus intensity wherever applicable, bokeh scaling of the intensity value wherever applicable, and the defocus kernel at the pixel at all slices of the captured focal stack. This representation of focus is compact compared to the full focal stack and encodes all the high-level as well as the fine characteristics of visible scene points.

4.2.1 In-focus Pixels

The composite focus measure applied at each pixel of a focal stack provides an estimate of the amount of focus of the pixel in each slice. In general, the response of a focus measure across a focal volume is most reliable and informative at high gradients. The reliable in-focus location at high-gradient pixels can be propagated to other pixels based on image content. We follow a similar principle to identify the in-focus position for all pixels.

We use cost-volume filtering [103] to generate a smooth focus map $\mathcal{I}$ which encodes the slice at which each pixel exhibits a focus peak. A cost volume, representing the cost of labeling a pixel to a focus location, is created with k nodes for each pixel, where k denotes the number of slices in stack. The cost of labeling a pixel p to a location l is computed as:

$$C_l^v(p) = cFM(p,l)^{-1}, \tag{4.4}$$

where, $cFM(p,L)$ is the response of the cFM evaluated at pixel p in slice l. The cost at each node is inversely proportional to the response of the cFM. Edge-aware filtering [40] of the cost volume based on a guidance image can be used to propagate confident in-focus labels to all pixels. We generate a guidance image by choosing the focal slice for which the cFM is maximized at each pixel. This is a neighborhood-agnostic image and provides a coarse but useful approximate of depth-edges. The final focus map $\mathcal{I}$ can be computed as the location of minimum cost for each pixel p in the filtered cost volume $C^{v'}$.

$$\mathcal{I}(p) = \arg\min_{i=1}^{k} C_i^{v'}(p) \tag{4.5}$$

Figure 4.5 Left-to-right: In-focus image $F_{\mathcal{I}}$ and focus map $\mathcal{I}$ for focal stacks captured using Canon EOS 70D, Canon EOS 1100D, Lytro Illum and the Apple iPhone 6.

Figure 4.5 illustrates a few examples of focus maps computed using the composite focus measure and cost volume filtering.

4.2.2 Dual-focus Pixels

The focus map $\mathcal{I}$ represents the location of best focus for each pixel. This may not be the only focus distance at which the pixel is in-focus. In some situations, focusing on the background leads to foreground objects being defocused by a large extent such that the background objects become visible through them. The pixels at the edges of the foreground object will now be in-focus at two focal slices, both of which must be considered for accurate geometric modeling of the scene. In principle, it is possible that a single pixel has more than two in-focus slice candidates. However, this situation is impractical given the exponential increase in size of the depth-of-field with increasing depth.

We identify pixels with dual-focus locations using the composite focus measure. This is a crucial attribute of wide-aperture images which is not considered by previous methods dealing with focal stacks. The result of cost volume filtering is a cost vector which can be reciprocated to build a focus vector for each pixel in the scene. The maxima of this vector indicates the location of best focus and is already encoded in $\mathcal{I}$. We estimate secondary peaks by applying non-maximum suppression to the focus vector for each pixel across focal slices. We seek pixels that show a secondary peak and lie close to a strong gradient to build a dual-focus map $\mathcal{I}_d$.

$$\mathcal{I}_d(p) = \left\{ \begin{array}{ll} l & \text{if } cFM(p,l) > cFM(p,l+\Delta l) \\ & \Delta l \in [-w,w] \;\&\; l \neq \mathcal{I}(p) \;\&\; \nabla F_{\mathcal{I}} > t_{\nabla} \\ 0 & \text{otherwise} \end{array} \right\}, \tag{4.6}$$

where the w parameter encodes a neighborhood of 10% of focal slices and t_{∇} is a threshold on image gradient. The pixels belonging to the dual-focus map $\mathcal{I}_d$ are considered separately for geometric refocusing as described in section 4.3.

4.2.3 Scaling pixel intensities

Natural scenes may consist of bright scene points corresponding to light-sources or specularities. Camera sensors typically have a limited dynamic range and very bright scene points usually manifest as uniform saturated intensities. Such pixels do not diminish in intensity even on blurring as their distributed intensity is still higher than the dynamic range of the sensor. This phenomenon is responsible for *bokeh* circles in wide-aperture images shown in Figure 4.9.

Pixel that contribute to bokeh circles need to be identified to simulate accurate geometric refocusing of the scene. Drawing inspiration from existing methods such as [35, 123, 124], we identify bokeh-causing pixels as those which have an intensity larger than a threshold $t_{\mathcal{B}}$ and do not change their intensity across focal slices:

$$\mathcal{B}(p) = \left\{ \begin{array}{ll} 1 & \text{if } \mathcal{G}^i(p) > t_{\mathcal{B}} \ \ \forall i \in [1, k] \\ 0 & \text{otherwise} \end{array} \right\}. \tag{4.7}$$

The focal slice at which the bokeh circle surrounding these points is the smallest is identified as the in-focus slice for the bokeh scene point. All pixels that exhibit bokeh are labeled using a bokeh mask $\mathcal{B}$. The true intensity of such pixels is identified by scaling up and fitting the appropriate pillbox function that results in the largest bokeh circle around the pixel in the focal stack. These pixels are considered separately for geometric refocusing described in section 4.3.

The in-focus map $\mathcal{I}$, the dual-focus map $\mathcal{I}_d$ and the bokeh map $\mathcal{B}$ form the in-focus representation for scene points. Additionally, we compute $F_{\mathcal{I}}$ which is a collection of in-focus pixels chosen from their in-focus slice $\mathcal{I}$. We also compute $F_{\mathcal{I}_d}$ which is a sparse collection of secondary in-focus intensities for pixels that show dual-focus positions. These maps and images efficiently represent the in-focus content within a focal stack.

4.2.4 Defocus Kernels

To estimate the defocus kernels of different pixels, we use the in-focus intensity $F_{\mathcal{I}}(p)$ of a pixel p as a representation of the luminosity of its scene point. Using this luminosity, we estimate the degree of defocus or the blur-radius for that pixel at other focus positions. Defocused intensities of the points from different depths may contribute to the same pixel on the sensor. We therefore do not use pixels that lie close to depth edges and restrict kernel estimation to equi-focal pixels.

We isolate the pixels that belong to similar regions in depth based on $\mathcal{I}$. These pixels come from scene points that were close to the same focal plane in the world and are called equi-focal pixels. They appear equally focused (or equally defocused) in any focal slice. The defocus radii of equi-focal pixels can be calibrated in a spatially-invariant manner. The defocus radii of pixels close to a depth-edge can consequently be computed from their nearest equi-focal region.

We use a generative approach to model the defocus kernel that an in-focus pixel subtends at other focal slices. For a defocused scene point, the rays from the point spread out across a defocus kernel of an appropriate size and shape (Figure 4.1). This can be modeled as an intensity distribution operation

from the source pixel to several pixels. The intensity collected at a pixel p from its neighbors q can be modeled as:

$$\Delta I_q(p) = h(p,q) \circledast F_{\mathcal{I}}(q) \quad \forall q \in N_p, \tag{4.8}$$

where $\Delta I_q(p)$ is the intensity that pixel q distributes to pixel p, $F_{\mathcal{I}}(q)$ the intensity of q, N_p the neighborhood around p, $h(x,y)$ the unknown defocus kernel or point-spread-function (PSF) that we would like to model and $\circledast$ denotes convolution. h is typically modeled as a normalized 2D Gaussian kernel with uniform sigma in both axes.

If the above model is restricted to equi-focal pixels, the defocus kernel can be estimated by finding the best spatially invariant convolution kernel that converts a source pixel's intensity into the target pixel as

$$I(p) = \sum_{q \in N_p} \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{(\frac{-d^2}{2\sigma^2})} F_{\mathcal{I}}(q), \tag{4.9}$$

where, σ denotes the space-invariant defocus radius, d the Euclidean distance between pixels p and q and $g(q)$ the focused intensity for each pixel in N_p. Here $I(p)$ is the rendered intensity of a defocused pixel p and can be compared to different focal slices in $\mathcal{G}$ to estimate corresponding defocus kernels.

We use geometric calibration to compute the defocus kernels by iterating across equi-focal pixels in a piece-wise manner. We estimate the size and the variable shape of defocus kernels similar to approaches such as [49, 116, 35]. The σ of the blur kernel depends on the distance between the current focus position and the pixel's in-focus position and the shape of the kernel depends on the location of the pixel on the sensor. Pixels towards the edges exhibit vignetting (clipping) and thus result in shortened blur kernels. Figure 4.6 shows a defocused image of multiple defocused small point light sources captured under low-light conditions. The defocus shapes at the extremeties are visibly shortened compared to those towards the center.

Algorithm 1 Size and Shape of Defocus Kernels

procedure $\Pi(x, y, \mathcal{I}(x,y), T)$
  $s_{xy} \leftarrow |\mathcal{I}(x,y) - T|$
  min $\leftarrow$ inf
  $p_{mn} \leftarrow F_{\mathcal{I}}(x, y, [m,n])$
  $r_{mn} \leftarrow \mathcal{G}_T(x, y, [m,n])$
  while z in sizes(s_{xy}) **do**
    while h in shapes(x,y) **do**
      $\hat{p}_{mn} \leftarrow gaussianBlur(p_{mn}, z, h)$
      $d \leftarrow \sum\sum |\hat{p}_{mn} - r_{mn}|$
      if $d < min$ **then**
        $\Pi(\mathcal{I}(x,y), T) \leftarrow (z, h)$
        min $\leftarrow d$
  return

We estimate blur kernels between all pairs of focal slices using a blur-and-compare framework. For each pair, we use the in-focus pixels $F_{\mathcal{I}}$ and compute their largest continuous rectangular region as

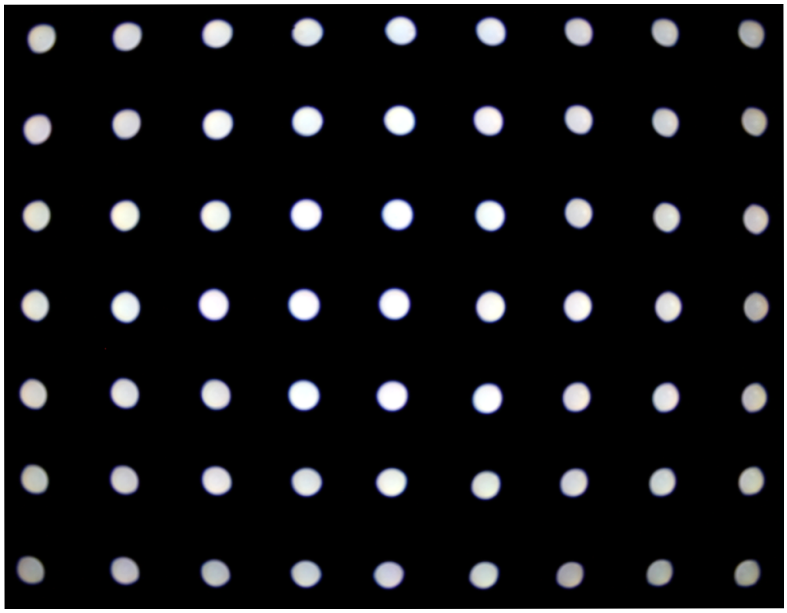

Figure 4.6 The change in shape of defocus kernels due to vignetting on a Canon 1100D DSLR at f/3.5 with a focal length of 18mm.

a reference sub-image. The centre of the rectangular region and its separation from the center of the image is noted to account for vignetting. We then blur the in-focus pixels from one slice with several candidate size and shape parameters and compare the blurred images with the other slice to compute the best match. The shapes are chosen from a set of shapes motivated by [35] and shown in Figure 4.6. The defocus kernel between a pair of focal slices is computed in a bidirectional manner and the size and shape parameters are recorded for both directions. Algorithm 1 outlines the process of estimating the blur radii between a pair of focal slices.

In Algorithm 1, p_{mn} is a patch of size $[m, n]$ close to pixel (x, y) in $F_{\mathcal{I}}$ such that all the pixels in the patch have the same value of $\mathcal{I}$ and r_{mn} is the corresponding patch from the target focal slice $\mathcal{G}_T$. The collection of tuples Π encodes the size and shape of defocus kernels between all pairs of focal slices in a focal stack. Since this is explicitly calibrated for each stack, minor differences in camera design are automatically encoded. Π is a collection of at-most k^2 sizes and k^2 shapes and thus represents the defocus content within the focal stack in a compact manner.

We can now represent a focal stack using the $\mathcal{I}, F, \mathcal{B}$ and Π constructs. This is a compact and robust representation that encodes all the focus and defocus properties of the scene points. The $\mathcal{I}$ maps encode the indices at which the pixel is likely to be in-focus while the F images store photometrically accurate intensities corresponding to these indices, with $\mathcal{B}$ labeling the pixels that require intensity scaling. The Π map encodes the size and shape of defocus that the pixel exhibits in all other not-in-focus slices. The overall storage complexity of our representation is two dense images $\mathcal{I}$ (single channel) and $F_{\mathcal{I}}$ (RGB), three sparse images $\mathcal{I}_d$ (single channel), $F_{\mathcal{I}_d}$ (RGB) and $\mathcal{B}$ (single channel) and at most k^2 defocus sizes and shapes. A full focal stack on the other hand stores k RGB images

with no implicit information about scene content. As an example, for a five slice focal stack (with each focal slice consisting of 1M pixels), the proposed representation provides a compression of 66%, whereas the proposed representation provides a compression of 93% for a similar 25-slice focal stack. Our representation is ideally suited for geometrically accurate and photo-realistic scene refocusing, described in the next section.

4.3 Geometric Scene Refocusing

We describe a geometric approach to scene refocusing using our representation for focal stacks. Our approach pays explicit attention to fine effects such foreground-background occlusions, mixing of blur kernels at depth-edges, vignetting and bokeh circles. Refocusing is formulated as the task of rendering a target depth-of-field, where the limits of the depth-of-field are defined by focal slice indices T chosen from the unique labels in the focus map $\mathcal{I}$.

Generating a novel focused image from our representation is a two-step process. The first step is to identify the appropriate intensity for each pixel and the second step is to distribute this intensity to its neighbors based on the defocus kernel at the pixel. We synthesize a refocused image in a piece-wise manner. For all unique labels in the focus map $\mathcal{I}$, the intensity of the corresponding pixels (or dual-pixels) is considered as the radiance of their scene points. This radiance is scaled appropriately for pixels that also belong to the Bokeh mask $\mathcal{B}$. The radiance of these pixels is then distributed to its neighbors based on the defocus kernel that pixel subtends on the target focus position. This follows the distributive energy model for wide-aperture images suggested in Equation 4.8. The pixels in a scene may correspond to different depths and the spatially varying defocus kernels need to be applied in the right order according to the distribution model:

$$\Delta I_q(p) = \sum_{q \in N_p} H(p, q, \sigma_q).F_{\mathcal{I}}(q), \tag{4.10}$$

where $H(p, q, \sigma_q) = \frac{1}{\sigma_q \sqrt{2R}} \mathrm{e}^{(\frac{-d^2}{2\sigma_q^2})}$ is a Gaussian point spread function with an appropriate σ_q and corresponding shape.

The order of energy distribution becomes important close to depth-edges. As shown in Figure 4.7, when the sensor is placed at a position v_1 which is a near-focus position, the defocused contribution of the background pixels is partially occluded by the presence of the foreground object. Note that our representation consists of dual-focus pixels and captures these pixels behind visible foreground segments. The energy distributed from a background pixel should not freely merge with foreground pixels irrespective of camera and scene geometry. To model the situation shown in Figure 4.7, we evaluate the impact of partial occlusions in a detailed manner.

The amount of partial occlusion of the defocus kernel depends on camera and scene geometry while the shape of the occlusion depends on the shape of the depth-edge. The size of the occlusion varies from 0 to 100% from a limiting point above the principal axis to a symmetric point below the principal axis in

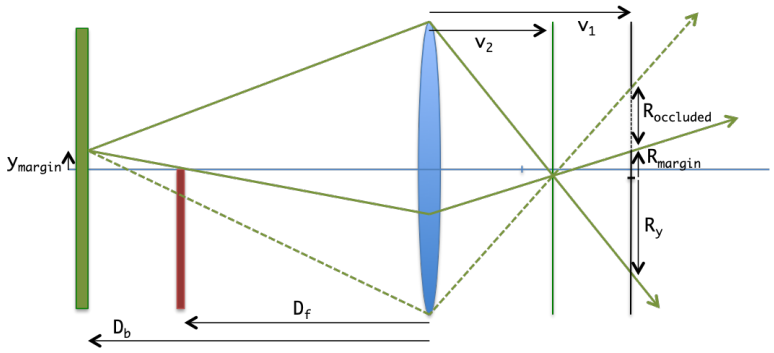

Figure 4.7 The presence of the foreground partially occludes the background kernel by an amount $R_{occluded}$. The occlusion free component R_{margin} can be computed using D_f, D_b, R_y, A and y_{margin} as shown in Eqn. 4.12

Figure 4.7. Figure 4.8 illustrates three regions of focus that are of importance, one focused beyond the background a, the second focused between the foreground and the background b and the third focused closer than the foreground c. The occluded portion of defocus kernels is denoted by o_a, o_b and o_c.

- At a sensor position close to a, the foreground pixels are severly defocused and dominate the image content. A small occlusion o_a in the defocus kernel of the background has minimal impact on image content.

- At sensor positions close to b, the spread of the background pixels overlaps with slightly defocused foreground, however, no background pixel overlaps with in-focus side of foreground pixels (above the principle axis in the diagram). Thus the contribution of a defocused background pixel to a foreground pixel must be disallowed based on the size and shape of o_b.

- When the sensor plane is focused in front of the foreground such as in position c, the background pixel spread overlaps above the principal axis and needs to be shortened based on o_c.

To model partial occlusions in our refocusing algorithm, we introduce an occlusion coefficient β which geometrically restricts the contribution of a source pixel to a target pixel. The energy distribution model for a pixel is formulated as

$$\Delta I_q(p) = \sum_{q \in N_p} \beta_{qp} H(p, q, \sigma_q).F_{\mathcal{I}}(q). \tag{4.11}$$

For background pixels below the occlusion boundary, $F_{\mathcal{I}_d}(q)$ may be used wherever applicable. The occlusion co-efficient is only considered when background-to-foreground mixing occurs, i.e. when $\mathcal{I}_q - \mathcal{I}_p > t_\beta$, with t_β indicating a threshold on number of focal slices. When the precise geometry of the lens and scene is known, the β_{qp} parameter can be computed based on the distance between pixels p and q and R_{margin} [11], where,

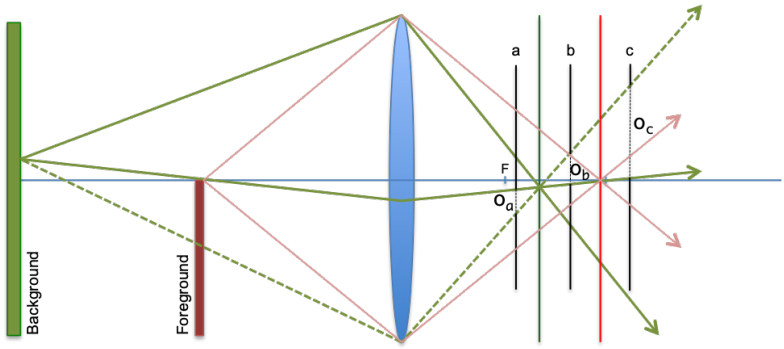

Figure 4.8 The defocused contribution of background pixels is occluded by the foreground. At sensor positions a, b and c, the occlusion due to the foreground on the background pixel's contribution is indicated by the dotted portion o_a, o_b and o_c respectively.

$$R_{margin} = \frac{R_q}{A/2} . \frac{D_f}{D_b - D_f} . y_{margin}. \tag{4.12}$$

D_f and D_b denote the object-side depth for the foreground and background objects, y_{margin} is the distance of source point y from the depth edge on the object side and R_q is the geometric kernel radius for pixel q for the current sensor position. The metric units of R_{margin} and y_{margin} can be converted to pixel units. In a focal stack, absolute depth values for source and target pixels q and p may not be known but their relative depth values are indicated by the focus map $\mathcal{I}$ and the depth ratio can be computed as $\frac{\mathcal{I}(p)}{\mathcal{I}(q)-\mathcal{I}(p)}$. The y_{margin} parameter can be estimated up to an unknown scale factor by calculating the distance of the source pixel from the target pixel in the $\mathcal{I}$ image. This can be converted to metric units using an approximate estimate of closest and farthest focus positions. Using the Euclidean distance d between pixels p and q, the value of y_{margin} can be computed as $y_{margin} = \frac{k\mathcal{I}(q)+c}{f} d$, where k and c are approximate depths of the nearest and farthest focus positions and f is the focal length of the lens. These constants can be estimated exactly if the capture-time sensor positions are known (using APIs such as MagicLantern). For free-form focal stacks, the constants can be approximated based on the visible scene content. Using these constructs, the occlusion coefficient β_{qp} is defined as:

$$\beta_{qp} = \left\{ \begin{array}{ll} 1 & \text{if } d \leq R_{margin} \\ 0 & \text{otherwise} \end{array} \right\}. \tag{4.13}$$

It may be noted that for dual focus pixels, the sign of y_{margin} is negative and the direction of intensity distribution from the dual pixel is automatically inverted.

Algorithm 2 outlines our overall method for rendering novel focused positions from our representation of a focal stack. In a front-to-back manner, all the focus labels in $\mathcal{I}$ are considered. A target pixel is generated by accumulating intensities distributed by its neighboring pixels. Bokeh scaling, dual-pixel intensities, partial occlusions at depth-edges and kernel shortening are considered explicitly before intensity distribution.

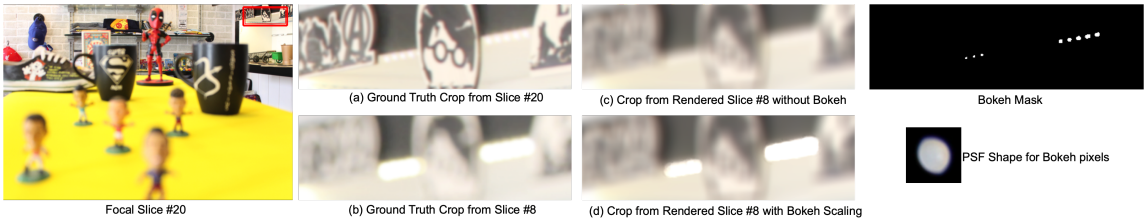

Figure 4.9 Bokeh simulation using saturated intensities and our geometric refocusing algorithm. The bokeh pixels identified in slice #20 are shown in the Bokeh mask. The asymmetric shape of the point-spread-function is selected from Figure 4.6 corresponding to the spatial location of the bokeh pixels. The defocused version of (a) using our algorithm without bokeh scaling is shown in (c) and with bokeh scaling is shown in (d). Note that our algorithm accurately simulates intensity saturation visible in the ground-truth image (b).

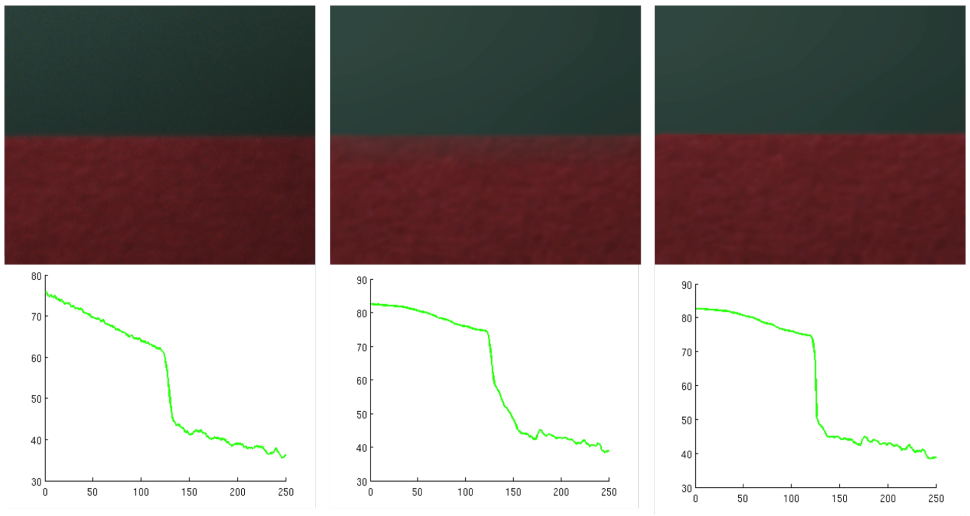

Figure 4.10 A synthetic red-green depth-edge experiment. The target focus position is similar to o_c from Figure 4.7. From left-to-right: Ground truth image, reconstructed image without β and reconstructed image with β. The second row show the corresponding plots of green intensity from top-to-bottom in the images. Our model with β correctly restricts distribution from background to foreground pixels.

Algorithm 2 Refocus Scene to Focus Position(s) T

procedure REFOCUS($\mathcal{I}, \mathcal{I}_d, \Pi, T$)
  for all labels l in $\mathcal{I}$ **do**
    for all pixels q with $\mathcal{I}(q) = l$ or $\mathcal{I}_d(q) = l$ **do**
      $i_p \leftarrow F_{\mathcal{I}}(q)$ or $F_{\mathcal{I}_d}(q)$
      if $\mathcal{B}(q) = 1$ **then**
        $i_q \leftarrow$ scaleBokeh(i_q)
      for all pixels p neighboring q **do**
        kernel $\leftarrow \Pi(\mathcal{I}(q), \mathcal{I}(p))$
        $i_p \leftarrow i_p +$ applyKernel(i_q, β_{qp},kernel);
  return

4.4 Experiments & Results

We perform quantitative and qualitative evaluation of our representation in terms of its capability of reconstructing the focal stack and producing novel refocused renderings of the scene. We also compare our method with that of other state-of-the-art methods in post-capture focus control and manipulation.

4.4.1 Dataset

We use our focal stack dataset of about 320 focal stacks described in section 3.1.1. Each focal stack is rescaled to the order of 1M pixels while preserving the aspect ratio. All experiments are performed on RAW images (or on images rescaled by the appropriate gamma factor). To eliminate pixel misalignment due to magnification, we use the enhanced correlation coefficient maximization approach [24]. The thresholds used in our approach are t_{∇}=20, $t_{\mathcal{B}}$=0.9 and t_{β}=30%.

4.4.2 Reconstructing Focal Slices

The litmus test for our representation of focus is its ability to reconstruct the focal slices in the stack. The model is expected to not only capture the basic blur profile of scene elements but to also re-create some of the fine transitions at depth-edges. We perform a quantitative and qualitative analysis of the reconstruction quality of our model and refocusing algorithm. Figure 4.11 illustrates an example on a focal stack of twenty-slices that is not a part of our focal stack dataset. The figure shows a comparison at three different focus positions - slice 2, slice 9 and slice 18. In the top-row left is the image captured using the camera and in the bottom row is the image refocused using our representation and refocusing algorithm. On a test dataset of ten such focal stacks that were not included in the computation of the composite focus measure, we achieve an average reconstruction PSNR of 41.2 dB per focal slice. The reconstruction quality suggests that our model for focus and refocusing algorithm works well.

4.4.2.1 Ablation Study

To elaborate the impact of the dual-focus representation, the bokeh scaling parameter and the occlusion coefficient, we apply of our refocusing algorithm without the three factors. Figure 4.12 shows an example where our secondary focus peak estimation is necessary for simulating fine focus effects close to depth edges. Figure 4.9 demonstrates the benefit of using our bokeh mask for intensity scaling and the application of defocus kernel shortening to simulate vignetting. Figure 4.13 illustrates the utility of the occlusion co-efficient for accurate focus simulation at deep depth-edges. The qualitative superiority of our approach is clearly visible in all three examples.

To further demonstrate the impact of the occlusion co-efficient at depth edges, we capture a focal stack of a test scene consisting of a red foreground plane and a green background plane at fixed distances from the camera. We reconstruct a target focal slice focused in front of the foreground object (similar to o_c of Figure 4.7). We compare this image with a ground-truth focal slice. We show the effect of using

Figure 4.11 Reconstructed Focal Slices 2, 9 and 18 for a focal stack using our representation and refocusing algorithm. The top-row shows the images captured using a DSLR and the bottom row shows our reconstructed images.

our defocus model without the occlusion coefficient β, in which case the green background incorrectly defocuses into the red foreground as shown in Fig. 4.10.

4.4.3 Refocusing the Scene

Using our representation, we can create novel focus positions for the scene by changing the target focus position T of algorithm 2. The target position T can be a single focus position chosen from one of the labels in $\mathcal{I}$ or a set of labels which may or may not be contiguous.

1. All-in-focus image: In this case, T consists of all labels from $\mathcal{I}$. The refocusing algorithm collects the in-focus pixels from all labels. Some focal stacks with their focus map and corresponding in-focus image are shown in Figures 4.5,4.15.

2. Extended depth-of-field: When T consists of a contiguous subset of labels from $\mathcal{I}$, an extended depth-of-field image can be created. The defocus kernels for such an image are chosen from the limiting labels of T on either side. An extended depth-of-field image is shown in Figure 4.15.

3. Non-photorealistic Focus: When T consists of multiple disjoint subsets of labels from $\mathcal{I}$, a non-photorealistic rendering can be created. The defocus kernels can be chosen from the closest limiting label of T for each subset. A non-photorealistic image is shown in Figure 4.15.

4.4.4 Comparison

We compare our model for focal stacks with other contemporary techniques that deal with defocus modeling and post-capture scene refocusing. Previous methods either do not deal with fine character-

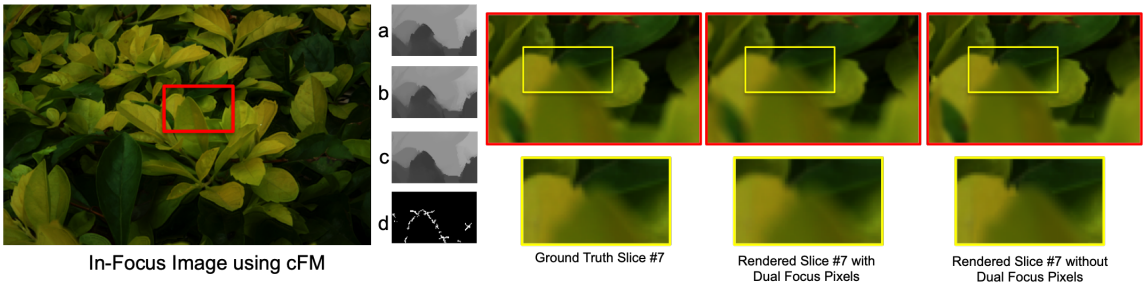

Figure 4.12 The in-focus image shown on the left is the $F_{\mathcal{I}}$ image created using the composite focus measure. Inset (a) shows the focus map $\mathcal{I}$ on using the LAP2 focus measure alone, inset (b) shows the focus map $\mathcal{I}$ on using the WAV1 focus measure alone, while inset (c) shows the focus map $\mathcal{I}$ on using the composite focus measure. Inset (d) is the dual focus pixel map $\mathcal{I}_d$. On using the dual focus pixels, the background dark green leaf is visible through the blurred light green foreground, which is similar to ground truth. Ignoring the dual pixels misses this fine effect as seen in the rightmost image.

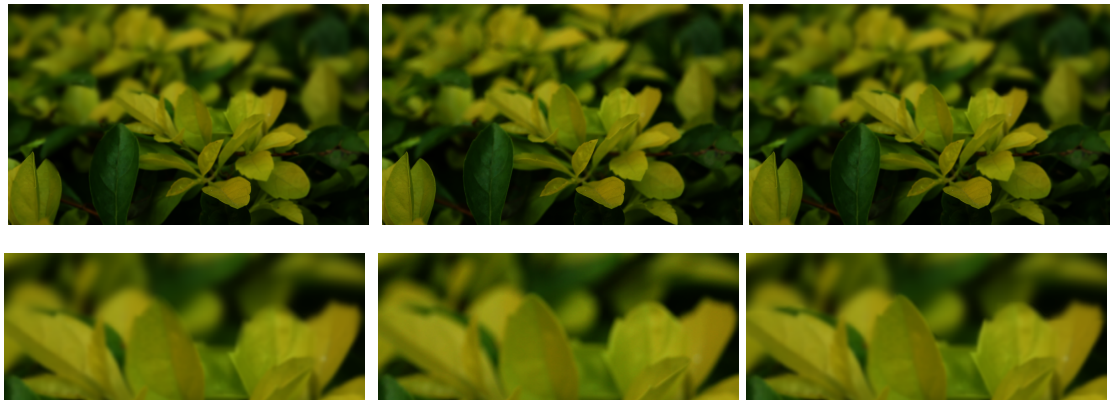

Figure 4.13 First row: Left to right: Simulated focal slice 2 without β, true focal slice 2, simulated focal slice 2 using β. Second row: Insets corresponding to the first row. Notice how the background cuts into the foreground at several leaf edges which is visible in the inset images.

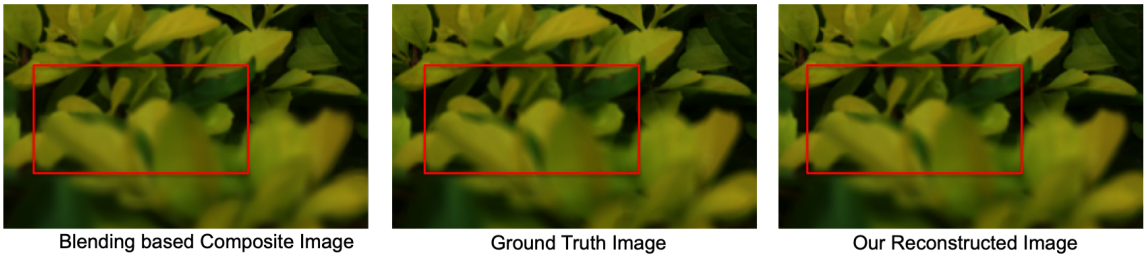

Figure 4.14 A comparison of blending based focal slice reconstruction of [10] with our geometric algorithm for refocusing. Note the patchy nature of depth-edges in the blending based composite image. The dual focus pixels in our representation enable accurate reconstruction.

Figure 4.15 An all-in-focus image (all focal slices in focus), an extended focus image (with multiple contiguous focal slices in focus) and a non-photorealistic image (with non-contiguous focal slices in focus) using our representation and refocusing algorithm.

istics of wide-aperture images or use highly specific camera hardware with precise knowledge of focus positions and/or scene depth. We compare our method with that of [124, 46] for quality of refocusing, [10] for dual-pixel capabilites and [35] for quality of defocus kernels.

4.4.4.1 Comparison of Refocused Rendering

Jacobs et al. [46] synthesize novel depth-of-field images using focal stacks by defining a target sensor defocus map and generating a rendering closest to that map using ray space analysis. Suwajanakorn et al. [124] also show reconstruction of defocused images from focal stacks however very little details about the process of generation of the defocused images is discussed. While our method is structurally similar to that of [46], we differ in two major aspects, definition of the target depth-of-field and dual-focus pixels. We compute fine pixel interactions from the focal slices directly by estimating dual-focus pixels and appropriately control their contribution during refocusing. As a result, the in-focus nature of the background pixels is clearly visible in our method without explicit knowledge of focus position during capture as shown in Figure 4.12.

4.4.4.2 Comparison with Alpha-Blending

Alpha-blending of different focal slices can be applied to generate refocused renderings, as described in Barron et al. [10]. Such methods usually assume an extended canvas of background pixels which are blurred and blended to create a refocused composite. Our dual-focus pixel method clearly identifies the pixel intensity and thus the textural information at partially occluded background pixel. Such awareness of true pixel intensity and color is not possible for mutually exclusive focus textures that are used in blending based methods. In Figure 4.14, we compare our method with the background stretched method of [10]. Background textural details becoming apparent through a blurred foreground is clearly visible in our approach.

4.4.4.3 Comparison of Bokeh Quality

Hach et al. [35] describe the rendering of bokeh highlights as well as the modeling and reconstruction of the PSF for a high-quality camera. They perform tedious calibration for an RGBD camera and learn the distribution of PSFs in order to simulate DoF effects in a post-capture framework. They also describe a kernel stitching algorithm to handle partial occlusions. While their model is designed for a high-end camera with a high dynamic range, we have shown a model which is applicable in a general setting, both in the presence and absence of depth information. We compute the size and shape of defocus kernels relative to the pixel location in the image and we also apply an intensity scaling to bokeh pixels as described in Sections 4.2 and 4.3. This leads to natural bokeh effects in our refocused renderings. While the defocus kernels we estimate may not be perfectly shaped for an arbitrary camera, the shortening of kernels due to vignetting and the intensity scaling method creates photo-realistic bokeh in our images, as shown in Figure 4.9.

4.5 Summary

In this chapter, we have proposed a robust model to represent the focus and defocus properties of a scene from a set of multi-focus images. We presented a geometric algorithm for refocusing from first principles, preserving the fine effects of focus in challenging geometric situations. Our algorithm renders by distributing each pixels intensity to its neighbors using geometrically correct kernel sizes and shapes. Pixels lying close to depth-edges and pixels with saturated intensities are also accounted for correctly. We show the qualitative and quantitative impact of our representation and refocusing algorithm. Such a representation of focus can be ideally suited for image editing toolkits requiring precise manipulation of multi-focus imagery. In the next chapter we propose a learning based approach for post-capture depth-of-field control.

Chapter 5

Learning Based Scene Refocusing

An image captured by a wide-aperture camera has a finite depth-of-field centered around a specific focus position. The location of the focus plane and the size of the depth-of-field depend on the camera settings at the time of capture. Points from different parts of the scene contribute to one or more pixels in the image and the size and shape of their contribution depends on their relative position to the focus plane. Post-capture control of the focus position is a very useful tool for amateur and professional photographers alike. Most portrait photographs are captured with an intent to achieve a shallow depth-of-field around the focused subject. Controlling the size of the depth-of-field or changing the focus position of the scene after it has been captured is an interesting problem in computational photography.

Defocus magnification and post-capture scene refocusing using one or more wide-aperture images has received some attention in computer vision literature. Changing the focus in a scene using a single image is however an ill-constrained problem as the in-focus intensity and the true point-spread-function for each scene point need be jointly estimated before re-blurring a pixel to the target image. Magnification of blur (or portraiture) is however more tractable using a single image as the amount of blur can be first estimated at each pixel followed by blur amplification.

Figure 5.1 Out proposed framework for single-image portraiture. Left: A narrow-aperture portrait image [25] with low defocus in the background. Middle: The composite focus measure [107] computed over the portrait image (contrast enhanced for visualization). Right: A wide-aperture result produced using our method.

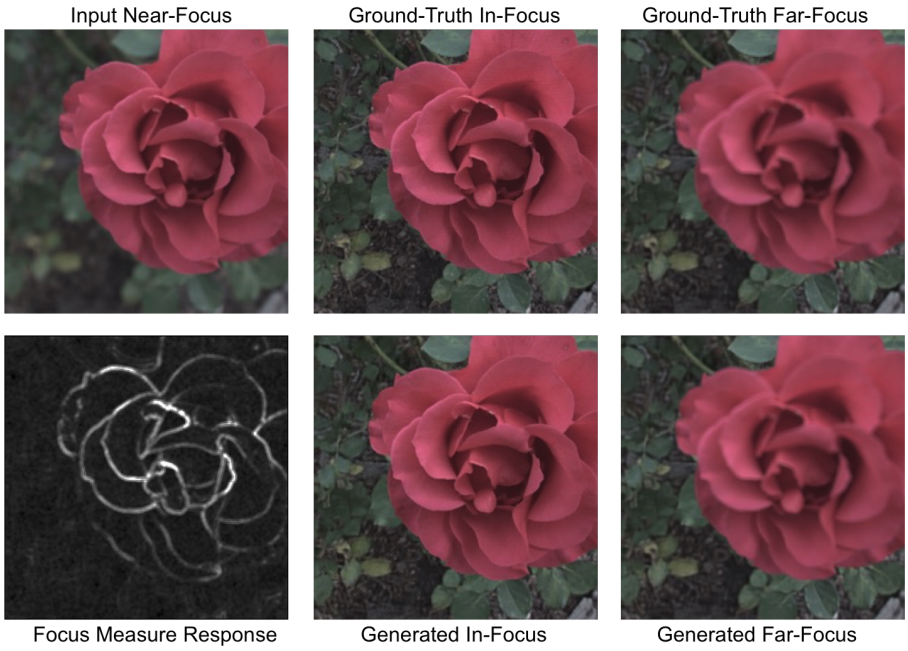

Figure 5.2 Refocusing a single-image: We use an input wide-aperture image along with its focus measure response to create a deblurred, in-focus radiance image. The radiance image is then used together with the input image to create a refocused image. The second and third columns show the quality of our deblurring and refocusing stages.

Multiple focused images of a scene, in the form of a focal stack, contain the information required to not only estimate the in-focus intensity but also the focus variation for each scene point. Focal stacks have been used in the past for tasks such as estimating a sharp in-focus image of the scene [3, 81], computing the depth-map of the scene [79, 124], and free-form scene refocusing [46, 144]. As described in previous chapters, focal stacks result in a loss of temporal resolution due to the finite time required to capture each slice. Focal stacks are therefore limited to static scenes. Most practical applications in portraiture would thereby benefit from a method that can manipulate the focus of a single image.

We introduce DefocusGAN and RefocusGAN which form a comprehensive image refocusing framework for defocus magnification as well as post-capture scene refocusing. Our approach starts with a single input image and enables post-capture control over its focus position and depth-of-field. This is a departure from existing methods in computational photography that provide post-capture control over the depth-of-field using full focal stacks.

Our work is motivated by the impressive performance of deep neural networks for tasks such as image deblurring, image-to-image translation and depth-map computation from a single image. We propose a single-step approach for defocus magnification and a two-stage approach to single image refocusing.

For defocus magnification we present an end-to-end deep neural network that takes an input image with focused and defocused pixels and selectively magnifies the blur at the defocused pixels, thereby

simulating a shallow depth-of-field. We train this blur magnification network using wide aperture images created from a light-field dataset [118] and use a combination of adversarial training, perceptual loss and a novel loss term based on a focus measure to improve generalization. Figure 5.1 is an example of our single-step defocus magnification on a generic portrait image. It may be noted that none of the images in our training data included human subjects. Existing methods [7, 125, 148, 147] estimate the blur at the high-gradient pixels and propagate it to the other locations in the image. Errors in blur estimation manifest as artifacts in the synthesized wide-aperture image, while our learned model is able to create renderings free of such artefacts.

For comprehensive scene refocusing, we use a two-stage approach. The first stage of our approach computes the radiance of the scene points by deblurring the input image. The second stage uses the wide-aperture image together with the computed radiance to produce a refocused image based on a refocus control parameter δ. We train conditional adversarial networks for both stages using a combination of adversarial and content loss [57]. Our networks are additionally conditioned by a focus measure response during deblurring and the computed radiance image during refocusing. We train our networks using wide-aperture images created from the large light-field dataset of scenes consisting of flowers and plants [118]. Figure 5.2 is an example of our comprehensive scene refocusing frameworks using adversarial networks.

5.1 Defocus Magnification

An image captured by an aperture-camera is a composition of light rays arriving from scene points at different distances from the camera. Light rays travel through the body of the lens according to the geometry dictated by the camera settings and finally fall on the sensor. All the scene points at a particular depth in the scene converge at the same distance behind the lens and contribute as focused pixels if the sensor is located at this specific distance. Scene points at other depths either converge in front of or behind this sensor position and thereby contribute to the image as defocused pixels. An image created from pixels at different depths in the scene can be modeled as:

$$\mathcal{I}(x,y) = \int\int \mathcal{H}(x,y,\delta(x,y))\ \mathcal{R}(x,y)\ dxdy, \tag{5.1}$$

where $\mathcal{R}$ is the radiance of the scene point corresponding to the pixel (x,y), $\mathcal{H}$ is the per-pixel defocus kernel, δ represents the size of the kernel and $\mathcal{I}$ is the captured image. The δ parameter represents the separation in depth between the in-focus sensor position of the pixel (x,y) and the current sensor position. Therefore, δ is negligible for in-focus pixels and large for defocused pixels. The defocus kernel $\mathcal{H}$ is typically modeled as a Gaussian with a spread corresponding to δ. In-focus pixels typically correspond to scene points at the same depth in the scene.

A finite region surrounding the in-focus plane, where the defocus blurs are small enough to be imperceptible is known as the depth-of-field of the image. The size of the depth-of-field is inversely proportional to the size of the aperture. On increasing the size of the aperture opening, the deviation in

incoming light rays increases thereby causing a reduction in the depth-of-field. Small/shallow depth-of-field is a useful compositional tool in photography in which the subject is usually kept in focus and other background or foreground regions are defocused. This is widely used in portrait photography, selfies and object photography. Mobile and point-and-shoot cameras are moving towards smaller form-factors and size and thus have narrow apertures. Such cameras are unable to capture small/shallow depths-of-field.

Defocus magnification or aperture magnification is the process of identifying the defocused pixels in the image and selectively blurring them further in a depth-aware manner to simulate a shallow depth-of-field. Defocus magnification simulates an increase in the size of the aperture. Ideally, enlarging the aperture would lead to a reduction in size of the depth-of-field as more pixels from the focused part of the image would now be defocused. However, existing image-processing methods keep the focused pixels intact and blur the defocused pixels further, thus simulating more blur in the background without reducing the number of focused pixels. This simulates the blurring of a shallow depth-of-field without reducing the size of the depth-of-field. Therefore the technique is more correctly referred to as defocus magnification rather than aperture magnification. It can be noted however that if small blurs can be detected accurately even for the pixels that lie within the depth-of-field, a defocus magnification method will simulate geometrically accurate aperture magnification. Generating a wide-aperture image $\mathcal{I}'$ with blur magnification in the blurred regions can be modeled as:

$$\mathcal{I}'(x,y) = \int\int h(x,y,m*k(x,y))\ \mathcal{I}(x,y)\ dxdy. \tag{5.2}$$

Here, $\mathcal{I}$ is the captured narrow-aperture image, h is a re-blurring kernel with a size depending on $k(x,y)$, which is a defocus measure that encodes the amount of blur at a pixel (x,y) in $\mathcal{I}$, and m is a magnification parameter that scales the amount of re-blurring. The in-focus pixels in $\mathcal{I}$ correspond to $k=0$ and will not undergo any re-blurring. The pixels that are already defocused have $k>0$ and will be re-blurred by a kernel of finite size $m*k$.

In the following section we propose an end-to-end deep neural network that magnifies the defocus of an input narrow-aperture image. We train a conditional adversarial network and use a novel loss term. We learn a residual image to convert a small aperture image to a wide-aperture image with a shallow depth-of-field.

5.1.1 Network Architecture

Defocus magnification is a complex image filtering operation defined in Equation 5.2. In-focus pixels are expected to remain in-focus while the pixels that are defocused are to be blurred further using a depth-dependent blur kernel. The challenge in producing a shallow depth-of-field image in a single-step is that the network must learn an implicit representation of focus and re-blur the pixels according to the amount of focus at each pixel. We use a conditional adversarial network for this task, drawing inspiration from [57] and [104] that use adversarial learning for the tasks of image de-blurring and scene refocusing respectively. We broadly discuss adversarial learning and consequently define our network for defocus magnification.

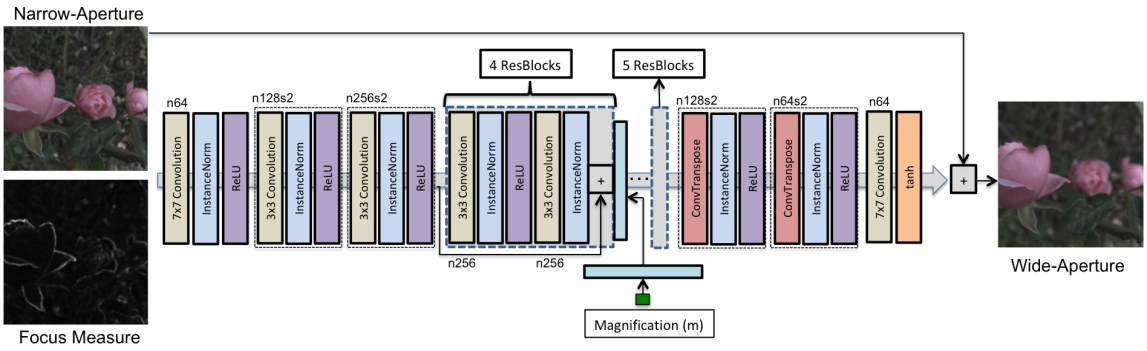

Figure 5.3 Our generator network $\mathcal{G}$ that produces a wide-aperture image from an input narrow-aperture image $\mathcal{I}$ along with its composite-focus-measure channel $f(\mathcal{I})$ and a magnification parameter m. The focus measure channel is appended to the RGB image to create an RBGF input to the network. The magnification parameter is converted to a 64×64 channel and appended after the fourth residual block. The structure of the network is explained in Section 5.1.1.

Generative adversarial networks (GANs) [31] are a class of deep neural networks that define the process of learning as a competition between a Generator network $\mathcal{G}$ and a Discriminator network $\mathcal{D}$. The task of the generator is to create an image parameterized by an arbitrary input which is typically a noise vector. The task of the discriminator is to distinguish between a real image and a generated image. The generator learns to create perceptually real images which can fool the discriminator. The objective function of GANs is defined as $\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}_{GAN}$, where $\mathcal{L}_{GAN}$ refers to the loss function:

$$\mathcal{L}_{GAN} = E_{y \sim p_r(y)} \left[\log \mathcal{D}(y)\right] + E_{z \sim p_z(z)} \left[\log(1 - \mathcal{D}(\mathcal{G}(z)))\right]. \tag{5.3}$$

Here $\mathcal{D}$ is the discriminator, z is the input noise vector to the generator $\mathcal{G}$, y is a real sample, p_r is the real distribution over target samples and p_z is a normal distribution.

Conditional adversarial networks (cGANs) are GANs with additional conditioning provided to the generator to create images of a specific kind. Conditional adversarial networks have been useful for tasks such as image reconstruction and image-to-image translation. Isola et al. [44] provide a comprehensive study of adversarial networks and propose a robust conditional GAN for tasks such as colorization, edge-to-photo synthesis, label-to-photo synthesis etc. Orest et al. [57] build on this and propose a cGAN architecture for the task of image de-blurring. Using a combination of adversarial and perceptual loss, they show high quality reconstruction of in-focus images.

Our conditional GAN for defocus magnification is similar to the cGAN proposed in [57]. The generator is adapted from the style transfer network of Johnson et al. [47] and consists of two strided convolution blocks with a stride of $\frac{1}{2}$ followed by four residual blocks, an additional input using the magnification parameter m, five more residual blocks and two transposed convolution blocks. Each residual block is based on the ResBlock architecture proposed by He et al. [41]. Each block consists of a convolution layer with dropout [119] regularization with a probability of 0.5, followed by instance normalization [130] and ReLU activation [84]. The single-valued magnification parameter m is converted to a 64×64 channel using a fully connected layer and appended to the output at the end of the

Figure 5.4 This figure shows the performance of our defocus magnification network on images from the test split of the light-field dataset. For each pair: Input image on the left, Defocus magnified image on the right. The left column has a magnification of $m = 2$ and the right column has a magnification of $m = 3$.

fourth ResBlock. The input to the generator is the narrow-aperture RGB image with an additional input channel which encodes the amount of focus at each pixel. We use the composite-focus-measure [107] which is our statistical combination of five robust measures of focus described in chapter 3. We compute the response of the composite measure at all pixels of the RGB image and append this as a channel to create an RGBF image, where F represents the focus channel. The focus measure channel improves performance and generalization as we shall show in our experiments. A global skip connection (ResOut) is added to accelerate learning and improve generalization [57]. The output of the generator network is a residual image which can be added to the RGB channels of the input image to generate the wide-aperture result $\mathcal{I}'$:

$$\mathcal{I}' = \mathcal{I} + \mathcal{G}_{\theta_G}\left(\mathcal{I} : f(\mathcal{I}), m\right). \tag{5.4}$$

Here, $\mathcal{I} : f(\mathcal{I})$ represents the RGB image appended with the F channel to create an RGBF image and m is the magnification parameter of Equation 5.2. The architecture of our generator network is shown in Figure 5.14.

The discriminator network used in our experiments is similar to [57] and is based on Wasserstein-GAN [5] with gradient penalty [33]. The discriminator is modeled as a critic network and is similar to PatchGAN [44, 68]. Except for the last layer, all convolution layers are followed by instance normalization [130] and Leaky ReLU [136] with α=0.2. We use a composite loss function for our generator. We use a combination of three different loss functions - adversarial loss from the discriminator, perceptual loss and focus-measure loss. The adversarial loss component is defined as:

$$\mathcal{L}_{cGAN} = \sum_{n=1}^{N} -\mathcal{D}_{\theta_D}(\mathcal{I} + \mathcal{G}_{\theta_G}(\mathcal{I} : f(\mathcal{I}), m)), \tag{5.5}$$

where $\mathcal{I} + \mathcal{G}_{\theta_G}(\mathcal{I} : f(\mathcal{I}), m)$ represents the result image created from the output of the generator.

Perceptual loss, defined in [47], is L2-loss calculated between CNN feature maps of the generated image and the target image. Differences in feature maps encode similarities between two images much better than estimating differences in the image space [141]. The perceptual loss component is computed as:

$$\mathcal{L}_X = \frac{1}{W_{ij}H_{ij}} \sum_x \sum_y (\phi_{ij}\mathcal{I}'^{gt}_{xy} - \phi_{ij}\mathcal{I}'_{xy})^2, \tag{5.6}$$

where ϕ_{ij} represent the feature maps in the VGG19 network trained on ImageNet [21] after the j^{th} convolution and the i^{th} max-pooling layer and W and H are the size of the feature maps.

We also use a loss term based on the response of the composite focus measure [107] over the input image $\mathcal{I}$. We use this term to improve generalization as it encodes the amount of focus at a pixel and is independent of image category. The focus measure loss component is computed as:

$$\mathcal{L}_f = \sum_x \sum_y f(\mathcal{I}_{xy}) \cdot ||\mathcal{I}'_{xy} - \mathcal{I}_{xy}||_1, \tag{5.7}$$

where $f(\mathcal{I})$ is the normalized response of the composite focus measure evaluated over the input image $\mathcal{I}$ and $\mathcal{I}'$ is the result image from the generator. At the in-focus locations in the input image where the value of f is high, the result and the input are expected to be identical and this is enforced by the focus measure loss term. The overall loss for the generator is a combination of the three loss terms:

$$\mathcal{L} = \mathcal{L}_{cGAN} \ + \ \lambda_X \mathcal{L}_X \ + \ \lambda_f \mathcal{L}_f. \tag{5.8}$$

The λ_X parameter is set to 100 as defined in [57] and the λ_f parameter is empirically set to 50.

5.1.2 Training Details

To train our network for defocus magnification we render narrow-aperture and wide-aperture images from a light-field dataset. Srinivasan et al. [118] captured a large dataset of 3343 light-fields of similar scenes consisting of flowers and plants. The images were captured using a Lytro Illum camera [73] and consist of 14×14 lenslet images at a spatial resolution of 376×541 pixels. Typically, only the central 10×10 or smaller grid is useful because the lenslet samples towards the corners suffer from heavy clipping. We simulate narrow-aperture images by applying the light-field rendering equation of [73] to the central 3×3 lenslet grid. To simulate wide-aperture images with more blurring we apply the rendering equation to larger grids of 5×5, 7×7 and 9×9 lenslet images. These images notationally correspond to a magnification parameter of $m = \{2, 3, 4\}$ respectively. The images created with larger grids naturally have higher blurring in the defocused pixel locations. All grids are centered around the same lenslet image and the shift-sum parameter is fixed across all renderings. The rendered images thereby correspond to increasing aperture sizes for the same view and serve as ideal training samples for blur magnification.

We divide the light-field dataset into training and test splits of 3000 and 343 light-fields each. Each rendered finite-aperture image is cropped to a square aspect ratio and rescaled to a size of 256×256

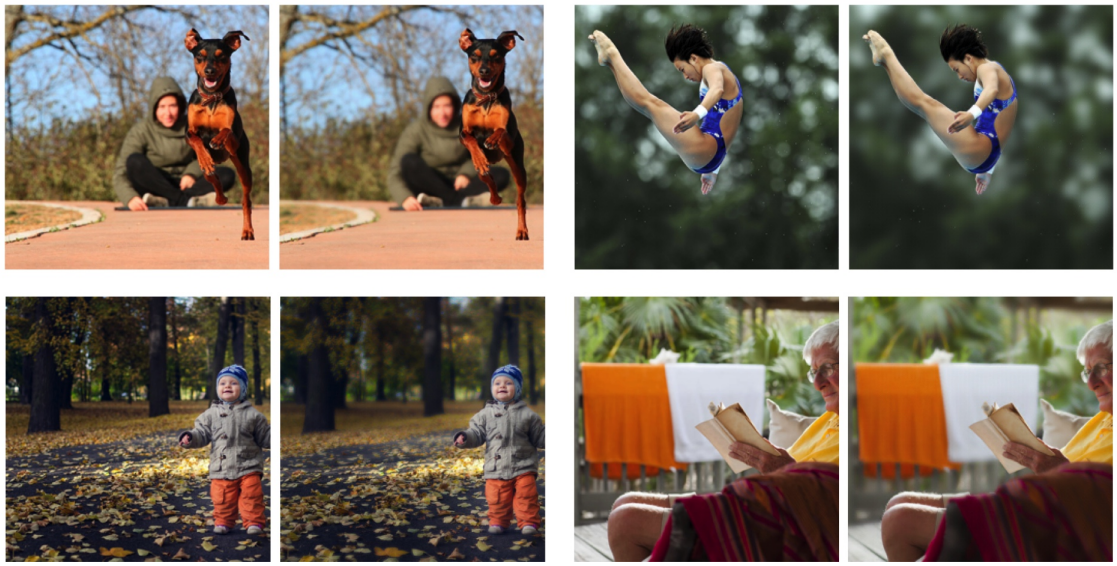

Figure 5.5 This figure shows the generalization of our defocus magnification network on images selected from the portrait dataset [25] and the blur detection dataset [114]. For each pair: Input image on the left, Defocus magnified image ($m = 2$) on the right. Note that these images are significantly different from images in the training split of the light-field dataset.

pixels before computing the composite focus measure over it. The input image to the network is a $256 \times 256 \times 4$ sized RGBF image. The magnification parameter m is provided as an input based on the target wide-aperture image. We augment the training data by rotating the square input images by $90°$, $180°$ and $270°$. This improves the generalization of our network on images that were not a part of the training data. Our training experiments are performed on an Nvidia GTX 1080Ti and the generator network is trained for 30 epochs. We use the Adam solver [50] for gradient descent. The learning rate is initially set to 10^{-4} and is linearly decreased to zero during the second half of the training stage. To test the quality of generalization of our network, we sample a set of portrait images from the portrait dataset of [25]. The images from the portrait dataset include human and other subjects which were not a part of the training data. The additional loss term based on the composite focus measure along with adversarial training enables our network to generalize to unseen data, which is demonstrated in Figure 5.5.

5.2 Single Image Scene Refocusing

In addition to single-image defocus magnification (or portraiture), we also propose a method for comprehensive refocusing of a scene from a single image. A standard approach to scene refocusing uses several wide-aperture images from a focal stack to generate a new image with the target depth-of-field. Refocusing is typically modeled as a composition of pixels from several focal slices to create a new pixel intensity. This reduces the task of refocusing to selecting a set of weights for each pixel across focal slices as is described in [46]. Other methods that use all the slices of a focal stack first estimate

the depth map of the scene and a corresponding radiance image, and then convolve the radiance image with geometrically accurate blur kernels, such as in [36]. In the case of single images, it is difficult to simultaneously estimate the true radiance as well as the defocus radius at each pixel. Moreover, the complexity of the size and shape of the defocus kernel at each pixel depends on the scene geometry as well as the quality of the lens as discussed in the previous chapter. A deep learning approach to refocus a wide-aperture image using a single end-to-end network does not perform very well and this is discussed in more detail in Section 5.3.

We approach the problem of refocusing a wide-aperture image as a cascaded operation involving two steps in the image space. The first step is a deblurring operation that computes the true scene radiance $\hat{G}^r$ from a given wide-aperture image G^i, where i denotes the focus position during capture. This involves deblurring each pixel in a spatially varying manner in order to produce locally sharp pixels. The second step applies a new spatially varying blur to all the sharp pixels to generate the image corresponding to the new focus position $G^{i+\delta}$, where δ denotes the change in focus position. The required scene-depth information for geometric refocusing can be assumed to be implicit within this two-stage approach. Srinivasan et al. [117] have shown how the forward process of blurring can actually be used to compute an accurate depth-map of the scene. Our two-stage approach to refocusing a wide-aperture image is briefly described below.

In the first stage, an in-focus radiance image is computed from a given wide-aperture image G^i and an additional focus measure m evaluated over G^i. The focus measure provides a useful cue that improves the quality of deblurring:

$$\hat{G}^r = \mathcal{G}^1_{\theta_G}\left(G^i : m(G^i)\right) \tag{5.9}$$

In the second stage, the generated in-focus image is used together with the input wide-aperture image to generate the target image corresponding to a shifted focus position $i + \delta$.

$$G^{i+\delta} = \mathcal{G}^2_{\theta_G}\left(G^i : \hat{G}^r, \delta\right) \tag{5.10}$$

We train end-to-end conditional adversarial networks for both these stages. While the deblurring network $\mathcal{G}^1_\theta$ is motivated by existing blind image-deblurring works in the literature, we provide motivation for our second network $\mathcal{G}^2_\theta$ by producing a far-focused slice from a near-focused slice using a simple optimization method.

Adversarial Learning: As described in section 5.1.1., generative adversarial networks (GANs) [31] define the task of learning as a competition between two networks, a generator and a discriminator. The task of the generator is to create an image based on an arbitrary input, typically provided as a noise vector, and the task of the discriminator is to distinguish between a real image and this generated image. The generator is trained to created images that are perceptually similar to real images, such that the discriminator is unable to distinguish between real and generated samples. The objective function of adversarial learning can be defined as:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}_{GAN}, \tag{5.11}$$

where $\mathcal{L}_{GAN}$ is the classic GAN loss function:

$$\mathcal{L}_{GAN} = E_{y \sim p_r(y)} \left[\log \mathcal{D}(y) \right] + E_{z \sim p_z(z)} \left[\log(1 - \mathcal{D}(\mathcal{G}(z))) \right], \tag{5.12}$$

where $\mathcal{D}$ represents the discriminator, $\mathcal{G}$ is the generator, y is a real sample, z is a noise vector input to the generator, p_r represents the real distribution over target samples and p_z is typically a normal distribution.

Conditional adversarial networks (cGANs), provide additional conditioning to the generator to create images in accordance with the conditioning parameters. Isola et al. [44] propose a robust cGAN architecture called pix2pix, where the generator learns a mapping from an image x and a noise vector z to an output image y as: $\mathcal{G} : x, z \rightarrow y$. The observed image is provided as conditioning to both the generator and the discriminator. We use cGANs for the tasks of de-blurring and refocusing and provide additional conditioning parameters to both our networks as defined in the following sections.

5.2.1 Deblurring a Wide-Aperture Image

We use a conditional adversarial network to deblur a wide aperture image G^i and estimate its corresponding scene radiance $\hat{G^r}$ as described in Equation 5.9. Our work draws inspiration from several deep learning methods for blind image-deblurring such as [57, 83, 109, 137]. Our network is similar to the state-of-the-art deblurring network proposed by Orest et al. [57]. Our generator network is built on the style transfer network of Johnson et al. [47] and consists of two strided convolution blocks with a stride of $\frac{1}{2}$, nine residual blocks and two transposed convolution blocks. Each residual block is based on the ResBlock architecture [41] and consists of a convolution layer with dropout regularization [119], instance-normalization [130] and ReLU activation [84]. The network learns a residual image since a global skip connection (ResOut) is added in order to accelerate learning and improve generalization [57]. The residual image is added to the input image to create the deblurred radiance image. The discriminator is a Wasserstein-GAN [5] with gradient penalty [33] as defined in [57]. The architecture of the critic discriminator network is identical to that of PatchGAN [44, 68]. All convolution layers except for the last layer are followed by instance normalization and Leaky ReLU [136] with an α=0.2.

The cGAN described in [57] is trained to sharpen an image blurred by a motion-blur kernel of the form $I_B = K * I_S + \eta$, where I_B is the blurred image, I_S is the sharp image, K is the motion blur kernel and η represents additive noise. In our case, the radiance image G^r has been blurred by a spatially varying defocus kernel and therefore the task of deblurring is more complex. We thereby append the input image G^i with an additional channel that encodes a focus measure response computed over the input image. We compute $m(G^i)$ as the response of the Sum-of-modified-Laplacian (SML) [90] filter applied over the input image. We also provide the input image along with this additional channel as conditioning to the discriminator. The adversarial loss for our deblurring network can be defined as:

$$\mathcal{L}_{cGAN} = \sum_{n=1}^{N} -\mathcal{D}^1_{\theta_D}(\mathcal{G}^1_{\theta_G}(x^i), x^i), \tag{5.13}$$

where $x^i = G^i : m(G^i)$ is the input wide-aperture image G^i concatenated with the focus measure channel $m(G^i)$.

In addition to the adversarial loss, we also use perceptual loss [47] as suggested in [57]. Perceptual loss is L2-loss between the CNN feature maps of the generated deblurred image and the target image:

$$\mathcal{L}_X = \frac{1}{W_{ij}H_{ij}} \sum_x \sum_y (\phi_{ij}(I^S)_{xy} - \phi_{ij}(\mathcal{G}_{\theta_G}(I^B))_{xy})^2, \tag{5.14}$$

where ϕ_{ij} is the feature map in VGG19 trained on ImageNet [21] after the j^{th} convolution and the i^{th} max-pooling layer and W and H denote the size of the feature maps. In this case, I^S and I^B represent the ground truth in-focus image and the input wide-aperture image respectively. The loss function for the generator is a weighted combination of adversarial and perceptual loss $\mathcal{L} = \mathcal{L}_{cGAN} + \lambda\mathcal{L}_X$.

The structure of our deblurring cGAN is shown in Figure 5.6. A few wide-aperture images along with the computed in-focus radiance image are shown in Figure 5.15.

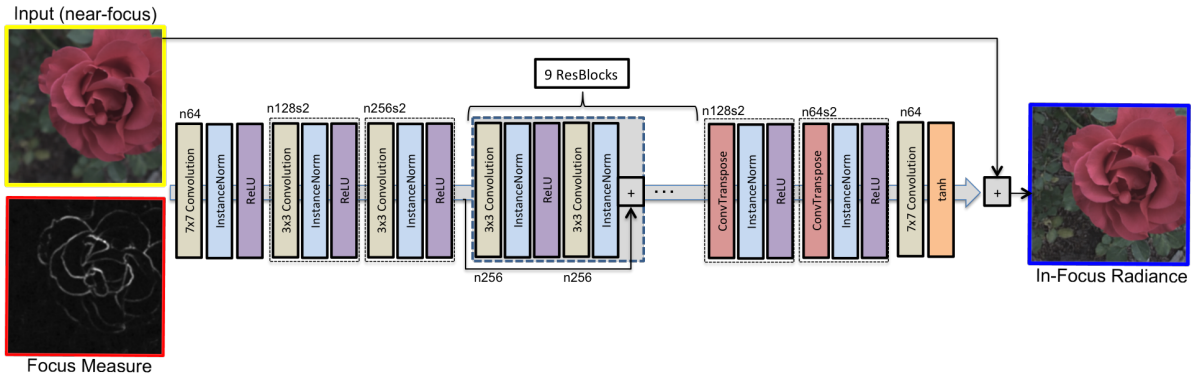

Figure 5.6 The architecture of the deblurring cGAN. It receives a wide-aperture image and its focus measure channel as input and computes an in-focus radiance image.

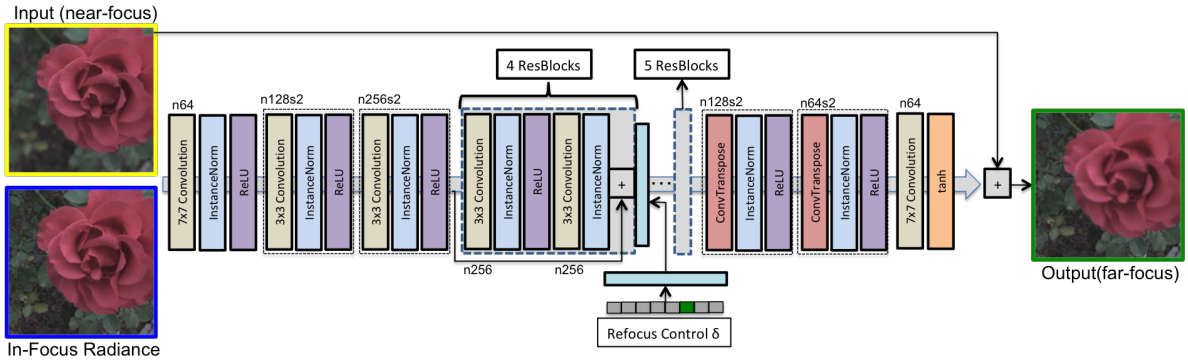

Figure 5.7 The architecture of the refocusing cGAN. It uses the generated in-focus image together with the original wide-aperture image and a refocus control parameter δ to compute a refocused image.

5.2.2 Refocusing a Wide-Aperture Image

The in-focus image computed from the above network not only represents the true scene radiance at each pixel, but can also serve as proxy depth information in conjunction with the input wide-aperture image. We motivate our second refocusing network $\mathcal{G}^2_{\theta_G}$ using a simple method that can refocus a near-focus image to a far-focus image and vice versa, using the computed radiance image.

As shown in the example in Figure 5.8, a near-focused image G^1 can be converted to a far focused image G^n using the radiance image $\hat{G}^r$ resulting from the deblurring network. Here 1 and n are used to denote the near and far ends of the focus spread of a focal stack. To refocus these images, the first step would be to compute the per-pixel blur radius between the input image G^1 and the radiance image $\hat{G}^r$. This can be achieved using a blur-and-compare framework wherein the in-focus pixels of the radiance image are uniformly blurred by different radii and the best defocus radius σ is estimated for each pixel using pixel-difference between a blurred patch and the corresponding patch in G^1. Inverting these defocus radii as $\sigma' = \sigma_{max} - \sigma$ followed by re-blurring the radiance image is the natural way to create the refocused image. This method can also be used to convert a far-focused image to a near focused image as shown in the second row of Figure 5.8. Free-form refocusing between arbitrary focus positions is not trivial though since there is no front-to-back ordering information in the estimated defocus radii.

For free-form scene refocusing, we use a conditional adversarial network similar to our deblurring network. We use the same cGAN architecture of the previous section, with different conditioning and an additional refocus control parameter δ. The refocus control parameter is used to guide the network to produce a target image corresponding to a desired focus position. The input to the network is the original wide-aperture image G^i concatenated with the scene radiance image $\hat{G}^r = \mathcal{G}^1_{\theta_G}(G^i : m(G^i))$ computed by the deblurring network. The refocus parameter δ encodes the shift between the input and output images and is provided to the network as a one-hot vector. The refocus vector corresponding to δ is concatenated as an additional channel to the innermost layer of the network, using a fully connected layer to convert the one-hot vector into a 64×64 channel.

The structure of the refocusing cGAN is shown in Figure 5.7. We use the same structure for the discriminator and the generator as that of the deblurring cGAN. The loss function for the generator is a summation of adversarial loss and perceptual loss. The discriminator network is conditioned using the input image and the in-focus radiance image. The cGAN loss for this network can be defined as:

$$\mathcal{L}_{cGAN} = \sum_{n=1}^{N} -\mathcal{D}^2_{\theta_D}(\mathcal{G}^2_{\theta_G}(x^i), x^i), \tag{5.15}$$

where $x^i = G^i : \hat{G}^r$ is the input wide-aperture image G^i concatenated with the scene radiance image $\hat{G}^r = \mathcal{G}^1_{\theta_G}(G^i : m(G^i))$. Refocused images generated from the input wide-aperture image, the in-focus image and different refocus parameters are shown in Figures 5.16,5.17.

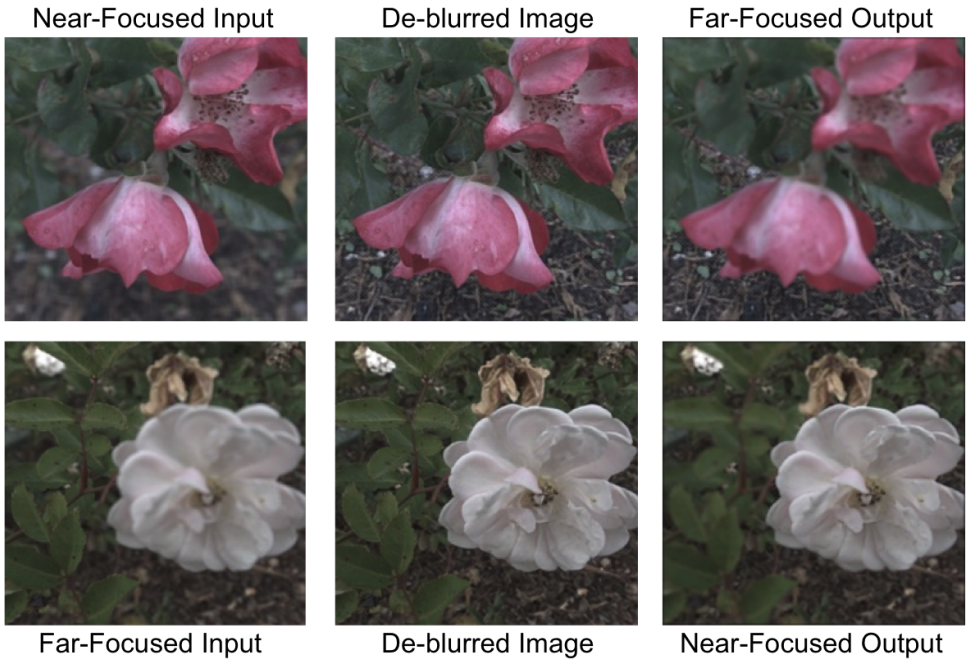

Figure 5.8 Refocusing using a simple image-processing operation over the input wide-aperture image G^1 and the deblurred in-focus image $\hat{G}^r$. The first row shows the input near-focused image, the de-blurred in-focus image from the network and the computed far-focused image. The second row shows equivalent far-to-near refocusing.

5.2.3 Training Details

For training both networks, we compute multiple wide-aperture images from the light-field dataset of scenes consisting of flowers and plants [118]. The method used to generate training images from light-fields is explained in the following section.

5.2.3.1 Focal Stacks from Light-Fields

A focal slice G^i, as described in previous chapters, is a wide-aperture image corresponding to a focus position i and can be defined as:

$$G^i = \int \int h^i(x, y, d_{x,y}) * \hat{G}^r(x, y)\, dx\, dy\,, \tag{5.16}$$

where h^i is the spatially varying blur kernel dependent on the spatial location of the pixel and the depth $d_{x,y}$ of its corresponding scene point and $\hat{G}^r$ is the true radiance of the scene point which is usually represented by the in-focus intensity of the pixel.

Focal slices, represented in equation 5.16, can be created from a light-field image of the scene. The Lytro light-field camera based on Ng et al. [73] captures a 4D light-field of a scene in a single shot, and can thereby be used for dynamic scenes. The different angular views captured by a light-field camera

Figure 5.9 A few examples of the light-field images in the Flowers dataset of [118].

can be merged together to create wide-aperture views corresponding to different focus positions. A large number of light-fields of structurally similar scenes have been captured by Srinivasan et al. [118]. Several other light-field datasets also exist such as the Stanford light-field archive [120] and the light-field saliency dataset [69]. Large quantities of similar focal stacks can be created from such light-field datasets.

Srinivasan et al. [118] captured a large light-field dataset of 3343 light-fields of scenes consisting of flowers and plants using the Lytro Illum Camera. Each image in the dataset consists of the angular views encoded into a single light-field image. A 14×14 grid of angular views can be extrapolated from the light-field, each having a spatial resolution of 376×541. Typically, only the central 8×8 views are useful as the samples towards the corners of the light-field suffer from clipping as they lie outside the camera's aperture. This dataset is described in detail in [118]. A few sample images from this dataset are shown in Figure 5.9. For our experiments, we use a central 7×7 grid of views to create focal stacks, so as to have a unique geometric center to represent the in-focus image. We generate a focal stack at a high focus resolution for each of these light-field images using the synthetic photography equation defined in [73]:

$$G^i(s,t) = \int\int L\left(u, v, u + \frac{s-u}{\alpha_i}, v + \frac{t-v}{\alpha_i}\right) du\, dv. \tag{5.17}$$

Here G^i represents the synthesized focal slice, $L(u,v,s,t)$ is the 4D light-field represented using the standard two-plane parameterization and α_i represents the location of the focus plane. This parameterization is equivalent to the summation of shifted versions of the angular views captured by the lenslets as shown in [73]. We vary the shift-sum parameter linearly between -s_{max} to +s_{max} to generate 30 focal slices between the near and far end of focus.

To reduce the size of focal stacks to an optimal number of slices, we apply our composite focus measure [107] and study the focus variation of pixels across the stack. For each pixel, we record the normalized response of the composite measure at each slice. We build a histogram of the number of pixels that peaked at each of the 30 slices across the 3343 light-field dataset. We find that in close to 90% of the images, all the pixels peak between slices 6 and 15 of the generated focal stack. The depth variation of the captured scenes is mostly covered by these ten focal slices. We thereby subsample each focal stack to consist of ten slices, varying from slice 6 to slice 15 of our original parameterization. Our training experiments use these 10-sliced focal stacks computed from the light-field dataset.

Config	Loss Function for $\mathcal{G}$	Train PSNR (dB)	Train SSIM	Test PSNR (dB)	Test SSIM
1	Without $\mathcal{L}_{cGAN}$	38.373	0.947	38.170	0.945
2	Without $\mathcal{L}_X$	35.825	0.960	35.946	0.961
3	Without $\mathcal{L}_f$	42.467	0.987	43.782	0.989
4	With $\mathcal{L}_{cGAN}, \mathcal{L}_X, \mathcal{L}_f$	**43.622**	**0.990**	**45.112**	**0.991**

Table 5.1 Our experiments on training and test splits of the light-field dataset. PSNR and SSIM values are computed between ground truth wide-aperture images and generated images from the network. The configuration 4 network, using all three loss functions, works best.

For training, the 3343 focal stacks are partitioned into 2500 training samples and 843 test samples. Each focal slice is cropped to a spatial resolution of 256×256 pixels. The s_{max} parameter while computing focal slices is set to 1.5 pixels. For the deblurring network, we use all the ten focal slices from the 2500 focal stacks for training. For the refocusing network, we experiment with three different configurations. In the first configuration a single refocus parameter of $\delta = +8$ is used. In the second configuration, the refocus parameter has four distinct values: $\delta = \{-9, -5, +5, +9\}$. In the third configuration, the refocus parameter can take any one of 19 possible values from -9 to $+9$. The deblurring network is trained for 30 epochs ($\sim$50 hours) and all configurations of the refocusing network are trained for 60 epochs ($\sim$45 hours). All training experiments were performed on an Nvidia GTX 1080Ti. The learning rate is set to 0.0001 initially for all network configurations. The learning rate is linearly decreased to zero after half the total number of epochs are completed. All networks are trained for a batch size of 1 and the Adam solver [50] is used for gradient descent. The λ parameter for scaling content loss is set to 100 as suggested in [57].

5.3 Experiments and Results

For defocus magnification we perform quantitative evaluation of our network by analysing the performance of defocus magnification over the training and test splits of the light-field dataset. We report the PSNR and the SSIM between ground truth and generated wide-aperture images in Table 5.1. The table shows four configurations of our generator network. The first configuration does not use $\mathcal{L}_{cGAN}$ during training, the second does not use $\mathcal{L}_X$, the third configuration does not use $\mathcal{L}_f$ and the fourth configuration, which is our proposed network, uses all three loss terms while training. It can be seen that $\mathcal{L}_{cGAN}$ and $\mathcal{L}_X$ have a higher contribution than $\mathcal{L}_f$, however the improvement in both training and test performance on using $\mathcal{L}_f$ suggests an overall benefit when the focus measure loss term is used.

In Figures 5.4 and 5.5 we show the qualitative performance of our defocus magnification network on narrow-aperture images. Image pairs in Figure 5.4 demonstrate blur magnification on narrow-aperture images from the test split of the light-field dataset with magnification parameters of $m = 2$ (left column) and $m = 3$ (right column). The focused pixels remain in-focus and the blur at all other pixel locations is magnified. Figure 5.5 demonstrates defocus magnification with magnification parameter of $m =$

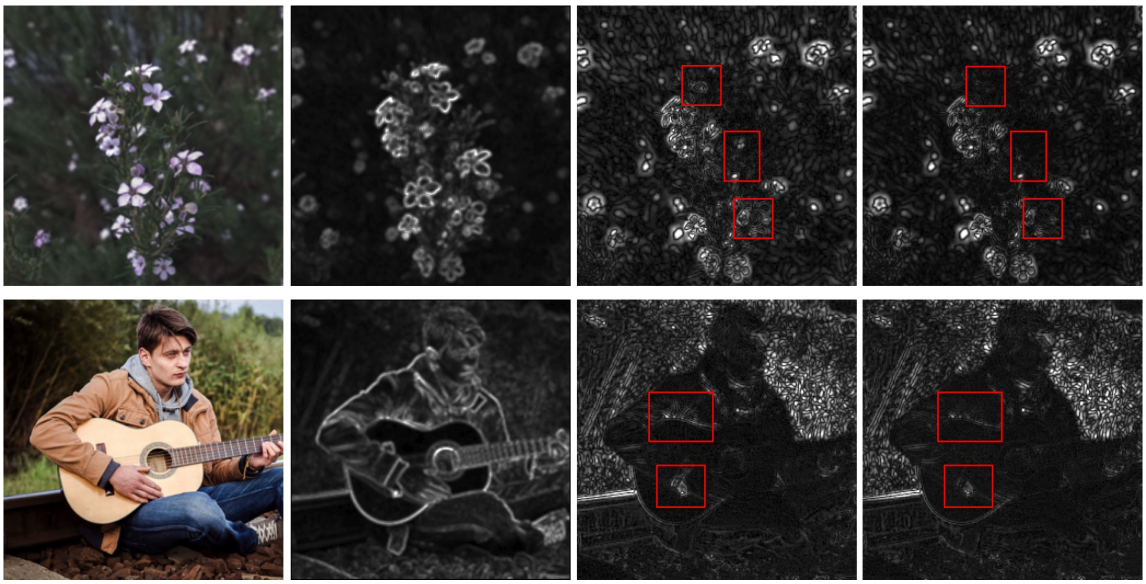

Figure 5.10 Comparison of our network configurations 3 and 4 from Table 5.1. Left-to-right: Input image, Focus Measure Response, difference image $|\mathcal{I} - \mathcal{I}'|$ for network 3, difference image $|\mathcal{I} - \mathcal{I}'|$ for network 4. Difference images are expected to be zero at in-focus pixels and high at other pixels. It can be seen that our proposed network 4 is better for both in-focus and defocused pixels. Images are best viewed in the electronic version.

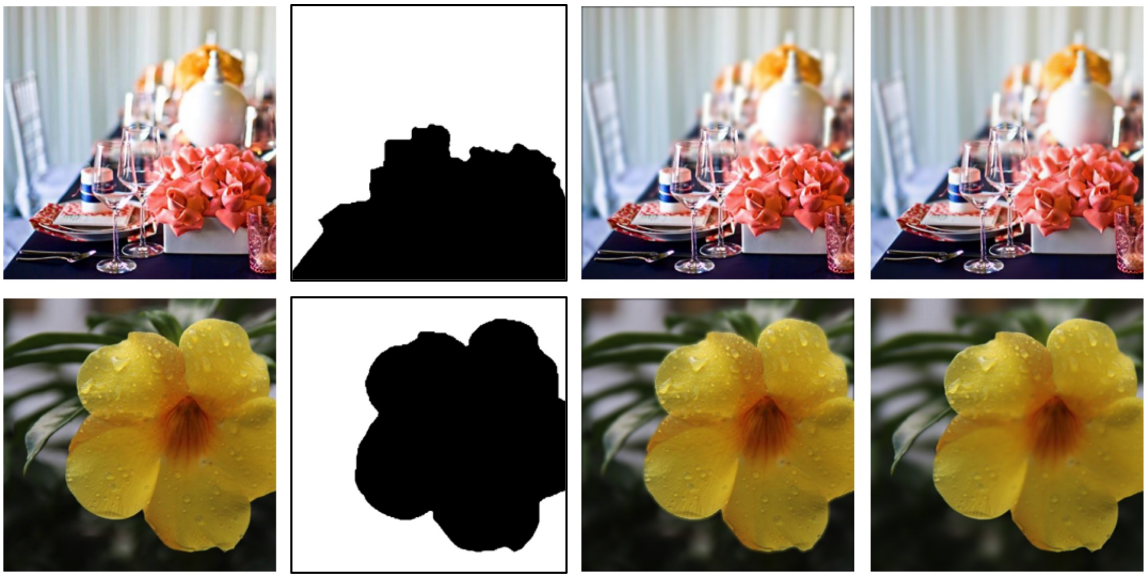

Figure 5.11 Left to Right: Input Image, ground truth defocus map, depth-aware defocus magnification (generated by re-blurring all pixels from defocused regions of the ground truth defocus map), our defocus magnified image with $m = 1$. Our network produces defocus magnification that is very close to ground truth re-blurring, while carefully preserving the in-focus pixels.

Figure 5.12 First Row: Input image, defocus magnification of Ma et al. [74], our defocus magnification. Our method keeps the in-focus pixels intact at depth-edges along the hair. Second Row: Input image, defocus magnification of Park et al. [95], our defocus magnification. Our approach keeps the in-focus pixels intact near depth edges along the hair and the face.

Figure 5.13 Our network can be used to iteratively increase the amount of defocus. From left to right: consecutive blur magnification of $m = 2$ on the input image (leftmost column). Integrity of foreground pixels is preserved even after multiple iterations of blur magnification.

Table 5.2 Quantitative evaluation of our deblurring network. PSNR and SSIM is reported for the test-split of the light-field dataset. We compare the performance of the deblurring network with and without the additional Sum-of-Modified-Laplacian (SML) focus measure channel. There is a marginal but useful improvement in the quality of deblurring on using the focus measure channel. As an indication of overall performance, we generate an in-focus image using the composite focus measure [107] applied on all slices of the focal stack and report its quality. Note that our method uses only a single image.

Deblurring Experiment	PSNR	SSIM
Ours (without additional Focus Measure)	34.88	0.937
Ours (with additional Focus Measure)	35.02	0.938
Composite Focus Measure (uses entire stack)	38.697	0.965

2 on images selected from the portrait dataset [25] and blur detection dataset [114]. These images are substantially different from our training images which do not consist of any human subjects. Our network magnifies the blur even in slightly defocused foreground segments, thus truly simulating a shallow depth-of-field.

To study the qualitative improvement provided by our focus measure loss term, we compare the blur magnification of the network configurations 3 and 4 from Table 5.1. We generate the target wide-aperture image using both networks and study the difference between the input and the output images. The difference is ideally expected to be zero at the in-focus pixels and high at the defocused pixels. The difference images are shown in Figure 5.10. The difference images corresponding to configuration 3 are shown in the third column and those from configuration 4 are shown in the fourth column of Figure 5.10. Higher errors in the in-focus regions of configuration 3 suggest that the focus measure loss term is useful for keeping the in-focus regions intact and blurring other regions. We compare our method with state-of-the-art methods [74] and [95] in Figure 5.12. Our approach keeps in-focus pixels intact, specifically at the boundaries of the foreground subject. The methods [74] and [95] fail at depth-edges because of erroneous labeling in the estimated defocus maps. Our blur magnification network can also be used iteratively till the desired blurring in the background is achieved. In Figure 5.13, we show the effect of iteratively applying a magnification of $m = 2$ on the input image on the left. In Figure 5.11, we show how our network generates close to ground-truth defocus magnification by comparing our synthesized images with depth-aware defocus rendering using the defocus-maps of [114].

For scene refocusing we provide a quantitative evaluation of the performance of our two-stage refocusing approach in Tables 5.2, 5.3. We compare the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) of the refocused images with the ground truth images from the focal stacks. Since this is the first work that comprehensively manipulates the focus position from a single image, there is no direct comparison of the generated refocused images with existing geometric techniques over focal stacks. However, we generate in-focus images using the composite focus measure [107] applied across the full focal stack and report the quantitative reconstruction quality in Table 5.2. We show the quantita-

Table 5.3 Quantitative evaluation of our refocusing network. The PSNR and SSIM values are reported on the test-split of the light-field dataset. The first two rows show the performance of our refocusing network without an additional in-focus image. This corresponds to an end-to-end, single stage approach to refocusing. The next three rows show the performance on using different refocus control parameters in our two-stage experiments. The final row shows the test performance of our refocusing network which was trained using ground truth in-focus images G^r but tested using the radiance images computed by the deblurring network $\hat{G}^r$. Note that the two-stage approaches significantly outperform their single-stage counterparts. The high PSNR and SSIM values quantitatively suggest that our network enables high-quality refocusing.

Experiment	Type	Refocus Control Steps	PSNR	SSIM
Without G^r	single-stage	+8	38.73	0.97
Without G^r	single-stage	{-9,-5,+5,+9}	38.4	0.956
With G^r	two-stage	+8	44.225	0.992
With G^r	two-stage	{-9,-5,+5,+9}	43.4	0.988
With G^r	two-stage	{-9,-8,..,0,..,+8,+9}	40.42	0.975
With AIF($\hat{G}^r$)	two-step	-9,-5,+5,+9	38.63	0.958

tive performance of our networks individually and report the PSNR and SSIM of the computed in-focus radiance image in comparison with the ground-truth central light-field image.

Our two-stage approach to refocusing is motivated by our initial experiments wherein we observed that an end-to-end refocusing network does not work well. Our experiments spanned several network architectures such as the purely convolutional architecture of the disparity estimation network of [48], the separable kernel convolutional architecture of [137], the encoder-decoder style deep network with skip-connections of [121] and the conditional adversarial network of [57]. These networks exhibit poor refocusing performance in both cases of fixed pairs of input-output focal slices as well as for the more complex task of free-form refocusing. Since the networks are only given input wide-aperture images while training, there may be several pixel intensities which do not occur sharply in either the input or output images, and the task of jointly estimating all true intensities and re-blurring them is difficult to achieve within a reasonable compute power/time budget for training. In Table 5.3, we compare our two-stage approach to refocusing with an equivalent single-stage, end-to-end network. This essentially compares the performance of our refocusing network with and without the additional radiance image computed by the deblurring network. It can be seen that the two-stage method clearly outperforms a single-stage approach to refocusing.

The deblurring network uses an additional focus measure channel to compute the radiance image $\hat{G}^r$. The benefit of using the focus measure is indicated in Table 5.2. For the refocusing network, we perform experiments on three different configurations. The configurations differ from each other in the number of refocus control parameters and are shown in Table 5.3. The first configuration is a proof-of-concept network and is trained on a single refocus parameter. This clearly exhibits the best performance as the training samples have a high degree of structural similarity. The network with four

control parameters performs better than the network with 19 parameters, which can be seen in Table 5.3. This can be attributed to two separate issues. The focal stacks created from the light-field dataset consist of ten slices that roughly span the depth range of the scene from near-to-far. However, in the absence of scene content at all depths, certain focal slices may be structurally very similar to adjacent slices. Training these slices with different control parameters can confuse the network. Secondly, in the case of the 19 parameter configuration, the total number of training samples increases to 250000 as there are 100 samples from each focal stack. We use a subset of size 30000 from these training images sampled uniformly at random. In the case of refocusing with 4 control parameters, the focus shift between input and output images is clearly defined and the network thereby captures the relationship better. All the training samples from the dataset can be used directly to train this network as there are only 12 training samples per focal stack in the four parameter configuration.

Figure 5.14 The performance of our two-stage refocusing framework on generic images. The first row has the input wide-aperture image and the second row shows the refocused image. The first four columns show the performance on structurally different light-field focal slices from another light-field dataset while the last column shows the performance on an image captured by a wide-aperture camera.

We show qualitative deblurring and refocusing results for several test samples in Figures 5.15,5.16,5.17. In Figure 5.14, we show the performance of our refocusing framework on generic images from different light-fields that were not images of flowers or plants, and also show the performance on an image captured using a wide-aperture camera. The performance suggests that our networks are implicitly learning both tasks quite well and can be used for high-quality refocusing of standalone images.

5.4 Summary

In this chapter, we presented an end-to-end approach for single-image defocus magnification and a a two-stage approach for comprehensive scene refocusing over a single-image. Our DefocusGAN and RefocusGAN frameworks use adversarial training and perceptual loss to train separate deblurring and

Figure 5.15 In-focus radiance images created using the deblurring network. The top row shows the input wide-aperture images and the bottom row shows the deblurred output from our deblurring network.

refocusing networks. We provide a focus measure channel as an additional conditioning for deblurring a wide- aperture image. We use the deblurred in-focus image as an additional condition- ing for refocusing and also propose a novel focus-measure based loss term. Our quantitative and qualitative results suggest high-quality performance on portraiture and refocusing. Our networks exhibit limited generalization and can further benefit from fine-tuning and training over multiple datasets.

Figure 5.16 Near-to-Far Refocusing generated with δ=+9 using our refocusing network. The top row shows the input wide-aperture images and the bottom row shows the output refocused images.

Figure 5.17 Far-to-Near Refocusing generated with δ=−9 using our refocusing network. The top row shows the input wide-aperture images and the bottom row shows the output refocused images.

Chapter 6

Focal Stacks for Computer Vision

Focal stacks as a collection of multi-focus images of the scene consist of proxy scene-depth information as scene points from different depths appear in-focus in different focal slices. If the order and focus position of the focal slices is known, it is possible to convert in-focus slice labels into a depth-map of the scene. Focal stacks can thus find use in several depth computation algorithms. In this chapter, we discuss the different applications for focal stacks for computer vision problems. We discuss how a focal stack can be converted into a dense depth-map using the composite focus measure discussed previously. We then show how a sparse depth-map with quick focus measure computation can be used for dense view interpolation on mobile devices.

6.1 Depth from Focus

Recovering the 3D structure of the scene from 2D images has been an important pursuit in computer vision. The size, relative position and shape of scene objects play an important role in understanding the world around us. The 2.5D depth map is a natural description of scene structure, corresponding to an image from a specific viewpoint. Multi-camera arrangements, structured lights, focus stacks, shading etc., can all recover depth maps under suitable conditions. Users' experience and understanding of the environment around them can be improved significantly if the 3D structure is available. The emergence of Augmented and Virtual Reality (AR/VR) as an effective user interaction medium enhances the importance of easy and inexpensive structure recovery of everyday environments around us.

Depth sensors using structured light or time-of-flight cameras are common today, with a primary use as game appliances [45]. They can capture dynamic scenes but have serious environmental, resolution and depth-range limitations. Multi-camera setups are more general, but are unwieldy and/or expensive. Focus and defocus can also provide estimates of scene depth. Today's DSLR cameras and most mobile cameras can capture focal stacks by manipulating the focus distance programmatically. Thus, depth from focus is a promising way to recover 3D structure of static scenes as it is accessible widely, albeit at a loss of temporal resolution.

We present a scheme to recover high quality depth maps of static scenes from a focal stack, improving on previous depth-from-focus (DfF) methods. We show results on several everyday scenes with different

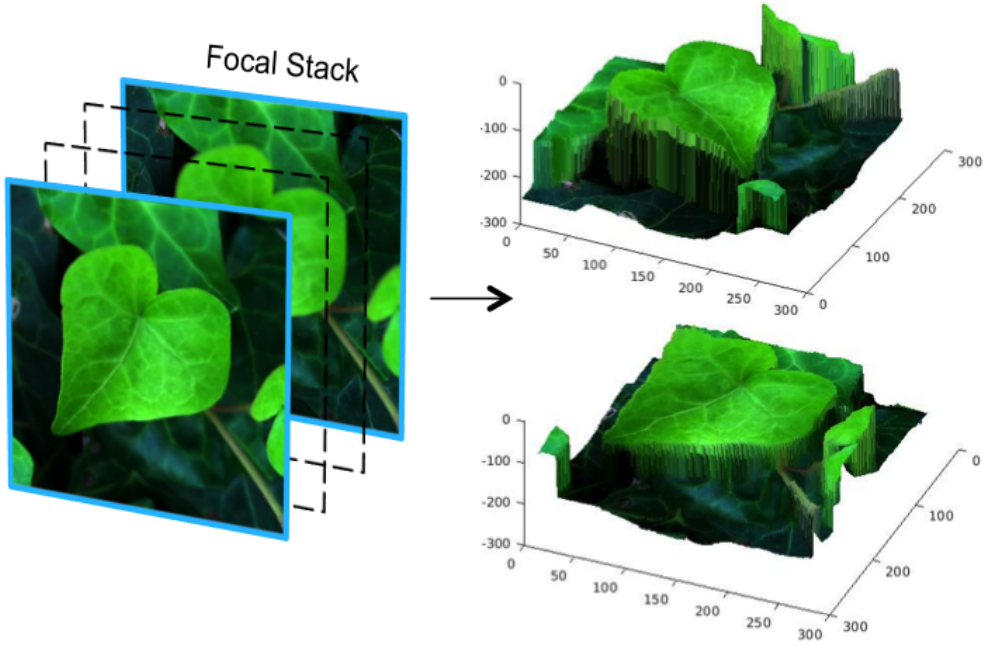

Figure 6.1 A coarse focal stack of an outdoor scene and its surface-mapped 3D depth is shown from two different viewpoints. The depth-map is computed using our composite focus measure. The smooth depth variation along the midrib of the leaf is clearly visible in the reconstructed depth rendering.

depth ranges and scene complexity. Figure 6.1 is an example of robust depth recovery that we facilitate. We use a two-stage pipeline for DfF, with the first stage estimating a fine depth at each pixel using a Laplacian fit over the composite focus measure. This gives both a depth estimate and a confidence value for it. In the second stage, a cost-volume propagation step distributes the confident depth values to their neighborhoods using an all-in-focus image as a guide.

We present qualitative and quantitative results on a large number and variety of scenes, especially everyday scenes of interest. The depth maps we compute can be used for applications that RGBD images are used for, typically at resolutions and fidelity higher than them.

6.1.1 Depth Estimation and Propagation

Figure 6.2 shows the pipeline of our depth-from-focus method. We first build a high resolution but noisy depth-map by fitting a Laplacian distribution to the composite focus measure at each pixel. We then build a high-resolution cost volume (256 depth labels) corresponding to the confident depth labels and use an MRF-based in-focus image for guidance to compute a smooth depth map of the scene.

6.1.1.1 Depth from Laplacian Regression

A Laplacian distribution has been shown to be a good model for depth [108] as it captures sharp depth edges well. Since the focus profile of a pixel is expected to be closely related to its depth profile,

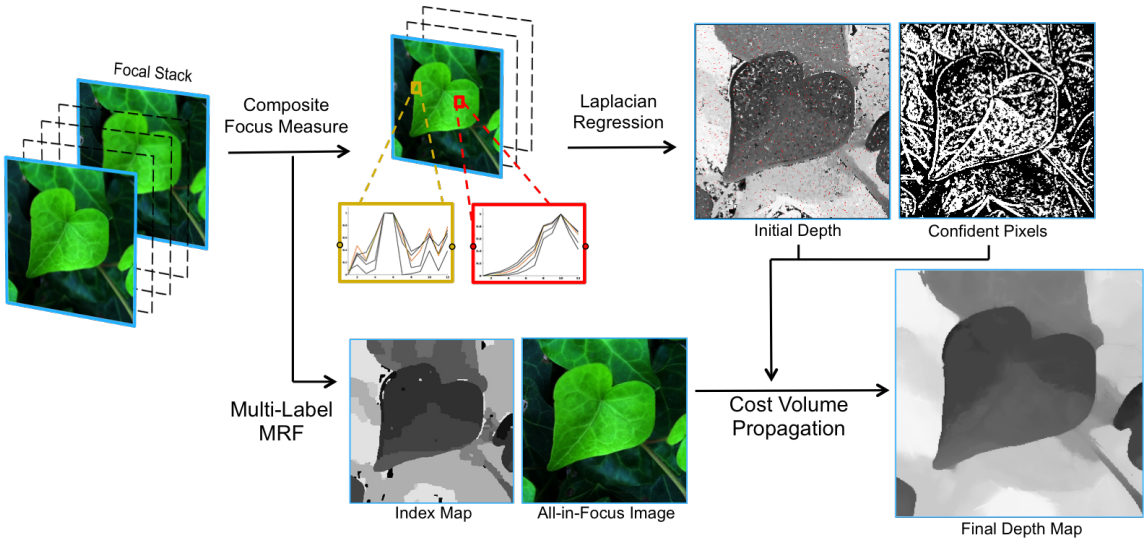

Figure 6.2 Our proposed pipeline to compute smooth depth-maps from focal stacks. The composite focus measure is evaluated at each pixel of the focal stack and the focus responses are used to (a) generate a high resolution depth value at each pixel using Laplacian regression and (b) generate an all-in-focus image using a multi-label MRF optimization. The all-in-focus image and the confident high resolution depths are used together to compute a smooth depth map using Cost-Volume Filtering.

we estimate the depth of a pixel by fitting a non-linear Laplacian distribution over its composite focus measure. For each pixel, we collect the focus responses of the composite focus measure as a set of data points (insets of Figure 6.2) and fit a Laplacian distribution over them. The Laplacian distribution has the form

$$g(x|\,\mu, b) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}, \tag{6.1}$$

with μ denoting the location and b denoting the scale or diversity.

We use a standard iterative non-linear regression framework for least squares fitting at each pixel. The estimated μ represents a smooth depth value. The real-valued μ estimates have a much finer resolution than the number of focal slices in the stack. We linearly rescale the values of μ from $[1, L] \subset \mathbb{R}$ to $[0, 255] \subset \mathbb{Z}$, representing high resolution depths. This linear scaling can be appropriately adjusted based on the blur between pairs of focal slices if the focal stack was captured with non-uniform focus steps. The rescaled μ at each pixel is notated as the initial depth $D_i(p)$ at the pixel. Laplacian fitting over the composite focus measure is a departure from standard DfF methods which simply assign the focal slice label at which a focus measure peaks. For example, in Figure 6.2, the focal stack consists of 11 focal slices and the depth resolution reported in several DfF methods is thereby limited to 11 depths, similar to the index map shown in the figure. Our initial depth after Laplacian regression (right-hand side of Figure 6.2) is already made up of 243 unique depth values.

The scale b of the Laplacian encodes the confidence of the depth value. Higher the value of b, lower is the confidence of computed depth. After normalizing the values of b, the confidence at each pixel is recorded as $D_c(p) = 1 - b(p)$.

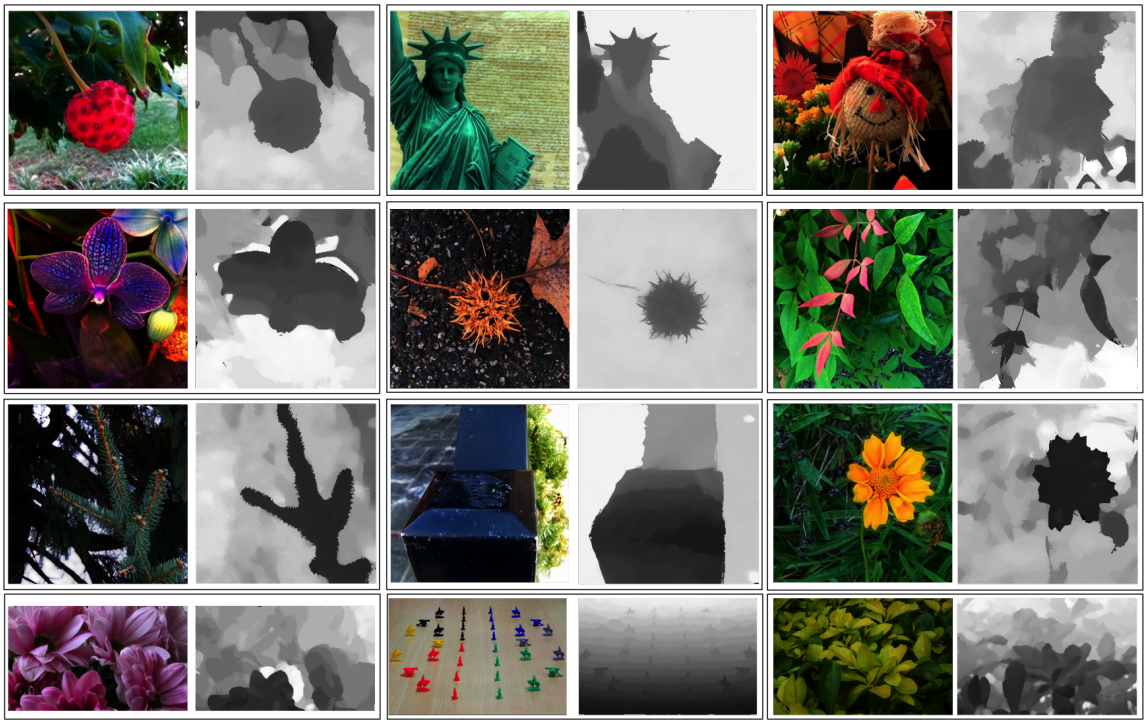

Figure 6.3 All-in-focus image and computed depth maps for different focal stacks from [69] and focal stacks that we captured. The first three rows show 9 focal stacks from [69] with different focal resolutions, indoor/outdoor scenes and varying levels of scene texture. The last row consists of three focal stacks that we captured using Canon EOS 1100D, 70D and 350D from left to right. These focal stacks had high focal resolution and degree of blur. Our composite focus measure and DfF pipeline clearly produces good depth reconstruction for various scene types.

6.1.1.2 Cost Volume Propagation

We use the Cost Volume Filtering technique [103] to propagate confident depth labels to other pixels. We build a high resolution cost-volume of 256 volumetric indices, each representing a depth value. The cost of a pixel for every label is assigned based on D_i and D_c. High confidence pixels are assigned zero cost to the label corresponding to their depth value D_i, and linearly increasing costs for other labels. All other pixels are assigned zero costs for all labels.

$$C_i(p) = \left\{ \begin{array}{cc} |D_i(p) - i| & \text{if } D_c(p) > t \\ 0 & \text{otherwise} \end{array} \right\}. \tag{6.2}$$

Here, $C_i(p)$ is the cost of assigning the label i to pixel p, with i indicating the 256 depth labels of the cost volume, D_i the initial depth and D_c the confidence from Laplacian regression, and $t(= 0.85)$ is the empirically computed confidence threshold.

A guided filtering operation over the cost volume generates the labeling for each pixel [40]. Guided image filtering enforces neighbourhood consistency along depth boundaries based on the intensity changes in a guidance image. We generate an all-in-focus image as the guidance image using a multi-label MRF over the composite focus measure. The data term and smoothness costs are assigned similar to Eqn. 3.1, with the sum of the composite focus measure providing the data cost for each pixel.

After filtering the cost volume C_i using the guidance image, a smooth depth map can be computed from the filtered cost volume C'_i in a winner-takes-all manner:

$$\mathcal{D}(p) = \arg\min_i C'_i(p) \tag{6.3}$$

Figure 6.2 shows the depth map generated using the guidance image and cost volume propagation.

6.1.2 Experiments and Results

We demonstrate results on real world focal stacks that we captured as well as other focal stacks used earlier [12, 69, 124]. Our data corpus for computing the cFM consists of several focal stacks with varying scene characteristics such as depth range, degree of blur, number of focal slices, textures in the scene, indoor/outdoor illumination etc. We use focal stacks from the light-field saliency dataset [69] representing everyday scenes and having focal resolution from 3 slices to 12. We also use other focal stacks with high degrees of defocus blur captured ourselves using DSLR cameras such as Canon 70D, 350D, 1100D as well as mobile devices such as the Nexus 5X. These vary in focal resolution from 5 to 40 slices. We also use focal stacks provided by previous researchers [12, 124]. On the Canon DSLR cameras, we used MagicLantern [75] to capture focal stacks and for the Nexus 5X we implemented a custom focal stack capture application using the Android Camera2 API.

We use standard parameter values as defined in [12, 98, 123] for any focus measures that require additional parameters. The offline process of computing the cFM is a compute intensive process as discussed in previous chapters. In this step, all FMs are evaluated for three different support-window

resolutions of 3×3, 7×7 and 11×11 and then averaged, to assemble a cumulative response across multiple regions of support. We reuse computed numerical values whenever possible, as multiple measures from the same family start with similar numerical computations. All our modules are implemented in Matlab except for the MRF module which is in C++. Once the cFM is computed, the computational complexity of our method is moderate. At runtime, we apply all FMs from the composite focus measure at a window size of 3×3 because noisy estimates are acceptable as they average out across the cFM but dilation due to larger window sizes results in more serious depth errors. Applying the composite focus measure, laplacian regression and depth propagation together takes about 60 seconds on a focal stack of 1K$\times$1K images on a standard desktop computer. We have also built Android and iOS applications which can capture few-sliced focal stacks and generate depth maps using our approach.

We show qualitative and quantitative results to evaluate our method. We compare the effectiveness of our composite focus measure against individual top focus measures defined in [12, 98, 123], using the same two stage DfF pipeline. We perform quantitative evaluation of our depth-map using a few light-field datasets from [135] and also use an evaluation strategy similar to [124]. We provide qualitative comparison with state-of-the-art techniques such as [12, 124] and also demonstrate good quality depth reconstruction on new focal stacks.

6.1.2.1 Quantitative evaluation

Figure 3.7 gives quantitative depth reconstruction results for the dataset from [135]. We synthesize 25 focal slices from synthetic light fields (buddha and medieval) and use these focal slices to compute depth using our pipeline. We build a high resolution (256 depths) depth map from just 25 slices, and the depth reconstruction is compared to the available ground truth depth in PSNR (comparing the estimated depth to 8-bit ground-truth grayscale depth). The results show a clear benefit of using our composite focus measure as opposed to other single focus measures. Our composite focus measure also performs much better than the top five (MCFS-5) measures selected from unsupervised feature selection [16].

Figure 6.4 gives depth computed by our method on two focal stacks. The first is from [124] and the other one is captured by us using a Canon 1100D. In both stacks, the focus ring movement between consequent slices is fixed and thus the depth change between them is quantized. Following [124], known depth values for two objects in the scene are used to compute the depths of the third object. Table 6.1 gives quantitative comparison of our method with [124]. On our Cards focal stack, we get an RMSE of 0.59 inches for the depth of the cards in the background which are at a depth of more than 30 inches from the camera. Lower error in depth-computation suggests that our method estimates depth maps at a higher quality.

6.1.2.2 Qualitative results

We demonstrate our results on standard datasets with qualitative comparison to other DfF methods in Figure 6.5. It can be seen that the detail in the depth map for the fruits dataset and the plants dataset is higher in our results, especially in the regions at low depth values. In the watch dataset, a much

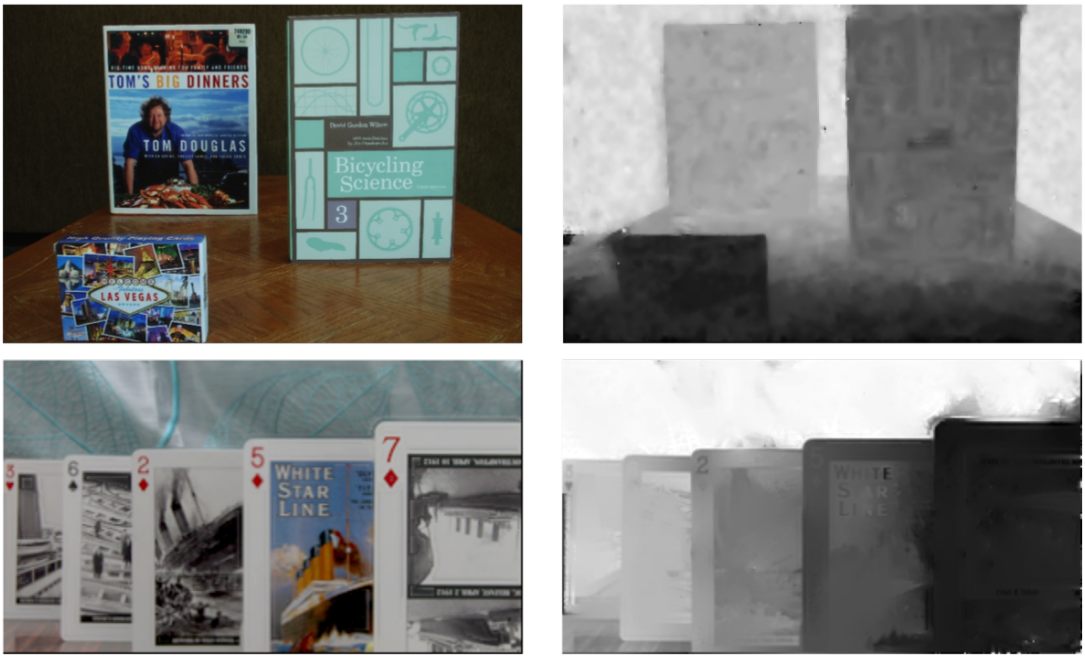

Figure 6.4 Focal stacks and computed depth-maps for the quantitative comparison of our approach with that of [124].

Known Depths	Estimated Depth	Ground Truth
$\hat{d}_{box}, \hat{d}_{bike}$	$\hat{d}_{cook} = 27.61$ inches	28 inches
$\hat{d}_{box}, \hat{d}_{cook}$	$\hat{d}_{bike} = 18.64$ inches	18.5 inches
$\hat{d}_{bike}, \hat{d}_{cook}$	$\hat{d}_{box} = 11.83$ inches	12 inches

Table 6.1 Computed depths for the *Books* focal stack using our method. We observe an average RMSE of 0.45 inches compared to an average RMSE of 2.66 inches reported in [124].

smoother variation from near to far can be observed in our results and in the flower dataset, the depth variation in the petals is clearly visible.

Figure 3.8 shows qualitative performance of our composite focus measure compared to the top individual focus measures from [98, 123] and also over FMs selected using [16]. We also provide depth-maps for focal stacks that we captured and focal stacks that were a part of [69] in Figure 6.3. The focal stacks shown in Figure 6.3 have varying degrees of defocus, number of focal slices, depth range, indoor/outdoor illumination conditions etc. The quality of the computed depth-maps indicates that our composite focus measure is robust and provides high quality depth reconstruction.

6.1.2.3 Limitations

Our DfF approach is limited to static scenes. Capturing focal stacks of dynamic scenes would require special cameras which can shoot multiple focus distances simultaneously. The assumption that each pixel has a single focus peak can fail if a focal stack ranges from macro to distant objects. Extreme

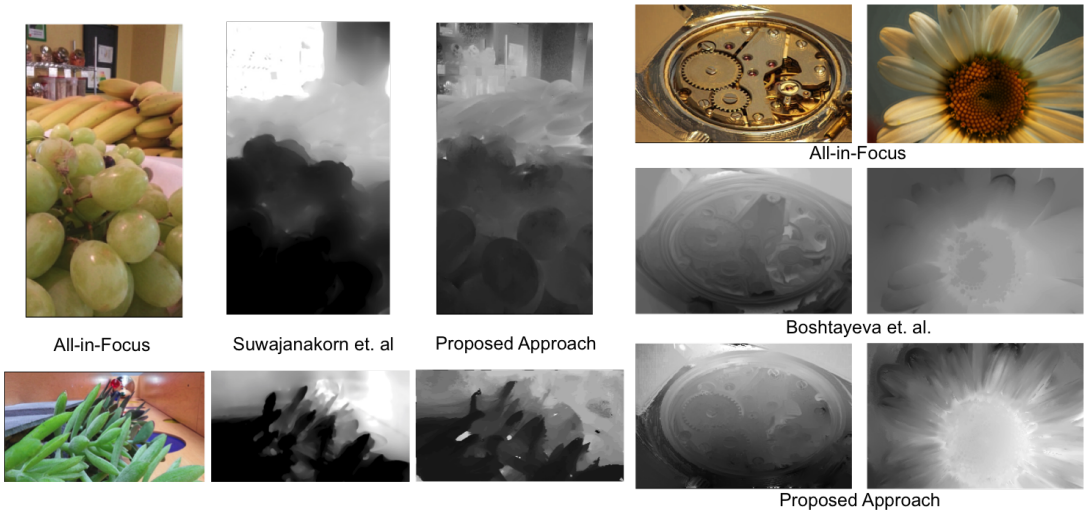

Figure 6.5 Comparison of our approach with that of Suwajanakorn et al. [124] and Boshtayeva et al. [12]. The comparison with [124] is shown on the left hand side and with [12] is shown on the right hand side. Our depth maps show improved resolution and smoothness, and the underlying image structure is more precisely retained in the depth image.

defocus in the foreground can result in previously occluded background pixels appearing sharp, giving two focus peak candidates for some pixel locations. The response of any FM is unreliable at such pixels.

6.2 Dense View Interpolation

We now present a dense view interpolation method for mobile devices in which a few images of the scene are captured in the form of focal stacks, and novel views of the scene are generated by interpolating the focal textures generated from these focal stacks. We use a model in which the scene is assumed to consist of a set of focal planes. From the captured focal stack at each viewpoint, an all-in-focus representation of the view is generated by merging the focused regions of each focal slice into one image. The focused regions are then shifted by automatically estimated disparities to generate novel views from novel viewpoints.

Synthesis of novel views for a given scene relies on either geometric estimation of the scene elements or image-based rendering [32, 46]. While geometric methods rely on accurate estimation of the depth of scene elements, image based methods generate novel views by interpolating a set of captured images. Light field rendering is an image based method which uses dense sampling of light rays for novel view synthesis. For aliasing free light field rendering, a very high sampling density of camera locations is necessary, as described by Chai et al. [17].

In order to achieve a dense sampling of scene light rays, it is useful to capture images at sparse camera locations and interpolate a dense set of intermediate images between them. We use an iterative method to estimate the focal textures at each camera location and interpolate these focal textures based

on appropriate disparity estimates to generate the intermediate images. Dense view interpolation for a scene consisting of two manually-identified depth layers was presented by Kubota et al. [54]. Levin et al. [66] enhanced this idea by capturing a focal stack of the scene and generating novel views of the scene from different viewpoints within the aperture area by geometrically shifting each focal slice with its appropriate disparity.

In our case, we model the scene as consisting of multiple focal regions, but do not need to capture an exhaustive focal stack. We also do not need to identify the focal regions manually. We use a mobile device to capture an adaptive focal stack of the scene from adjacent camera positions and adaptively estimate the disparity and blur parameters required for accurate view interpolation between the camera positions. We use mobile devices as they are highly accessible and they tightly integrate optical lenses, electronic sensors and versatile computing. Such a combination is ideal for computational photography which often requires adaptive optical sensing and embedded computing [2].

6.2.1 Imaging and Interpolation Model

We assume that the scene consists of objects at n different depth-of-field regions and we capture focal stacks from several adjacent camera locations. The imaging model treats each captured image as a combination of ideal *focal textures* located at n focal regions of the scene.

A focal stack g^i is a set of differently focused images captured using a camera with a finite aperture. The scene consists of n focal textures f^i_j at each of the n focal planes captured in the focal stack. These focal textures at a given camera location C^i can be understood as

$$f^i_j(x,y) = \left\{ \begin{array}{cc} f^i(x,y) & \text{if } z^i(x,y) = j \\ 0 & \text{otherwise} \end{array} \right\} \tag{6.4}$$

where f^i_j is the j^{th} focal texture, f^i the ideal all-in-focus image at the camera location C^i, and $z^i(x,y) \in [1,n]$ is the index of the focal slice at which the pixel (x,y) is maximally in focus. The true f^i_j textures are unknown. The focal stack g^i can be modeled as a sum of these textures convolved with appropriate blur kernels as

$$g^i_k = \sum_{j=1}^{n} h_{kj} * f^i_j, \quad k \in [1,n] \ , \tag{6.5}$$

where h_{kj} represents the blur kernel (usually a Gaussian of blur radius R_{kj}) between the focal planes k and j with h_{kk} being a delta function. Equation 6.5 is an extension of [54] and approximates the focal stack using the focal textures and blur kernels. This linear imaging model may not be correct at depth discontinuities, but does not produce significant interpolation artifacts. With known blur kernels, Equation 6.5 consists of n equations in n unknown f^i_j textures.

Once the f^i_j textures are estimated, the all-in-focus image f^i can be evaluated as

$$f^i = f^i_1 + f^i_2 + f^i_3 + \ldots + f^i_n \ . \tag{6.6}$$

Our objective is to compute the all-in-focus image at an arbitrary intermediate location, between two focal stacks from locations C^i and C^{i+1}. We parameterize the location using the fraction α of the distance between them, with a value of 0 at C^i and a value of 1 at C^{i+1}.

The focal textures f_j^i and f_j^{i+1} can be interpolated to generate the intermediate image as shown in Figure 6.6. The f_j^i textures are shifted forwards by αd_j and the f_j^{i+1} textures are shifted backwards by $-(1-\alpha)d_j$, where d_j is the disparity of the j^{th} focal texture between camera locations C^i and C^{i+1}. The disparity d_j can be either horizontal or vertical based on the change in position from C^i to C^{i+1}. The interpolated forward and backward all-in-focus images for a horizontal shift can be generated similar to Equation 6.6 by first shifting the f_j^i textures and then adding them to generate the all-in-focus intermediate image:

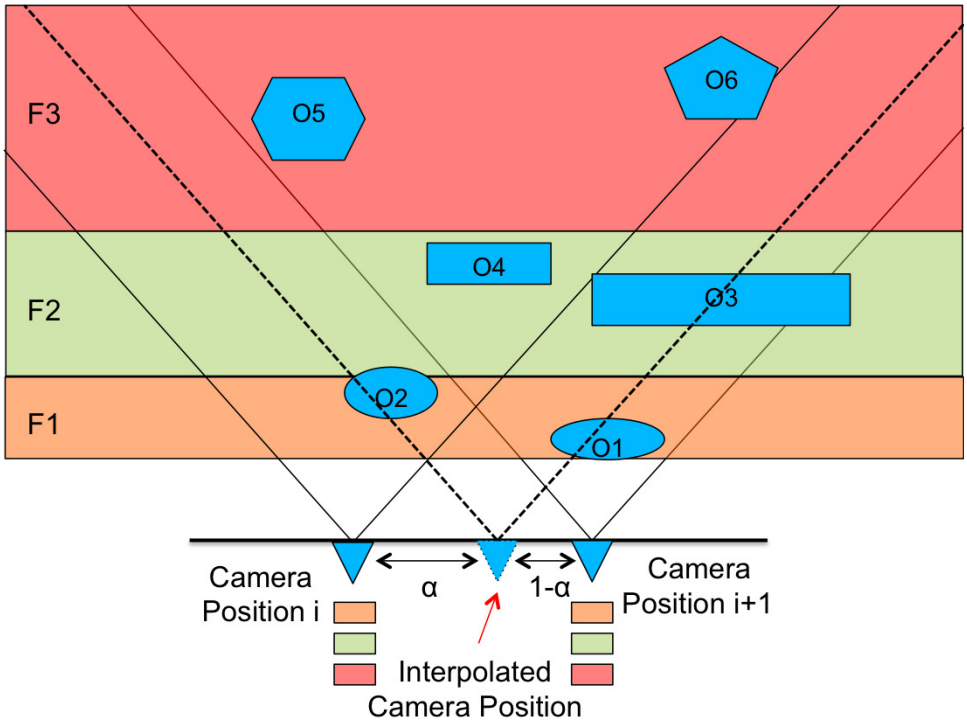

Figure 6.6 All-in-focus image at α is interpolated from images at C^i and C^{i+1}. Scene objects denoted by O_1 to O_6 and focal zones denoted by F_1 to F_3. The focal slices are refined from the captured focal stack such that $F_1 \cup F_2 \cup \ldots \cup F_n$ covers the entire desired range and $F_1 \cap F_2 \cap \ldots \cap F_n$ is negligible.

$$f'(x,y;\alpha) = \sum_{j=1}^{n} f_j^i(x - \alpha d_j, y) \tag{6.7}$$

$$f''(x,y;\alpha) = \sum_{j=1}^{n} f_j^{i+1}(x - (\alpha-1) d_j, y) \tag{6.8}$$

A similar expression can be derived for shifts between vertically separated camera positions.

The all-in-focus image f_α at the intermediate location α can be approximated using a weighted average of f' and f'' [54] as

$$f_\alpha(x, y) = (1 - \alpha)f'(x, y; \alpha) + \alpha f''(x, y; \alpha). \tag{6.9}$$

6.2.2 Capturing the Focal Stack

The focal stack g^i is a set of differently focused images captured at camera location C^i. Ideally, a focal stack is captured by controlled motion of the camera lens relative to the camera sensor by predefined distances. The distance of lens motion in each step is decided such that a minimal number of images are captured while ensuring that the entire depth variation of the visible scene is covered, as described in [145]. It is possible to implement such a capture mechanism on certain DSLR cameras which enable live control of focal length and aperture settings, and on certain Linux based mobile phones which allow for programmatically controlling the lens focus distance [131]. This control is however not available on a large number of mobile devices and cameras in which there is no explicit programmatic control of the lens focus distance like most Android based smartphones. The Android SDK provides a function call to only retrieve the focus distance of the camera but the function returns unreliable and incorrect results on several devices.

We present a simple method to capture a focal stack on devices that do not possess explicit focus distance control. Most cameras and smartphones are equipped with auto-focus and touch based manual-focus features which can fix the focus of the camera based on the user's desired focus location. We make use of this focus control mechanism in order to capture a set of images with different focus settings and emulate a focal stack from these captured images.

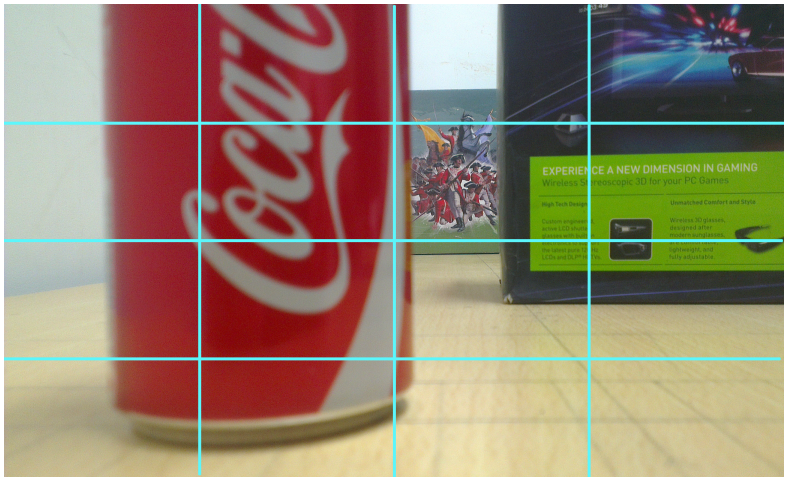

Figure 6.7 The focal stack is captured using 16 focus regions.

In order to capture images focused at different locations in the scene, we divide the visible scene into sixteen different rectangular regions as shown in Figure 6.7. Images are captured by sequentially setting the focus area to each of these regions. After capturing these images, the image slices that might

be redundant (due to different rectangular focus regions mapping to the same physical focus distance) are eliminated by measuring absolute image differences between the captured slices. As the redundant images are similar in both the focused and defocused regions, analyzing pixel differences works well even though there might be small misalignment between the captured images due to camera shake. The focal stack is thus emulated as a set of a few images having different focus distances, without any explicit order between the images and without explicit knowledge of the focus distance itself. Thus, interpolation of the focal textures based on geometric disparities similar to [66] is not directly possible. However, with explicitly estimated disparities for each focal slice between adjacent camera locations, it becomes possible to interpolate intermediate views.

6.2.3 Focal Textures and View Interpolation

To handle scenes with unknown number of focal regions, we need to estimate the number of focal regions in the scene. We assume that the number of images remaining after eliminating redundant focal slices corresponds to the different visible focal regions in the scene. For view interpolation, we need to estimate the textures f_j^i, the disparities d_j, and the blur radii R_{kj} from the available focal stack g^i.

Figure 6.8 The focal stack at a certain C^i consisting of three focal slices, along with the sharpness index map built over the stack.

We capture a focal stack g^i at different camera locations C^i such that there is a small horizontal or vertical separation between any two adjacent camera locations. Each focal stack g^i at C^i is explicitly registered to the first focal slice g_1^i. This is done in order to eliminate magnification errors due to focus/defocus and to account for camera motion during capture. The time taken for the described focal stack capture mechanism is relatively large as many images are being captured with a change in focal settings after each capture. This can possibly lead to misalignment in the focal slices. In order to register the focal stack, we use the Enhanced Correlation Coefficient [24] algorithm, which

uses a modified correlation coefficient model to accurately register images suffering from photometric distortions. Registering the focal stack ensures per-pixel alignment of the stack.

6.2.3.1 Sharpness Index Map

In order to estimate the blur and disparity parameters required for view interpolation, we make use of a sharpness index map built over the focal stack. In the index map, each pixel across the stack is labeled to the focal texture at which the weight w_k is maximized, where

$$w_k(x, y) = \nu(P_k(x, y, d)) \tag{6.10}$$

represents the variance of a patch of size d centered at the pixel in the g_k^i image [145]. Thus, the index map for any focal stack labels each pixel of the registered stack to the image where it appears maximally sharp (in-focus). Figure 6.8 shows the index map built from the focal stack. The index map so generated can now be used to automatically build rectangular focal templates required for further parameter estimation.

6.2.3.2 Estimating Disparities

We can identify the disparity d_j between adjacent camera locations for every focal plane j using the index map. We use template matching based on FFT correlation, using automated selection of templates. For each focal image g_k^i, we use the index map to locate a set of rectangles in which all pixels are labeled to the same focal region, similar to connected component analysis over pixels. We collect these rectangles in a list sorted by size. We select the largest located rectangle as the focal template and match it in the same focal image of the next camera position g_k^{i+1} to estimate the disparity d_k between the two camera locations for that layer. The focal templates extracted from the index map are shown in Figure 6.9.

The disparity estimation method may fail for uniform textured regions if pixels do not register correctly to their true focal region. Also, if the template is large, then it may be partially occluded in the adjacent image because of change in camera position. To solve this, we discard template matches with a high vertical disparity when the camera movement between C^i and C^{i+1} is horizontal and vice versa, or if the disparity for a focal region is abnormally large or small compared to neighboring focal disparities, and use the next largest rectangular template.

6.2.3.3 Blur kernel estimation

The blur kernels h_{kj} can also be estimated automatically once we have extracted focal templates for each focal region. They are estimated by sequentially blurring the template with incremental blur radii until the best match is found. The blur radii are estimated for every R_{kj} pair but we assume that the defocus blur between the focal layers k and j would be constant for the two focal slices i.e. $R_{kj} = R_{jk}$. Thus for a scene consisting of three focal regions, three blur parameters of R_{12}, R_{13} and

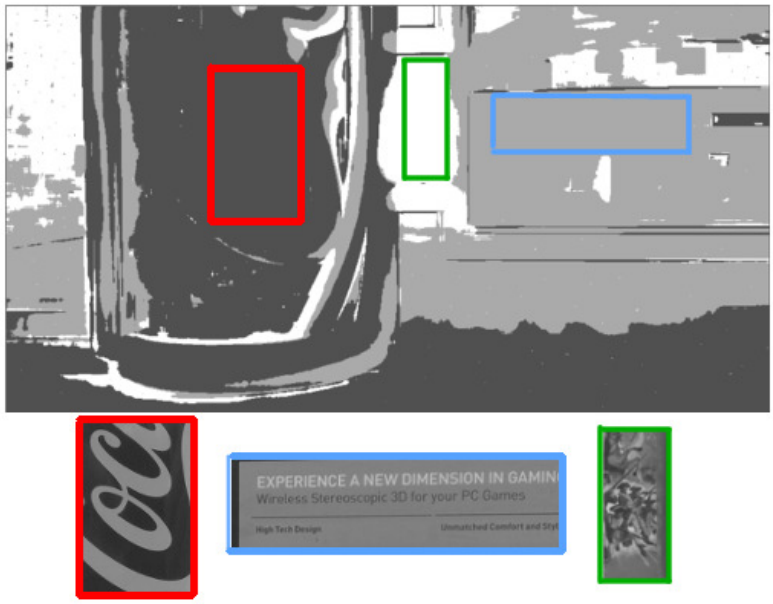

Figure 6.9 The focal templates for the three focal regions extracted from the sharpness index map by generating a list of uniform index rectangles sorted by size.

R_{23} are estimated. The defocus blur over the chosen templates is shown in Figure 6.10. Small errors in blur estimation have been empirically shown to cause little or no interpolation artifacts [54].

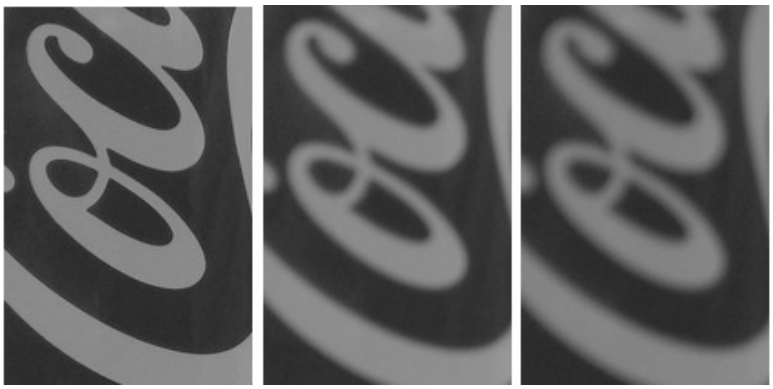

Figure 6.10 The first focal template extracted from the f_1^i, f_2^i and f_3^i textures. The defocus blur between focal regions is estimated using these templates.

6.2.3.4 Solving for Focal Textures

Once the focal stack g^i is aligned due to explicit registration and the d_j and h_{kj} parameters are estimated, we solve Equation 6.5 in the frequency domain to estimate the f_j^i textures as follows

$$G_k^i = \sum_{j=1}^{n} H_{kj} F_j^i, \quad k \in [1, n] \tag{6.11}$$

F, G and H are the Fourier transforms of the f textures, the images g and the kernels h respectively, with $H_{ii} = 1$. Equation 6.11 gives a system of n equations in n unknowns, where each variable is a matrix in the frequency domain. In [54], the acquired equation is a system of two equations in two variables and interpolated views are generated using linear filtering of the input images without explicitly identifying the F images, but filter corrections are applied to eliminate artifacts caused by H_{kj} filter divergence at DC.

We require a general solution for Equation 6.11 with n focal layers. We iteratively solve for the F textures in the frequency domain. We use an iterative method similar to [55] but solve for the focal textures in the frequency domain as expensive convolution operations in each linear equation are replaced by per-element matrix multiplication. After k iterations:

$$\begin{aligned}
F_1^{i,k} &= G_1^i - [H_{12} F_2^{i,k-1} + H_{13} F_3^{i,k-1} \\
&\qquad + \ldots + H_{1n} F_n^{i,k-1}] \\
F_2^{i,k} &= G_2^i - [H_{21} F_1^{i,k} + H_{23} F_3^{i,k-1} \\
&\qquad + \ldots + H_{2n} F_n^{i(k-1)}] \\
&\qquad \ldots \\
F_n^{i,k} &= G_n^i - [H_{n1} F_1^{i,k} + H_{n2} F_2^{i,k} \\
&\qquad + \ldots + H_{n(n-1)} F_{n-1}^{i,k}]
\end{aligned} \tag{6.12}$$

where $F_j^{i,k}$ represents the F_j^i texture after the k^{th} iteration. We apply the Gauss-Seidel iterative method to a system of linear equations in non-square matrices in the frequency domain [29]. Such a solution assumes a prior on the F textures. We use the sharpness index map to segment each g_j^i image into a prior f_j^i texture. Thus, all the pixels labeled to certain index j are picked from the focal stack image g_j^i to generate $f_j^{i,0}$. This estimates all the focal textures as a collection of best focused pixels at that focal plane. We solve Equation 6.12 using these priors and compute the inverse Fourier transform of the F matrices to estimate the f_j^i textures at each camera location. This solution of f_j^i textures does not require any explicit control over the blur kernels as in [54] and is scalable to many focal layers. The overall procedure involving the estimation of focal textures and interpolation is described in the following algorithm

Algorithm 3 PROCEDURE

1: Capture an approximate focal stack at each C^i. Evaluate number of visible focal layers n.
2: Register each image in g^i to g_1^i.
3: Estimate the sharpness index map over the focal stack.
4: Estimate the disparity d_j between adjacent stacks for each layer j.
5: Estimate blur radii R_{kj}.
6: Solve Equation 6.5 for f_j^i.
7: Interpolate to get the α images.

6.2.3.5 View Interpolation

Once the f_j^i textures are computed, we can interpolate novel α views between any two camera locations using equations 6.7 and 6.8. The all-in-focus image can be estimated using Equation 6.6. As an example consider four clockwise camera locations forming a square C^{1-4}. An intermediate image at α and β (horizontal and vertical shifts) can be estimated by first interpolating the (f_j^1,f_j^2) and (f_j^3,f_j^4) along the horizontal direction using equations 6.7,6.8 and then correspondingly along the vertical direction. Thus the interpolation is composed of three sets of focal texture shifts and summation. The overall interpolation can be written as:

$$\begin{aligned} f_{\alpha\beta} =&(1-\alpha)(1-\beta)f^1 + \alpha(1-\beta)f^2 \\ &+ (1-\alpha)\beta f^3 \qquad + \alpha\beta f^4 \end{aligned} \tag{6.13}$$

where

$$f^i = \sum_{j=1}^{n} f_j^i(x - \alpha d_j^x, y - \beta d_j^y) \quad i \in [1,4] \tag{6.14}$$

The steps $1-6$ in the described Algorithm 3 can be computed as pre-processing steps in the background once the focal stacks are captured and the interpolation in step 7 requires lesser computational resources and can be applied actively based on the user's α, β input. Generating a set of dense interpolated views of the scene can also enable novel view generation using light field rendering from novel viewpoints that are not explicitly on the camera plane.

6.2.4 Experiments

We experiment with an HTC One X mobile device which is equipped with a quad-core 1.5GHz Nvidia Tegra 3 processor having an inbuilt ULP GeForce GPU. We use the OpenCV for Tegra 2.4.5 library with the Tegra GPU enabled for all OpenCV primitive image processing and core function calls. This device uses an 8MP camera which has a maximum aperture of f/2. The camera can capture 5 shots per second in burst mode without any intermediate auto-focus calls. For capturing the focal stack however, there is a need to adjust the focus after each image and it thus takes about 0.8 to 0.9 seconds to capture an image. Our focal stack is a set of 16 images where each 1280x760 image is focused on a different rectangular region of the visible scene. We find a 4x4 block to be the ideal size for the focus

Figure 6.11 Horizontally interpolated all-in-focus images at $\alpha = 0, 0.2, 0.4, 0.6, 0.8, 1.0$. Expanded views of the images are shown below each image. Different focal textures being shifted by their appropriate disparity is visible.

regions as it captures most of the elements in the scene and also has a reasonable capture time. 3x3 or 5x5 blocks would lead to objects being missed from focus and large capture times respectively. For two slices i and j, j is classified as redundant if the sum of absolute pixel differences of i and j is less than 10% of the maximum pixel difference across the stack for i. Pixel differences are estimated at half the image resolution for computational efficiency. The redundant image removal can also be pipelined with camera focus adjustment for each shot during capture.

We capture a focal stack at four camera locations lying on the edges of a $1.5cm$ square at an overall capture time of about one minute. The focal stacks are then registered and the pixel difference approach is reapplied to eliminate any redundant image that might not have been identified without per-pixel alignment. This step helps in case of significant camera shake during capture. The index map is built at one camera location and the focal templates for each focal texture are isolated.

The disparity and blur parameters are estimated using the focal templates. Since the relative distance between the camera positions is low, we can reuse the same templates to identify disparities between all camera locations. Also, the blur kernels are dependent on the distance of the scene objects from the camera and thus for planar movement of the camera along the square, the blur kernels are constant at all four camera locations. Estimating the index map, the blur radii and the focal templates is thus done once and takes about 2 minutes. Image registration, disparity estimation and focal texture estimation takes about $10, 15$ and 40 seconds respectively per camera location. The focal textures at each camera location are estimated by processing each color channel independently using Equation 6.12. The overall capture and processing time is about 2.5 to 3 minutes per camera location.

The interpolation of novel views is performed in near-real-time based on user input by shifting the focal textures by the appropriate disparities and merging them to generate the intermediate view. The processing involving the generation of focal textures can be fully computed on the device. However, considering the time taken for processing and the fact that the battery usage for a mobile device should be as low as possible, this approach can also be extended to a cloud based framework, where the device captures, registers and removes the redundancies from the focal stack, and the resulting images are uploaded to a cloud service which processes and feeds the focal textures back to the device for real-time view interpolation. The results of all-in-focus view generation and interpolation are shown in Figure 6.11. We have presented a comprehensive framework for dense view interpolation through these results.

Chapter 7

Conclusions & Future Work

In this thesis, we provide a comprehensive study of focus, defocus blur and depth-of-field and present our contribution to several areas in epsilon focus photography. We propose a composite measure of focus that is a cumulative estimate of several top performing measures of focus for the task of depth-from-focus. We find that this measure is more robust and more general than any single measure for estimating the amount of focus at a pixel in an image. Using our composite focus measure, we propose a compact yet detailed representation of focus, that separately encodes the focused and defocused segments in an image as a part of a focal stack of images. Our representation is capable of encoding the fine characteristics of wide-aperture images such as background objects becoming visible through disappearing foreground segments, saturated bokeh circles and complex pixel interactions at depth edges. Our representation of focus can faithfully reconstruct any wide-aperture image of the scene and can thus find use in photo-editing toolkits. We show that an accurate model for focus can lead to precise scene depth measurements using our composite focus measure. We demonstrate that quick focus measurements in a scene can enable useful image-based applications such as view interpolation in the scene. Extending the capability of focus control to single images, we build data-driven models for focus estimation and manipulation that can change focus and defocus blur from a single image input. Such models are becoming more and more powerful in terms of their representational complexity and we show how adversarial learning strategies can faithfully encode information about whether a pixel is in focus or not and how to photo-realistically change the amount of focus at a pixel.

The work presented in this thesis encompasses some of the first efforts in computational photography for comprehensive estimation and manipulation of focus and defocus blur. The fine characteristics of focus and the learning based approach to focus manipulation is one of the first works in these directions. With the advent of dual-pixel cameras and wider apertures on portable cameras in general, applications such as focus manipulation and portraiture have gained a lot of interest in recent days. Depth and defocus map estimation are solved in a coupled manner in papers such as Lee et al. [61] and Hazirbas et al. [39]. These works extend the scope of scene structure understanding using focus as a cue. Furthermore, the fine characteristics of focus and careful focus manipulation is studied in works such as Hermann et al. [43]. The work presented in these papers demonstrates that portraiture from a single-image is now widely accessible and is becoming a well studied problem in computational photography.

In the future, an optimal representation of focused and defocused pixels, which may be a combination of geometric and data driven primitives can serve as the right platform for post-capture scene refocusing. It may also be meaningful to encode focus estimation and control at lower levels in the information processing pipeline of the ISP, thereby making the process more amenable to handheld cameras. Identifying the correct depth-of-field for a desired image is often guided by the size of the aperture which in-turn has an impact other capture time parameters such as exposure time and the required sensor sensitivity. The task of capturing the desired depth-of-field can be approached as a simultaneous optimization of all such parameters and compact representations and solutions similar to those proposed in this thesis can aid in solving such problems.

Related Publications

"Geometric Scene Refocusing", Parikshit Sakurikar & P.J. Narayanan. *Arxiv - December 2020.*

"Defocus Magnification using Conditional Adversarial Networks", Parikshit Sakurikar, Ishit Mehta & P.J. Narayanan. *IEEE Winter Conference on Applications of Computer Vision (WACV) 2019*, Hawaii, USA.

"RefocusGAN: Scene Refocusing using a Single Image", Parikshit Sakurikar, Ishit Mehta, Vineeth N. Balasubramanian & P.J. Narayanan. *European Conference on Computer Vision (ECCV) 2018*, Munich, Germany.

"Composite Focus Measure for High Quality Depth Maps", Parikshit Sakurikar & P.J. Narayanan. *IEEE International Conference on Computer Vision (ICCV) 2017*, Venice, Italy.

"Focal Stack Representation and Focus Manipulation", Parikshit Sakurikar & P.J. Narayanan. *Asian Conference on Pattern Recognition (ACPR) 2017*, Nanjing, China.

"Dense View Interpolation on Mobile Devices using Focal Stacks", Parikshit Sakurikar & P.J. Narayanan. *CVPR International Workshop on Mobile Vision (IWMV) 2014*, Columbus, Ohio, USA.

Other Publications

"Appearance Editing with Free-viewpoint Neural Rendering", Pulkit Gera, Aakash K.T., Dhawal Sirikonda, Parikshit Sakurikar & P.J. Narayanan. *Arxiv - October 2021*.

"Fast Analytic Soft Shadows from Area Lights", Aakash K.T., Parikshit Sakurikar & P.J. Narayanan. *Eurographics Symposium on Rendering (EGSR) 2021*, Saarbrücken, Germany.

"A Flexible Neural Renderer for Material Visualization", Aakash K.T., Parikshit Sakurikar, Saurabh Saini & P.J. Narayanan. *ACM SIGGRAPH Asia Technical Briefs 2019*, Brisbane, Australia.

"The Astropy project: building an open-science project and status of the v2. 0 core package", A.M Price-Whelan and the Astropy Contributors, *The Astronomical Journal, 2018*.

"SLFT: A Physically Accurate Framework for Tracing Synthetic Light Fields", Udyan Khurana, Parikshit Sakurikar & P.J. Narayanan. *Indian Conference on Vision, Graphics and Image Processing (ICVGIP) 2018*, Hyderabad, India.

"Structured Adversarial Training for Unsupervised Monocular Depth Estimation", Ishit Mehta, Parikshit Sakurikar & P.J. Narayanan. *International Conference on 3D Vision (3DV) 2018*, Verona, Italy.

"Beyond OCRs for Document Blur Estimation", Pranjal Rai, Sajal Maheshwari, Ishit Mehta, Parikshit Sakurikar & Vineet Gandhi. *IAPR International Conference on Document Analysis and Recognition (ICDAR) 2017*, Kyoto, Japan.

"SynCam: Capturing sub-frame synchronous media using smartphones", Ishit Mehta, Parikshit Sakurikar, Rajvi Shah & P.J. Narayanan. *IEEE International Conference on Multimedia and Expo (ICME) 2017*, Hong Kong.

"Intrinsic Image Decomposition using Focal Stacks", Saurabh Saini, Parikshit Sakurikar & P.J. Narayanan. *Indian Conference on Vision, Graphics and Image Processing (ICVGIP) 2016*, Guwahati, India.

"Comparison Sorting on Hybrid Multicore Architectures for Fixed and Variable Length Keys", Dip Sankar Banerjee, Parikshit Sakurikar & Kishore Kothapalli. *International Journal of High Performance Computing Applications (IJHPCA) 2014*.

"CPU and/or GPU: Revisiting the GPU vs. CPU myth", Kishore Kothapalli et al. . *Arxiv - March 2013*.

"Fast, Scalable Parallel Comparison Sort on Hybrid Multicore Architectures", Dip Sankar Banerjee, Parikshit Sakurikar and Kishore Kothapalli, *Third International Workshop on Accelerators and Hybrid Exascale Systems (AsHES) 2013*, Boston, Massachusetts, USA.

"Increasing Intensity Resolution on a Single Display using Spatio-Temporal Mixing", Pawan Harish, Parikshit Sakurikar & P.J. Narayanan. *Indian Conference on Vision, Graphics and Image Processing (ICVGIP) 2012*, Mumbai, Maharashtra, India.

"Spatio-temporal Mixing to Increase Intensity Resolution on a Single Display", Pawan Harish, Parikshit Sakurikar & P. J. Narayanan. Poster at *CVPR Workshop on Computational Cameras and Displays (CCD) 2012*, Providence, Rhode Island, USA.

"Fast Graph Cuts using Shrink-Expand Reparameterization", Parikshit Sakurikar and P. J. Narayanan, *IEEE Winter Conference on Applications of Computer Vision (WACV) 2012*, Breckenridge, Colorado, USA.

Bibliography

[1] A. Abuolaim, A. Punnappurath, and M. S. Brown. Revisiting autofocus for smartphone cameras. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[2] A. Adams, E. Talvala, S. Park, D. Jacobs, B. Ajdin, N. Gelfand, J. Dolson, D. Vaquero, J. Baek, M. Tico, H. Lensch, W. Matusik, K. Pulli, M. Horowitz, and M. Levoy. The frankencamera: An experimental platform for computational photography. SIGGRAPH 2010, page 29. ACM.

[3] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *ACM Transactions on Graphics*, volume 23, pages 294–302. ACM, 2004.

[4] M. Aggarwal and N. Ahuja. Split aperture imaging for high dynamic range. In *Proceedings of the Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 10–17 vol.2.

[5] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 214–223, 2017.

[6] A. Ashok and M. A. Neifeld. Pseudorandom phase masks for superresolution imaging from subpixel shifting. *Appl. Opt.*, 46(12):2256–2268, Apr 2007.

[7] S. Bae and F. Durand. Defocus magnification. In *The Computer Graphics Forum*, volume 26-3, pages 571–579, 2007.

[8] S. W. Bailey, J. I. Echevarria, B. Bodenheimer, and D. Gutierrez. Fast depth from defocus from focal stacks. *The Visual Computer*, 31(12):1697–1708, 2015.

[9] S. W. Bailey, J. I. Echevarria, B. Bodenheimer, and D. Gutierrez. Fast depth from defocus from focal stacks. *The Visual Computer*, 31(12):1697–1708, 2015.

[10] J. T. Barron, A. Adams, Y. Shih, and C. Hernndez. Fast bilateral-space stereo for synthetic defocus. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4466–4474, 2015.

[11] S. S. Bhasin and S. Chaudhuri. Depth from defocus in presence of partial self occlusion. In *IEEE International Conference on Computer Vision*, volume 1, pages 488–493, 2001.

[12] M. Boshtayeva, D. Hafner, and J. Weickert. A focus fusion framework with anisotropic depth map smoothing. *Pattern Recognition*, 48(11):3310–3323, 2015.

[13] M. Boshtayeva, D. Hafner, and J. Weickert. A focus fusion framework with anisotropic depth map smoothing. *Pattern Recognition*, 48(11):3310 – 3323, 2015.

[14] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23, 2001.

[15] M. Brown. From raw to srgb and back: Modeling the onboard camera processing. https://www2.securecms.com/ICIP2013/Tutorial_T3.asp.

[16] D. Cai, C. Zhang, and X. He. Unsupervised feature selection for multi-cluster data. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2010.

[17] J. X. Chai, X. Tong, S. C. Chan, and H. Y. Shum. Plenoptic sampling. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 307–318, 2000.

[18] S. Chaudhuri and A. N. Rajagopalan. *Depth from defocus: a real aperture imaging approach*. Springer Science & Business Media, 2012.

[19] W. Chen, F. Kou, C. Wen, and Z. Li. Automatic synthetic background defocus for a single portrait image. *IEEE Transactions on Consumer Electronics*, 63(3):234–242, 2017.

[20] O. Cossairt, C. Zhou, and S. Nayar. Diffusion coded photography for extended depth of field. *ACM Trans. Graph.*, 29(4):31:1–31:10, July 2010.

[21] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[22] H. Du, X. Tong, X. Cao, and S. Lin. A prism-based system for multispectral video acquisition. In *2009 IEEE 12th International Conference on Computer Vision*, pages 175–182, Sept 2009.

[23] L. Du and Y.-D. Shen. Unsupervised feature selection with adaptive structure learning. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

[24] G. Evangelidis and E. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, 2008.

[25] F. Farhat, M. M. Kamani, S. Mishra, and J. Z. Wang. Intelligent portrait composition assistance: Integrating deep-learned models and photography idea retrieval. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, pages 17–25. ACM, 2017.

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

[27] G. Garg, E.-V. Talvala, M. Levoy, and H. P. Lensch. Symmetric photography: Exploiting data-sparseness in reflectance fields. In *Proceedings of the 17th Eurographics Conference on Rendering Techniques*, EGSR '06, pages 251–262, 2006.

[28] T. Georgeiv, K. C. Zheng, B. Curless, D. Salesin, S. Nayar, and C. Intwala. Spatio-angular resolution tradeoffs in integral photography. In *Proceedings of the 17th Eurographics Conference on Rendering Techniques*, EGSR '06, pages 263–272, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[29] G. H. Golub and C. F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, 1996.

[30] Y. Gong and S. Zhang. Ultrafast 3-d shape measurement with an off-the-shelf dlp projector. *Opt. Express*, 18(19):19743–19754, Sep 2010.

[31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems (NIPS)*, pages 2672–2680, 2014.

[32] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 43–54. ACM, 1996.

[33] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5767–5777, 2017.

[34] M. Gupta, Y. Tian, S. G. Narasimhan, and L. Zhang. A combined theory of defocused illumination and global light transport. *Int. J. Comput. Vision*, 98(2):146–167, June 2012.

[35] T. Hach, J. Steurer, A. Amruth, and A. Pappenheim. Cinematic bokeh rendering for real scenes. In *Proceedings of the 12th European Conference on Visual Media Production*, CVMP '15, pages 1:1–1:10, 2015.

[36] T. Hach, J. Steurer, A. Amruth, and A. Pappenheim. Cinematic bokeh rendering for real scenes. In *Proceedings of the 12th European Conference on Visual Media Production*, CVMP '15, pages 1:1–1:10, New York, NY, USA, 2015. ACM.

[37] S. W. Hasinoff and K. N. Kutulakos. Light-efficient photography. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2203–2214, 2011.

[38] S. W. Hasinoff, K. N. Kutulakos, F. Durand, and W. T. Freeman. Time-constrained photography. In *IEEE International Conference on Computer Vision*, pages 333–340. IEEE, 2009.

[39] C. Hazirbas, S. G. Soyer, M. C. Staab, L. Leal-Taix, and D. Cremers. Deep depth from focus, 2018.

[40] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[41] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[42] F. S. Helmli and S. Scherer. Adaptive shape from focus with an error estimation in light microscopy. In *ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis.*, pages 188–193, 2001.

[43] C. Herrmann, R. S. Bowen, N. Wadhwa, R. Garg, Q. He, J. T. Barron, and R. Zabih. Learning to autofocus, 2020.

[44] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.

[45] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *ACM Symposium on User Interface Software and Technology*, 2011.

[46] D. E. Jacobs, J. Baek, and M. Levoy. Focal stack compositing for depth of field control. *Stanford Computer Graphics Laboratory Technical Report*, 1, 2012.

[47] J. Johnson, A. Alahi, L. Fei-Fei, C. Li, Y. W. Li, and F. fei Li. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 2016.

[48] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics*, 35(6):193:1–193:10, Nov. 2016.

[49] E. Kee, S. Paris, S. Chen, and J. Wang. Modeling and removing spatially-varying optical blur. In *2011 IEEE International Conference on Computational Photography (ICCP)*, pages 1–8, 2011.

[50] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[51] K. Kodama and A. Kubota. Efficient reconstruction of all-in-focus images through shifted pinholes from multi-focus images for dense light field synthesis and rendering. *IEEE Transactions on Image Processing, 2013*, 2013.

[52] C. Kolb, D. Mitchell, and P. Hanrahan. A realistic camera model for computer graphics. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 317–324, 1995.

[53] A. Kubota, K. Aizawa, and T. Chen. Reconstructing dense light field from array of multifocus images for novel view synthesis. *IEEE Transactions on Image Processing*, 16(1):269–279, 2007.

[54] A. Kubota, K. Aizawa, and T. Chen. Reconstructing dense light field from array of multifocus images for novel view synthesis. *IEEE Transactions on Image Processing*, 16(1):269–279, 2007.

[55] A. Kubota, K. Takahashi, K. Aizawa, and T. Chen. All-focused light field rendering. In *Proceedings of the Fifteenth Eurographics conference on Rendering Techniques*, EGSR'04, pages 235–242, 2004.

[56] A. Kumar and N. Ahuja. A generative focus measure with application to omnifocus imaging. In *IEEE International Conference on Computational Photography*, pages 1–8. IEEE, 2013.

[57] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8183–8192, 2018.

[58] S. Kuthirummal, H. Nagahara, C. Zhou, and S. Nayar. Flexible depth of field photography. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):58–71, 2011.

[59] S. Kuthirummal and S. K. Nayar. Multiview radial catadioptric imaging for scene capture. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 916–923, New York, NY, USA, 2006. ACM.

[60] K. N. Kutulakos and S. W. Hasinoff. Focal stack photography: High-performance photography with a conventional camera. In *In IAPR Machine Vision Applications*, 2009.

[61] J. Lee, S. Lee, S. Cho, and S. Lee. Deep defocus map estimation using domain adaptation. In *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12214–12222, 2019.

[62] History of photographic lens design. https://en.wikipedia.org/wiki/Historyofphotographiclensdesign.

[63] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.*, 26(3), July 2007.

[64] A. Levin, S. W. Hasinoff, P. Green, F. Durand, and W. T. Freeman. 4d frequency analysis of computational cameras for depth of field extension. In *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, pages 97:1–97:14, New York, NY, USA, 2009. ACM.

[65] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(2):228–242, 2008.

[66] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242, 2008.

[67] A. Levin, P. Sand, T. S. Cho, F. Durand, and W. T. Freeman. Motion-invariant photography. In *ACM SIGGRAPH 2008 Papers*, SIGGRAPH '08, pages 71:1–71:9, New York, NY, USA, 2008. ACM.

[68] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision (ECCV)*, pages 702–716, 2016.

[69] N. Li, J. Ye, Y. Ji, H. Ling, and J. Yu. Saliency detection on light field. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2014.

[70] W. Lijun, S. Xiaohui, Z. Jianming, W. Oliver, H. Chih-Yao, K. Sarah, and L. Huchuan. Deeplens: Shallow depth of field from a single image. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):6:1–6:11, 2018.

[71] D. Liu, J. Gu, Y. Hitomi, M. Gupta, T. Mitsunaga, and S. K. Nayar. Efficient space-time sampling with pixel-wise coded exposure for high-speed imaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):248–260, Feb 2014.

[72] P. Llull, X. Liao, X. Yuan, J. Yang, D. Kittle, L. Carin, G. Sapiro, and D. J. Brady. Coded aperture compressive temporal imaging. *Opt. Express*, 21(9):10526–10545, May 2013.

[73] Lytro. https://www.lytro.com/.

[74] K. Ma, H. Fu, T. Liu, Z. Wang, and D. Tao. Deep blur mapping: Exploiting high-level semantics by deep neural networks. *IEEE Transactions on Image Processing (TIP*, 27-10:5155–5166, 2018.

[75] Magic lantern. http://magiclantern.fm/.

[76] M. T. Mahmood, A. Majid, and T.-S. Choi. Optimal depth estimation by combining focus measures using genetic programming. *Information Sciences*, 181(7):1249–1263, Apr. 2011.

[77] M. T. Mahmood, A. Majid, and T.-S. Choi. Optimal depth estimation by combining focus measures using genetic programming. *Information Sciences*, 181(7):1249–1263, Apr. 2011.

[78] S. Matsui, H. Nagahara, and R. I. Taniguchi. Half-sweep imaging for depth from defocus. In *Advances in Image and Video Technology*, pages 335–347. Springer, 2012.

[79] M. Möller, M. Benning, C.-B. Schönlieb, and D. Cremers. Variational depth from focus reconstruction. *IEEE Transactions on Image Processing*, 24:5369–5378, 2015.

[80] A. N. Murali Subbarao, Tae-Sun Choi. Focusing techniques. *Optical Engineering*, 32:32 – 32 – 13, 1993.

[81] H. Nagahara, S. Kuthirummal, C. Zhou, and S. K. Nayar. Flexible depth of field photography. In *European Conference on Computer Vision*, pages 60–73. 2008.

[82] H. Nagahara, S. Kuthirummal, C. Zhou, and S. K. Nayar. Flexible depth of field photography. In *Proceedings of the 10th European Conference on Computer Vision: Part IV*, ECCV '08, pages 60–73, Berlin, Heidelberg, 2008. Springer-Verlag.

[83] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 257–265, 2017.

[84] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, ICML, pages 807–814, 2010.

[85] V. P. Namboodiri and S. Chaudhuri. Recovery of relative depth from a single observation using an uncalibrated (real-aperture) camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6. IEEE, 2008.

[86] S. G. Narasimhan, S. J. Koppal, and S. Yamazaki. *Temporal Dithering of Illumination for Fast Active Vision*. 2008.

[87] S. G. Narasimhan and S. K. Nayar. Enhancing resolution along multiple imaging dimensions using assorted pixels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):518–530, April 2005.

[88] S. K. Nayar and V. Branzoi. Adaptive dynamic range imaging: optical control of pixel exposures over space and time. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1168–1175 vol.2, Oct 2003.

[89] S. K. Nayar and Y. Nakagawa. Shape from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):824–831, 1994.

[90] S. K. Nayar and Y. Nakagawa. Shape from focus. *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 16(8):824–831, 1994.

[91] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light Field Photography with a Hand-Held Plenoptic Camera. Technical report, Apr. 2005.

[92] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11), 2005.

[93] T. M. Nimisha, A. K. Singh, and A. N. Rajagopalan. Blur-invariant deep learning for blind-deblurring. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4762–4770, 2017.

[94] M. O'Toole, R. Raskar, and K. N. Kutulakos. Primal-dual coding to probe light transport. *ACM Trans. Graph.*, 31(4), July 2012.

[95] J. Park, Y.-W. Tai, D. Cho, and I. S. Kweon. A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2017.

[96] A. P. Pentland. A new sense for depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4), 1987.

[97] N. Persch, C. Schroers, S. Setzer, and J. Weickert. Introducing more physics into variational depth–from–defocus. In *German Conference on Pattern Recognition*, pages 15–27, 2014.

[98] S. Pertuz, D. Puig, and M. A. Garcia. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, 46(5):1415 – 1432, 2013.

[99] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph.*, 23(3):664–672, 2004.

[100] Pixel 2 cameras. https://research.googleblog.com/2017/10/portrait-mode-on-pixel-2-and-pixel-2-xl.html.

[101] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: Motion deblurring using fluttered shutter. *ACM Trans. Graph.*, 25(3):795–804, July 2006.

[102] R. Raskar, K.-H. Tan, R. Feris, J. Yu, and M. Turk. Non-photorealistic camera: Depth edge detection and stylized rendering using multi-flash imaging. SIGGRAPH '04, 2004.

[103] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[104] P. Sakurikar, I. Mehta, V. N. Balasubramanian, and P. J. Narayanan. Refocusgan: Scene refocusing using a single image. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[105] P. Sakurikar, I. Mehta, and P. J. Narayanan. Defocus magnification using conditional adversarial networks. In *Winter Conference on Applications of Computer Vision (WACV)*, January 2019.

[106] P. Sakurikar and P. J. Narayanan. Dense view interpolation on mobile devices using focal stacks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 138–143, June 2014.

[107] P. Sakurikar and P. J. Narayanan. Composite focus measure for high quality depth maps. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1623–1631, 2017.

[108] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, pages 1161–1168, 2005.

[109] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schlkopf. Learning to deblur. *Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 38(7):1439–1451, 2016.

[110] S. M. Seitz, Y. Matsushita, and K. N. Kutulakos. A theory of inverse light transport. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, 2005.

[111] P. Sen, B. Chen, G. Garg, S. R. Marschner, M. Horowitz, M. Levoy, and H. P. A. Lensch. Dual photography. *ACM Trans. Graph.*, 24(3), July 2005.

[112] C.-H. Shen and H. H. Chen. Robust focus measure for low-contrast images. In *2006 Digest of Technical Papers International Conference on Consumer Electronics*, pages 69–70, 2006.

[113] J. Shi, X. Tao, L. Xu, and J. Jia. Break ames room illusion: depth from general single images. *ACM Transactions on Graphics (TOG)*, 34(6):225, 2015.

[114] J. Shi, L. Xu, and J. Jia. Discriminative blur detection features. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2965–2972. IEEE, 2014.

[115] J. Shi, L. Xu, and J. Jia. Just noticeable defocus blur detection and estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 657–665, 2015.

[116] Y. Shih, B. Guenter, and N. Joshi. Image enhancement using calibrated lens simulations. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *European Conference on Computer Vision*, pages 42–56. Springer, 2012.

[117] P. P. Srinivasan, R. Garg, N. Wadhwa, R. Ng, and J. T. Barron. Aperture supervision for monocular depth estimation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[118] P. P. Srinivasan, T. Wang, A. Sreelal, R. Ramamoorthi, and R. Ng. Learning to synthesize a 4d RGBD light field from a single image. In *IEEE International Conference on Computer Vision, (ICCV)*, pages 2262–2270, 2017.

[119] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[120] Stanford light-field archive. http://lightfield.stanford.edu/.

[121] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang. Deep video deblurring for hand-held cameras. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 237–246, 2017.

[122] J. Sun, J. Sun, S. B. Kang, Z. B. Xu, X. Tang, and H. Y. Shum. Flash cut: Foreground extraction with flash and no-flash image pairs. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[123] J. Surh, H. G. Jeon, Y. Park, S. Im, H. Ha, and I. S. Kweon. Noise robust depth from focus using a ring difference filter. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[124] S. Suwajanakorn, C. Hernandez, and S. M. Seitz. Depth from focus with your mobile phone. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015.

[125] Y.-W. Tai and M. S. Brown. Single image defocus map estimation using local contrast prior. In *IEEE International Conference on Image Processing (ICIP)*, pages 1797–1800. IEEE, 2009.

[126] H. Tang, S. Cohen, B. Price, S. Schiller, and K. N. Kutulakos. Depth from defocus in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[127] H. Tang and K. N. Kutulakos. What does an aberrated photo tell us about the lens and the scene? In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–10, 2013.

[128] How to properly use the thin-lens formula. http://www.panohelp.com/thinlensformula.html.

[129] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[130] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016.

[131] D. Vaquero, N. Gelfand, M. Tico, K. Pulli, and M. Turk. Generalized autofocus. In *IEEE Workshop on Applications of Computer Vision (WACV'11)*, January 2011.

[132] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Trans. Graph.*, 26(3), July 2007.

[133] N. Wadhwa, R. Garg, D. E. Jacobs, B. E. Feldman, N. Kanazawa, R. Carroll, Y. Movshovitz-Attias, J. T. Barron, Y. Pritch, and M. Levoy. Synthetic depth-of-field with a single-camera mobile phone. *ACM Transactions on Graphics (TOG)*, 37(4):64, 2018.

[134] T.-C. Wang, J.-Y. Zhu, N. K. Kalantari, A. A. Efros, and R. Ramamoorthi. Light field video capture using a learning-based hybrid imaging system. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017)*, 36(4), 2017.

[135] S. Wanner, S. Meister, and B. Goldluecke. Datasets and benchmarks for densely sampled 4d light fields. In *Proceedings of the Vision, Modeling, and Visualization Workshop*, 2013.

[136] B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolutional network. *CoRR*, abs/1505.00853, 2015.

[137] L. Xu, J. S. J. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pages 1790–1798, 2014.

[138] G. Yang and B. J. Nelson. Wavelet-based autofocusing and unsupervised segmentation of microscopic images. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 3, 2003.

[139] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar. Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum. *IEEE Transactions on Image Processing*, 19(9):2241–2253, Sept 2010.

[140] L. Zhang and S. Nayar. Projection defocus analysis for scene capture and image display. *ACM Trans. Graph.*, 25(3), July 2006.

[141] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

[142] W. Zhang and W.-K. Cham. Single-image refocusing and defocusing. *IEEE Transactions on Image Processing*, 21(2):873–882, 2012.

[143] C. Zhou, O. Cossairt, and S. Nayar. Depth from diffusion. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1110–1117, June 2010.

[144] C. Zhou, D. Miau, and S. K. Nayar. Focal sweep camera for space-time refocusing. *Technical Report, Department of Computer Science, Columbia University*, CUCS-021-12, 2012.

[145] C. Zhou, D. Miau, and S. K. Nayar. Focal sweep camera for space-time refocusing. *Technical Report, Department of Computer Science, Columbia University*, 2012.

[146] C. Zhou and S. Nayar. What are good apertures for defocus deblurring? In *2009 IEEE International Conference on Computational Photography (ICCP)*, pages 1–8, April 2009.

[147] X. Zhu, S. Cohen, S. Schiller, and P. Milanfar. Estimating spatially varying defocus blur from a single image. *IEEE Transactions on image processing (TIP)*, 22(12):4879–4891, 2013.

[148] S. Zhuo and T. Sim. Defocus map estimation from a single image. *Pattern Recognition*, 44(9):1852–1858, 2011.

[149] T. E. Zickler, P. N. Belhumeur, and D. J. Kriegman. Helmholtz stereopsis: Exploiting reciprocity for surface reconstruction. *International Journal of Computer Vision*, 49(2):215–227, Sep 2002.