Novel Imaging Systems Using Nanophotonic Devices

by

Zhengyu Huang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical and Computer Engineering)
in The University of Michigan
2021

Doctoral Committee:

Professor Theodore B. Norris, Chair
Professor Jeffrey A. Fessler
Professor Odest Chadwicke Jenkins
Associate Professor Zhaohui Zhong

Zhengyu Huang
zyhuang@umich.edu
ORCID ID: 0000-0003-1919-5285

DEDICATION

To my parents

ACKNOWLEDGEMENTS

The past years of study at University of Michigan has been a wonderful and amazing journey for me. I am grateful to all the people who have accompanied me and supported me.

I feel tremendously lucky to have had the opportunity to work with Prof. Theodore Norris. I am very grateful to him for agreeing to take me on as a graduate student and all the supports he provided throughout my study. He is a divergent thinker who always comes up with great ideas and suggestions. He has a strong passion to explore the frontier of research and I am fortunate to have the chance to work on many interesting research projects. I could not have asked for a better mentor.

I am very grateful to Prof. Jeffrey Fessler, Prof. Zhaohui Zhong and Prof. Odest Chadwicke Jenkins, for serving on my doctoral committee and all their guidance and constructive feedback they provided. This dissertation would not be possible without their help. I appreciate the constant discussions with Prof. Fessler on projects and the close collaboration with Prof. Zhong on the hardware device. I would like to thank Prof. Evgenii Narimanov for the collaboration on the hyperbolic metamaterial project. I am thankful to Prof. Il Yong Chun for working together on the light field reconstruction project. I also want to thank Panqu Wang for his mentoring on stereo depth estimation project during my internship.

I am also glad to work with my great lab members: Jessica Ames, Momchil Mihnev, You-Chia Chang, Miao-Bin Lien, Heather George, Gong Cheng, Nooshin Mohammadi Estakhri, Zhen Xu, Yifan Shen and Liangqing Cui. I would like to thank Miao-Bin for introducing me to the interesting field of light field photography. I would like to thank Gong Cheng for the collaboration on the THz dichroism project, and thank Dehui Zhang and Zhen Xu for working together on the focal stack camera project.

I would also like to thank all my friends who accompanied me during my PhD study. It is them who make the life in Ann Arbor much more enjoyable. I want to say special thanks my two best friends Long Cheng and Tong Zhu, for their constant help and great time spent together.

Finally, I would like to express my deepest gratitude to my parents. Thank you for raising me up with all the love and kindness, and for supporting me and caring me all the time.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF FIGURES ix

LIST OF TABLES xv

LIST OF APPENDICES xvi

ABSTRACT xvii

CHAPTER

I. Introduction 1

1.1 Concepts in imaging 2
1.1.1 Microscopy 2
1.1.2 Light field photography 3
1.1.3 Focal stack photography 5
1.2 Emerging nanomaterials and new imaging approaches 7
1.2.1 Hyperbolic metamaterial 7
1.2.2 Graphene 8

II. Nanoscale Fingerprinting with Hyperbolic Metamaterials . . 13

2.1 Introduction 13
2.2 HMM device design and working principle 14
2.3 Operating configurations 16
2.4 COMSOL implementation 18
2.5 Deviation from effective medium approximation 20
2.6 3D geometry noise analysis 21
2.7 Summary 22

III. Learning Based Light Field Reconstruction 23

3.1 Introduction . . . 23
3.2 Iterative Neural Networks for Light Field Reconstruction . . . 24
3.2.1 Momentum-Net structure . . . 25
3.2.2 Benefits of Momentum-Net . . . 27
3.2.3 Momentum-Net experimental setup . . . 28
3.2.4 Momentum-Net results . . . 29
3.3 Non-iterative neural networks for light field reconstruction . . . 33
3.3.1 Algorithm for non-iterative light field reconstruction 33
3.3.2 Training of depth estimation NN and refining NN . 36
3.3.3 Experimental setup . . . 36
3.3.4 Results . . . 37
3.4 Summary . . . 38

IV. Unsupervised Depth Estimation from Focal Stack . . . 40

4.1 Introduction . . . 40
4.2 Related work . . . 40
4.3 Method . . . 41
4.3.1 All-in-focus image estimation . . . 41
4.3.2 Differentiable focal stack synthesis . . . 42
4.3.3 Network training . . . 43
4.4 Experimental setup and results . . . 43
4.5 Summary . . . 45

V. Focal Stack Based 3D Tracking . . . 46

5.1 Introduction . . . 46
5.2 Focal stack imaging with transparent sensors . . . 47
5.3 3D tracking of point objects . . . 47
5.4 3D extended object tracking . . . 51
5.5 Summary . . . 52

VI. Focal Stack Camera Depth Estimation Performance Comparison and Design Exploration . . . 53

6.1 Introduction . . . 53
6.2 Methods . . . 54
6.2.1 Focal stack depth imaging . . . 54
6.2.2 Light field depth imaging . . . 54
6.2.3 Network structure . . . 54
6.2.4 Focal stack dataset . . . 55
6.3 Experiments and results . . . 57
6.3.1 Training setup . . . 57
6.3.2 Sensor resolution dependence . . . 57

6.3.3 Aperture size dependence . . . 58
6.3.4 Focal stack and light field camera comparison . . . 58
6.4 Summary . . . 62

VII. Secure Imaging using Focal Stack Camera . . . 63

7.1 Introduction . . . 63
7.2 Related work . . . 65
7.2.1 Image inpainting . . . 65
7.2.2 Forgery localization . . . 67
7.2.3 Focal stack . . . 67
7.3 Method . . . 68
7.3.1 Generating CNN inpainted focal stack . . . 69
7.3.2 Detecting CNN inpainted focal stack . . . 70
7.4 Experiments and results . . . 71
7.4.1 Implementation . . . 71
7.4.2 Results . . . 72
7.5 Summary . . . 78

VIII. Conclusions and Future Work . . . 79

8.1 Nanoscale fingerprinting with hyperbolic metaterials . . . 79
8.2 Learning based light field reconstruction . . . 80
8.3 Unsupervised depth estimation from focal stack . . . 81
8.4 Focal stack based 3D tracking . . . 81
8.5 Focal stack camera design exploration . . . 82
8.6 Secure imaging using focal stack camera . . . 82

APPENDICES . . . 83
A.1 Volume plasmon polariton modes in HMM . . . 83
A.2 Effective medium theory (EMT) description of the permittivity tensor . . . 85
A.3 Calculation of the scattering strength . . . 85
A.4 Target material dependence of the scattering strength . . . 85
A.5 Target shape dependence of the scattering strength . . . 87
B.1 Single-point object focal stack from CMOS camera . . . 88
B.2 Synthesizing multi-point object focal stack . . . 88
B.3 Extended object focal stack . . . 89
B.4 Neural network architectures and training . . . 89
B.5 Ranging performance comparison . . . 93
C.1 Effect of JPEG augmentation for training . . . 100
D.1 Focal stack collection . . . 102
D.2 Focal stack alignment . . . 102
D.3 Depth registration . . . 103

BIBLIOGRAPHY 105

LIST OF FIGURES

Figure

1.1 (a) Airy pattern produced by imaging a point object using a perfect lens with a circular aperture.(b) Airy pattern produced by light from two point sources passing through a circular aperture and meets the Rayleigh criterion. 3

1.2 Example sub-aperture images of the 'boardgame' light field in the HCI dataset. 4

1.3 Example horizontal EPIs of the 'boardgame' light field in the HCI dataset. 5

1.4 Illustration of circle of confusion for a sensor plane with focusing distance d_f. 6

1.5 Iso-frequency curves of HMM. (a) Type I HMM. (b) Type II HMM. Adapted from [83]. 7

1.6 Example HMM structure. (a) metal-dielectric layered structure. (b) Wire array structure. Adapted from [83]. 8

1.7 (a) Graphene lattice. (b) Graphene reciprocal lattice in k-space. (c) Graphene energy band structure. Adapted from [72]. 9

1.8 (a) Photograph of an aperture partially covered by graphene. The line scan profile shows the intensity of transmitted white light along the yellow line. (b) Transmittance spectrum of single-layer graphene (open circles). The red line is the transmittance expected for 2D Dirac fermions. The green curve takes into account a nonlinearity and triangular warping of graphene's electronic band structure. Adapted from [66]. 10

1.9 (a) Device structure of the graphene phototransistor. (b) Schematic of band diagram and photoexcited hot carrier transport under light illumination. Electrons and holes are represented by grey and red spheres, respectively. Vertical arrows represent photoexcitation, and lateral arrows represent tunnelling of hot electron (grey) and hole (red). Adapted from [55]. 11

1.10 (a) Schematic showing simultaneous capture of multiple images of a 3D object on different focal planes using focal stack camera. Inset: photograph of focal stack camera used in experiments with two transparent focal planes. (b) Upper panel: optical image of a 4×4 transparent graphene photodetector array, Lower panel: schematic of the all-graphene phototransistor design. It includes a top graphene layer as transistor channel and a bottom graphene patch as floating gate, separated by a silicon tunneling barrier (purple). 11
2.1 (a): Structure design of the HMM. (b): The norm square of the scattered electric field for the TM wave incident from top at wavelength 1200 nm. 14
2.2 (a): Iso-frequency curve of the type II HMM for the TM wave, illustrating how the localized beam propagation angle θ can be determined. Wavevector k (blue arrow); group velocity v_g (red arrow). (b): Localized beam angle versus wavelength/unit cell size for different wavelengths using exact simulation (solid lines with filled circles). Asterisks: The beam angle of the HMM structure shown in (a) at corresponding wavelengths. The dashed line indicates the beam angle using EMT at corresponding wavelengths. 15
2.3 Two possible device configurations. Dashed lines show the change in the beam direction as the wavelength is increased from shorter (blue) to longer (red). A photodetector measures the scattered power $P(\lambda)$. 17
2.4 (a): Scattering strength versus wavelength for different nanoparticle spacing [see configuration in Fig. 2.3(a)]. n = 1.73 for the bottom target. (b): Scattering strength versus wavelength for different gap sizes [see configuration in Fig. 2.3(b)]. n = 1.73 for both the targets. (c) Scattering strength versus wavelength for four different target material combinations at gap = 100 nm [see configuration in Fig. 2.3(b). L, left target; R, right target]. 18
2.5 The permittivity values of the metal (ε_m) and dielectric (ε_d) layer . 19
3.1 The architecture of the Momentum-Net, showing its updating rules at the i-th iteration. 26
3.2 PSNR maximization comparisons between different INNs (Light field photography system with $n_F = 5$ detectors obtain a focal stack of light fields consisting of $S = 81$ sub-aperture images; averaged PSNR values across three test reconstructed images). 29
3.3 Error map comparisons of reconstructed sub-aperture images (at the angular coordinate $(5, 5)$) from different MBIR methods. The PSNR values in parenthesis were measured from reconstructed light fields. 31
3.4 Comparisons of estimated depths from light fields reconstructed by different MBIR methods. SPO depth estimation [117] was applied to reconstructed light fields 32
3.5 Proposed CNN-based method for light field reconstruction and depth estimation using focal stack. 34

3.6 Sub-aperture images and epipolar slices of the reconstructed light field and the estimated 4D ray depth. (a) Ground truth light field visualized at the corner view. (b) Reconstructed light field via the proposed method at the corner view (PSNR = 42.23 dB). (c) Estimated center view depth via the proposed method. (d) Estimated center view depth via single image sequential CNN [97]. 38
3.7 Error maps of the reconstructed light field sub-aperture view ($u = -1, v = 3$). The PSNR values shown in parenthesis are calculated from reconstructed light fields. 38
4.1 Flow chart of the proposed unsupervised depth from focus method. 41
4.2 Visualization of all-in-focus (AIF) image estimation. First and last image in the focal stack sequence are shown in the first two columns. 3rd column: estimated AIF images. 4th column: ground truth AIF images. 42
4.3 Visualization of the depth estimation result. 44
5.1 Experimental demonstration of focal stack imaging using double stacks of graphene detector arrays. (a) A schematic of measurement setup. A point object (dotted circle) is generated by focusing a green laser beam (532 nm) with the lens. Its position is controlled by a 3D motorized stage. Two detector arrays (blue sheets) are placed behind the lens. An objective and CCD camera are placed behind the detector array for sample alignment. A chopper modulates the light at 500 Hz and a lock-in amplifier records the AC current at the chopper frequency. (b) Images captured by the front and back photodetector planes with objects at three different positions along the optical axis (12 mm, 18 mm, 22 mm respectively). The grayscale images are generated using responsivities for individual pixels within the array, normalized by the maximum value for better contrast. The point source is slightly off-axis in the image presented, leading to the shift of spot center. (c) The illustrations of the beam profiles corresponding to the imaging planes in (b). The focus is shifting from the back plane (top panel) toward the front plane (bottom panel). 48
5.2 (a-b) Tracking results for single point object. Results are based on images captured with the graphene photodetector arrays. (c-d) Tracking results for three points objects. Results are based on data synthesized from multi focal-plane CMOS images (downsampled to 9 $\times$ 9) of single point source. (e): Tracking results for rotating two-point objects on one testing trajectory. Results are based on data synthesized from single point source images captured with graphene photodetector arrays. 50
5.3 3D extended-object tracking and its orientation estimation using focal stack data collected by a CMOS camera, in (a) the x-y-plane perspective and (b) in the x-z-plane perspective. The estimated (true) ladybug's position and orientation are indicated by green (orange) dots and green (orange) overlaid ladybug images. 52

6.1 Network structure for depth estimation from focal stack. All convolutions have filter size of 3×3, stride 1, and the output channel number for each layer is indicated beneath. Blue border around a layer indicates that Batch Normalization and leaky ReLU are applied to the output. Red border indicates tanh non-linearity is applied to the output. n_{F} is the number of images in the focal stack. 55
6.2 Example focal stacks showing the 2nd, 4th and 6th images in the stack sequence. Last column shows the ground truth depth maps. Rows correspond to HCI dataset, DDFF dataset, CVIA dataset and Nikon dataset, respectively. 56
6.3 Example focal stacks with different camera parameters in Nikon dataset. (a) Schematic illustrating focal stack generation with down-sample rate = 3. (b) Focal stack examples ($n_{\mathrm{F}} = 2$) captured with different down-sample rate and aperture setting. The depth estimated from the focal stack and the ground truth depth are also shown. 59
6.4 RMSE of the depth estimated from focal stack images on DDFF dataset, CVIA dataset and Nikon dataset as a function of resolution down-sample rate (left column), aperture size (right column) and number of sensor planes n_{F}. 60
6.5 Qualitative disparity estimation results from light field data and focal stack data. (a) Results on HCI dataset. (b) Results on DDFF dataset. 61
7.1 Focal stack system for inpainting region localization. (a) Imaging system schematic showing depth dependent defocus blur of a cube-ball object. (b) Inpainting localization CNN estimates inpainting regions from a focal stack. 65
7.2 Example real and inpainted focal stacks. Only the first and the last image in each focal stack is shown. The region to be inpainted is shown as white in the second row. 68
7.3 Localization F_1 scores for focal stack data with networks trained on GMCNN dataset with JPEG augmentation and tested on GMCNN data (1st column), EdgeConnect (2nd column) and Gated Convolution (3rd column) datasets. The robustness against Gaussian noise (1st row), resizing (2nd row) and JPEG compression(3rd row) are shown for each model. Symbol '*' on x-axis indicates the result without JPEG compression. 69
7.4 Example localization results of the model trained on GMCNN dataset and tested on Gated Convolution dataset. Probability threshold of 0.5 is used for classification. F_1 scores are indicated in green for each prediction. 71
7.5 Localization F_1 scores for focal stack data with networks trained on EdgeConnect dataset with JPEG augmentation and tested on GMCNN (1st column), EdgeConnect (2nd column) and Gated Convolution (3rd column) datasets. Symbol '*' on x-axis indicates the result without JPEG compression. 73

7.6 Localization F_1 scores for focal stack data with networks trained on Gated Convolution dataset with JPEG augmentation and tested on GMCNN (1st column), EdgeConnect (2nd column) and Gated Convolution (3rd column) datasets. Symbol '*' on x-axis indicates the result without JPEG compression. 74
7.7 Localization F_1 scores for focal stack data with networks trained on GMCNN dataset with JPEG augmentation and tested on GMCNN (1st column), EdgeConnect (2nd column) and Gated Convolution (3rd column) datasets, showing the total pixel dependence. Symbol '*' on x-axis indicates the result without JPEG compression. 76
A.1 Effective medium theory calculation of the permittivity tensor components, using eqn. A.2. Device behaves as type II HMM ($Re(\varepsilon_\perp) < 0$, $Re(\varepsilon_\parallel) > 0$)) in the yellow shaded region and as normal anisotropic medium ($Re(\varepsilon_\perp)$, $Re(\varepsilon_\parallel) > 0$)) in the green shaded region. 84
A.2 The scattered power versus wavelength in the system without target. 86
A.3 Scattering strength versus wavelength for different bottom target material at fixed spacing of 400 nm. 86
A.4 (a) The device configuration after changing the shape of the bottom target from square to semi-circle. (b) Comparison of scattering strength versus wavelength for semi-circle target (red) and square target (blue). Both targets are at spacing of 400 nm. 87
B.1 Experimental set-up for capturing the extended object (ladybug) focal stack, using CMOS sensor. 90
B.2 Neural network architectures for 3D ranging. B is the general batch size of the data (e.g., in training, B is the training batch size; in testing with a single sample, B= 1). (a) Network for estimating single point object's x or y coordinate. (b) Network for estimating single point object's z coordinate. (c) Network for estimating M-point object's (x_i, y_i, z_i) coordinates tuple. 90
B.3 Convolutional neural network architectures for extended object tracking and orientation estimation. B is the general batch size of the data (e.g., in training, B is the training batch size; in testing with a single sample, B= 1). (a) Network for estimating extended object's spatial coordinates (x, y, z). (b) Network for estimating extended object's orientation. 92
B.4 Single-point object tracking performance (only 10 test samples are shown). Focal stack data from: (a-b) 4×4 transparent graphene detector. (c-d) 4×4 CMOS sensor. (e-f) 9×9 CMOS sensor. (g-h) 32×32 CMOS sensor. (i-j) 4×4 Avg. 20 CMOS sensor. (k-l) 9×9 Avg. 20 CMOS sensor. 95
B.5 2-point object with 2 possible shapes tracking performance (only 7 test samples are shown). Focal stack data from: (a-b) 4×4 transparent graphene detector. (c-d) 4×4 CMOS sensor. (e-f) 9×9 CMOS sensor. (g-h) 32×32 CMOS sensor. (i-j) 4×4 Avg. 20 CMOS sensor. (k-l) 9×9 Avg. 20 CMOS sensor. 96

B.6 2-point object with 3 possible shapes tracking performance (only 7 test samples are shown). Focal stack data from: (a-b) 4×4 transparent graphene detector. (c-d) 4×4 CMOS sensor. (e-f) 9×9 CMOS sensor. (g-h) 32×32 CMOS sensor. (i-j) 4×4 Avg. 20 CMOS sensor. (k-l) 9×9 Avg. 20 CMOS sensor. 97
B.7 3-point object with 2 possible shapes tracking performance (only 4 test samples are shown). Focal stack data from: (a-b) 4×4 transparent graphene detector. (c-d) 4×4 CMOS sensor. (e-f) 9×9 CMOS sensor. (g-h) 32×32 CMOS sensor. (i-j) 4×4 Avg. 20 CMOS sensor. (k-l) 9×9 Avg. 20 CMOS sensor. 98
B.8 3-point object with 3 possible shapes tracking performance (only 4 test samples are shown). Focal stack data from: (a-b) 4×4 transparent graphene detector. (c-d) 4×4 CMOS sensor. (e-f) 9×9 CMOS sensor. (g-h) 32×32 CMOS sensor. (i-j) 4×4 Avg. 20 CMOS sensor. (k-l) 9×9 Avg. 20 CMOS sensor. 99
C.1 Localization F_1 scores for focal stack data with networks trained on GMCNN dataset without JPEG augmentation and tested on GMCNN (1st column), EdgeConnect (2nd column) and Gated Convolution (3rd column) datasets. Symbol '*' on x-axis indicates the result without JPEG compression. 101
D.1 Setup of the RGB camera and the Intel RealSense D415 Depth Camera. 103
D.2 The first and the last image of the focal stack used for focal stack alignment. 104
D.3 Example images used for RGB camera and depth camera calibration. 104

LIST OF TABLES

Table

3.1 Average PSNR of the reconstructed light field and reconstruction time (on CPU/GPU) for 100 test samples. Values in parenthesis are GPU reconstruction times. 39

4.1 Result of unsupervised depth from focus. 45

6.1 RMSE of depth map estimated from focal stack and light field. Focal stack of $n_{\mathrm{F}} = 7$ is used. For DDFF and HCI, the RMSE is calculated on the disparity map with unit of pixel. For CVIA, the RMSE is calculated on the depth map with unit of meter. Largest possible aperture is used in all experiments. 59

7.1 F_1 scores of the model trained on GMCNN inpainted focal stacks with focusing disparity range [-1, 0.3], and evaluated on focal stacks inpainted by GMCNN, EdgeConnect and Gated Convolution. Three values in each field correspond to the results on focal stacks with focusing disparity range [-1, 0.3], [-0.8, 0.5] and [-1.2, 0.5], respectively. 76

B.1 Single-point object 3D ranging RMSE (unit: mm) table on testing set. Avg. 20 means spatial averaging with window size 20 is performed on the raw high-resolution focal stack. 93

B.2 Multi-point object 3D ranging RMSE (unit: mm) table of x on testing set. Avg. 20 means spatial averaging with window size 20 is performed on the raw high-resolution focal stack. First column encodes different object configurations, e.g., 2p3s means 2-point object with 3 possible shapes. 93

B.3 Multi-point object 3D ranging RMSE (unit: mm) table of y on testing set. Avg. 20 means spatial averaging with window size 20 is performed on the raw high-resolution focal stack. First column encodes different object configurations, e.g., 2p3s means 2-point object with 3 possible shapes 94

B.4 Multi-point object 3D ranging RMSE (unit: mm) table of z on a testing set. Avg. 20 means spatial averaging with window size 20 is performed on the raw high-resolution focal stack. First column encodes different object configurations, e.g., 2p3s means 2-point object with 3 possible shapes. 94

LIST OF APPENDICES

Appendix

A. Nanoscale Fingerprinting with Hyperbolic Metamaterials 83

B. Focal Stack Based 3D Tracking 88

C. Secure Imaging using Focal Stack Camera 100

D. Focal Stack Camera Depth Estimation Performance Comparison and Design Exploration 102

ABSTRACT

Novel technologies and innovations lead to new applications. This dissertation demonstrates new applications enabled by novel nanophotonic devices. I will describe two nanophotonic devices: hyperbolic metamaterial and transparent graphene photodetector and show their applications when combined with proper algorithms.

The first nanophotonic device I will introduce is hyperbolic metamaterial. Hyperbolic metamaterial has been known to support high-k mode waves. Several methods have been proposed to use hyperbolic metamaterial for imaging beyond the diffraction limit. However, they suffer from high loss, or require coherent illumination. We take a different route to this task and propose a novel method for nanostructure discrimination based on hyperbolic metamaterial. Instead of imaging the objects of interest, we showed that nano-sized objects with different configuration have different scattering spectrum, which could be used for fingerprinting and discriminating between different object configurations with deep subwavelength resolution.

The second nanophotonic device I will describe is the focal stack camera made from transparent graphene photodetectors. Single layer graphene is highly transparent, which only absorbs about 2% of the light. The recent developed focal stack camera, made from multiple planes of such transparent graphene imaging array stacked along the optical axis, is able to capture the focal stack of a scene in a single exposure. Note that before the introduction of such focal stack camera, capturing of a focal stack is only possible for a static scene using sequential exposure by adjusting the focusing depth. Combining with proper algorithms, we demonstrated several applications using such focal stack data, including light field reconstruction, depth estimation, 3D object tracking and secure imaging: we proposed an iterative neural network based method, Momentum-Net, for light field reconstruction, with improved convergence speed compared to existing iterative neural network based methods; We further sped up the reconstruction by proposing a non-iterative learning based method; An unsupervised depth estimation from focal stack method is also developed, which achieves significantly better depth accuracy, compared to single-image based method; We designed a neural network based method to track objects in 3D, without the need of light field reconstruction or depth estimation and achieves great tracking accuracy;

We also demonstrated image forgery detection using focal stack data. Compared with single-image based forgery detection, the proposed focal stack based method has significantly better generalization ability on new unseen datasets and is robust against common post-processing methods. Since the design of the focal stack camera would affect its 3D sensing performance, we investigated the dependence of the camera parameters, on its depth estimation performance. The performance is further compared with the light field camera on several datasets, highlighting scenarios where the focal stack camera might be preferred.

CHAPTER I

Introduction

Imaging is the process of represent or reproduce objects of interests in a visual representation. The process of how we see the world is one familiar example of the imaging process: light rays emanating from the objects enters the human's eye through the pupil, get refracted by the cornea and lens and then form an image on the retina. Cameras also work in a similar way, with the role of the pupil replaced by a lens aperture, and the retina by a image sensor. Imaging are not limited to the macroscopic world. It also plays a very important role in the microscopic world. Microscopes are developed for this purpose and they allow us to examine extremely small structures of interests, such as bacterial, protein and DNA.

Countless efforts have been made to improve the imaging speed, image quality and the image resolution. With innovations in the imaging hardware and software, enormous progress has been achieved. For example, in the history of photography, photosensitive films were used to detect the light, which requires additional chemical processes to develop the final image. The introduction of electronic image sensors such as CCD or CMOS to detect the light, has revolutionized the way photos are taken: images nowadays are stored as digital signals and films are no longer needed. It also enables additional post-processing and editing to the captured images.

This dissertation concerns imaging-related applications in both microscopic world and macroscopic world, enabled by novel nanophotonic devices. The thesis is organized as follows: chapter II presents a novel method of nanoscale structure fingerprinting and discrimination using hyperbolic metamaterial, with deep subwavelength resolution. Chapter III describes learning based light field reconstruction from focal stack methods, improving the reconstruction quality and speed. Chapter IV introduces an unsupervised learning based method for estimating depth from focal stack. Chapter V presents a fast and accurate 3D object tracking method using focal stack. Chapter VII proposes to use focal stack as a tamper-evident secure image file to make

the images and videos more secure against malicious manipulation and forgery. The remaining part of this chapter provides background information of our works in this dissertation.

1.1 Concepts in imaging

1.1.1 Microscopy

A typical imaging system consists of a lens system, and an image sensor. Over the past years, lots of efforts have been spent to design lens using computer-aided software, and grind them precisely to minimize the aberration and improve the image quality. However, even with a perfect lens with no aberration, the resolution of the microscope is still limited by a fundamental physical limit due to the diffraction of the light. Such ideal imaging system is called diffraction-limited imaging system and the spatial resolution $\Delta_{x,y}$ it can achieve is on the order of λF, where λ is the wavelength of the light used for imaging and F is the f-number of the imaging system.

Diffraction refers to the phenomenon of the bending of waves around the corners of an obstacle or through an aperture. It can be explained by the Huygens–Fresnel principle: each point in a propagating wavefront behaves like an individual spherical wavelets. These spherical wavelets can be considered as the new source of the wave. Sum of these spherical wavelets leads to the light propagation and the apparent bending of the light around obstacles.

Fig. 1.1(a) shows the diffraction pattern produced by imaging a point object with perfect lens with circular aperture. Instead of a point according to the geometric optics, it exhibits a circular disk pattern due to the diffraction of light from the circular aperture. The angle between the central maximum and the first minimum is given by $\theta_{\min} = 1.22\frac{\lambda}{A}$, where A is the aperture diameter. The fact that the image of an point object is a disk with finite size, instead of a point, limits the resolution of an imaging system: when two point objects are close to each other with significant overlap, two points can no longer be distinguished clearly, as illustrated in Fig. 1.1(b). Rayleigh criterion is one of the criterion that defines when two patterns are merely distinguishable. It is defined as the case when the first minimum of one Airy pattern overlaps with the maximum of the other Airy pattern. According to the Rayleigh criterion, the resolution of a diffraction-limited system is given by: $\Delta_{x,y} = 1.22\lambda N$.

To resolve objects beyond the diffraction limit, numerous novel imaging systems have been proposed. Betzig et al. [9] proposed to turn the fluorescence of individual molecules on and off, and by imaging the same region multiple times, a super-resolved

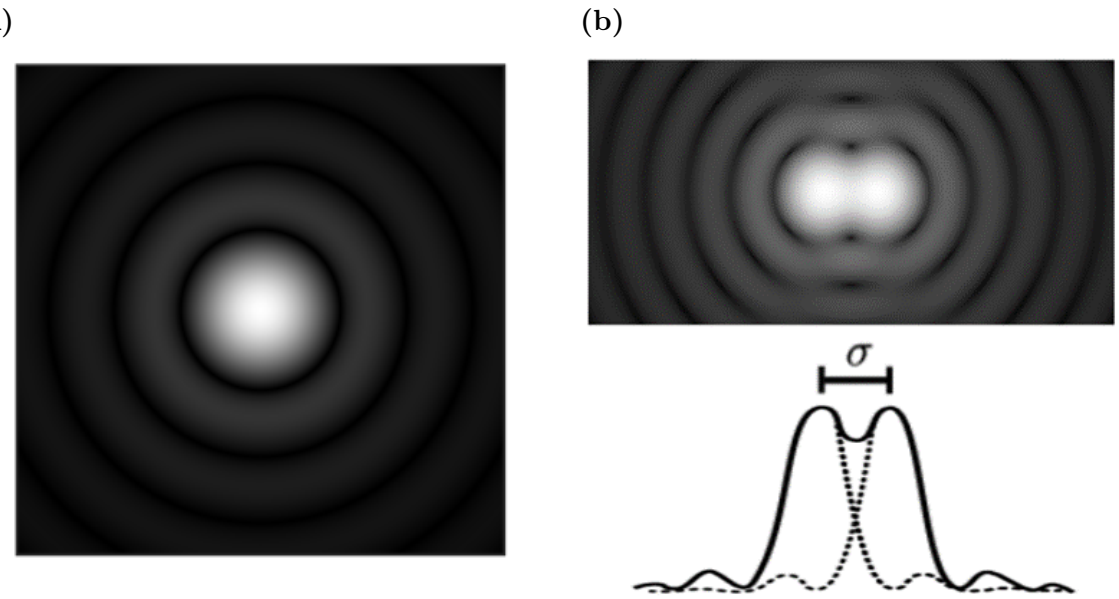

Figure 1.1. (a) Airy pattern produced by imaging a point object using a perfect lens with a circular aperture.(b) Airy pattern produced by light from two point sources passing through a circular aperture and meets the Rayleigh criterion.

image can be obtained. Hell et al. [37] proposed the stimulated emission depletion (STED) microscopy, in which one laser is used to stimulates fluorescent molecules to glow, and another laser is used to cancels out all fluorescence except for that in a manometer-sized volume. Scanning over a 3D volume with nanometer-sized spatial step leads to a super-resolved image. These inventions have wide applications, especially in biology, and were awarded the 2014 Noble Prize in Chemistry.

Superlens [79] was also proposed as a way to achieve super-resolution imaging. It is a slab of special material with negative refractive index, which is able to amplify the evanescent waves that typically decay exponentially along propagation and hence preserve high spatial frequency signals.

1.1.2 Light field photography

Describing the geometric distribution of light is an important topic in computer graphics, which has applications including novel view synthesis [51], synthetic aperture imaging [73, 42], 3D display [61]. Plenoptic function [2], was proposed by Adlson and Bergen for this purpose, which define the distribution of light as a 5D function $P(x, y, z, \theta, \phi)$, 3D for each spatial position (x, y, z) and 2D for each angular direction of the light ray (θ, ϕ). Based on this definition, the value of the plenoptic function $P(x, y, z, \theta, \phi)$ indicates the radiance of the ray at location (x, y, z), traveling along direction (θ, ϕ).

When the light is propagating in the free-space that are free from absorption and scattering, the radiance of the ray is preserved along its propagation. Hence the 5D plenoptic function can be reduced to a 4D function called light field. A 4D light field can be parameterized using two parallel reference planes placed at arbitrary positions. In this two-plane parameterization, every light ray can be identified by its interception at the first plane coordinates $\boldsymbol{\nu} = (u, v)$ and the second plane coordinates $\boldsymbol{x} = (x, y)$, with its radiance being $l(\boldsymbol{x}, \boldsymbol{\nu})$.

Such a 4D light field can be thought of as a collection of 2D conventional images $I^{u_0,v_0}(x, y)$, called sub-aperture images, each with a different view point (u_0, v_0). These sub-aperture images are visually similar to each other, but with minute differences. This is because a point in 3D space maps to different spatial locations in different sub-aperture images due to parallax. For example, a pixel at (x, y) in view (u, v), if unoccluded, corresponds to the pixel at $(x - D, y)$ in view $(u + 1, v)$, where D is the disparity of the 3D point. The disparity D of the point is directly related to its depth d_o as:

$$D = b \cdot f \cdot \left(\frac{1}{d_o} - \frac{1}{d_f} \right), \tag{1.1}$$

where b is the separation between sub-aperture images (baseline), f is the focal length, d_f is the light field focusing depth, d_o is the depth of the 3D object. Fig. 1.2 shows two sub-aperture images of 'boardgame' light field from the HCI light field dataset [38], illustrating the parallax effect due to the change in the viewpoint.

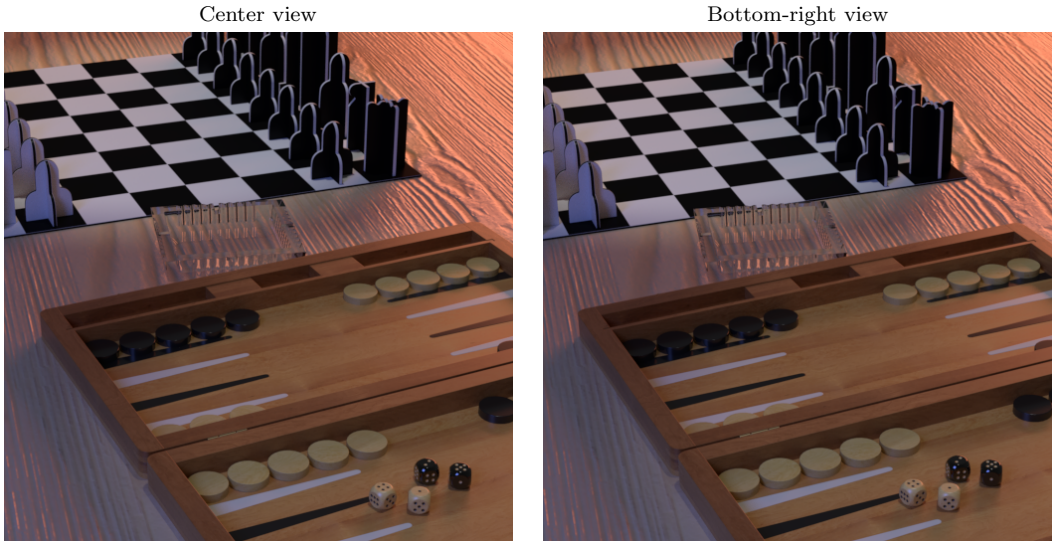

Figure 1.2. Example sub-aperture images of the 'boardgame' light field in the HCI dataset.

Instead of fixed (u, v), as in a sub-aperture image $I^{u_0,v_0}(x, y)$, fixing (x, u) or fixing (y, v) leads to the so called epi-polar images (EPI): $l(x, y_0, u, v_0) \triangleq E^{y_0,v_0}(x, u)$ is called horizontal EPI and $l(x_0, y, u_0, v) \triangleq E^{x_0,u_0}(y, v)$ is called vertical EPI. Fig. 1.3 shows example EPI images of the 'boardgame' light field from the HCI light field dataset. A apparent feature of the EPI is that it has a stripe-like structure with different slopes. It is directly due to the parallax effect described above and can be used as a cue for depth estimation [117]: a vertical stripe in EPI indicates the pixel is at the focusing depth and a positive slope, indicates the pixel is located closer to the focusing depth.

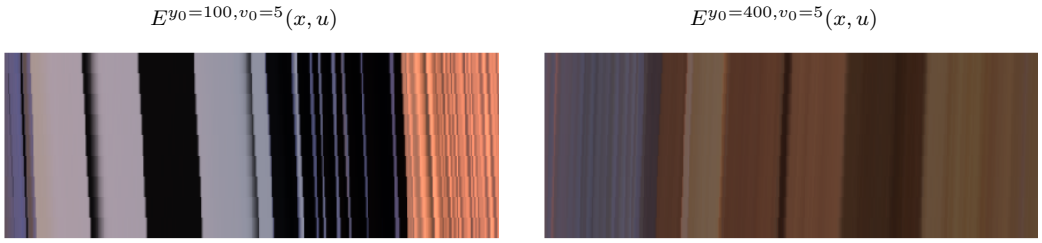

Figure 1.3. Example horizontal EPIs of the 'boardgame' light field in the HCI dataset.

There exists multiple ways to capture a light field, including using an array of cameras, or using a plenoptic camera, such as Lytro and Raytrix[1]. The camera array works by directly capturing images of the scene from different viewpoint. The plenoptic camera, works by multiplexing the 4D light field onto a 2D sensor plane using a microlens array. It is also possible to reconstruct a light field using focal stack data, which will be described in chapter III.

Light field can be used for tasks including depth estimation [117, 93], material recognition [104] and pose estimation [121].

1.1.3 Focal stack photography

Focal stack refers to a set of images of the same scene, captured with varying focus positions. These images contain depth-dependent defocus blur and encode 3D information about the scene.

In conventional focal stack photography, a focal stack is typically collected by multiple exposures with changing focus position. This approach, however, is only applicable to static scenes with no moving objects. Motion artifacts can be minimized by rapid sequential acquisition of the focal stack images; Another method is to use

[1]https://raytrix.de/

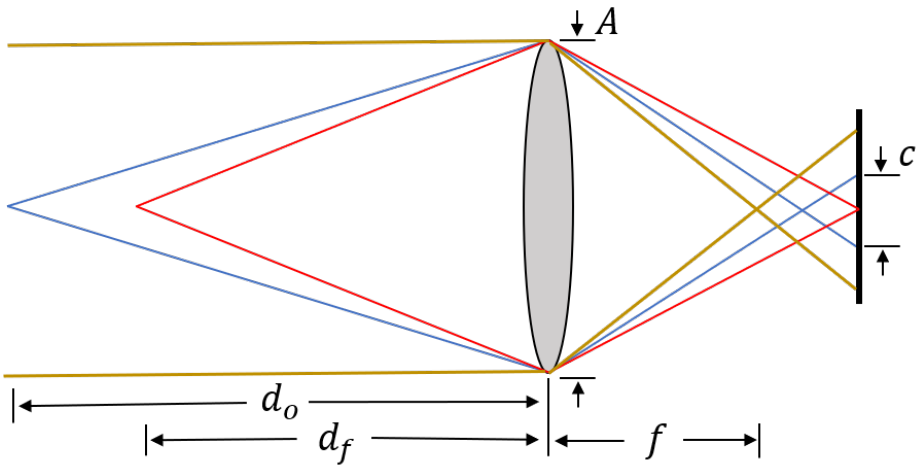

Figure 1.4. Illustration of circle of confusion for a sensor plane with focusing distance d_f.

liquid lens [50], which can change focal length quickly, though such lenses suffer from aberrations and hence degraded image quality.

According to the thin lens equation:

$$\frac{1}{d_o} + \frac{1}{d_i} = \frac{1}{f}, \tag{1.2}$$

where d_o is the object distance, d_i is the image distance and f is the focal length of the lens. Either changing d_i by moving the sensors or changing f using a liquid lens will leads to the image focusing at a different depth and a set of these images form a focal stack. Each image in the focal stack contains depth-dependent defocus blur as illustrated in Fig. 1.4. Specifically, for a camera with an aperture of diameter A, the diameter of the circle of confusion c is given by:

$$c = A\frac{|d_o - d_f|}{d_o}\frac{f}{d_f - f}, \tag{1.3}$$

where d_f is the distance from an in-focus object point to the lens (camera focusing distance) and d_o is the distance from an out-of-focal-plane object to the lens.

Other than directly capturing a focal stack, a focal stack can be synthesized from a light field using the add-shift algorithm [73]. It works by shifting each sub-aperture image of the light field according to the desired focusing depth and then averaging the shifted images. The resulted averaged image is the image focused at the desired focusing depth. Repeating the algorithm with different focusing depth construct a focal stack.

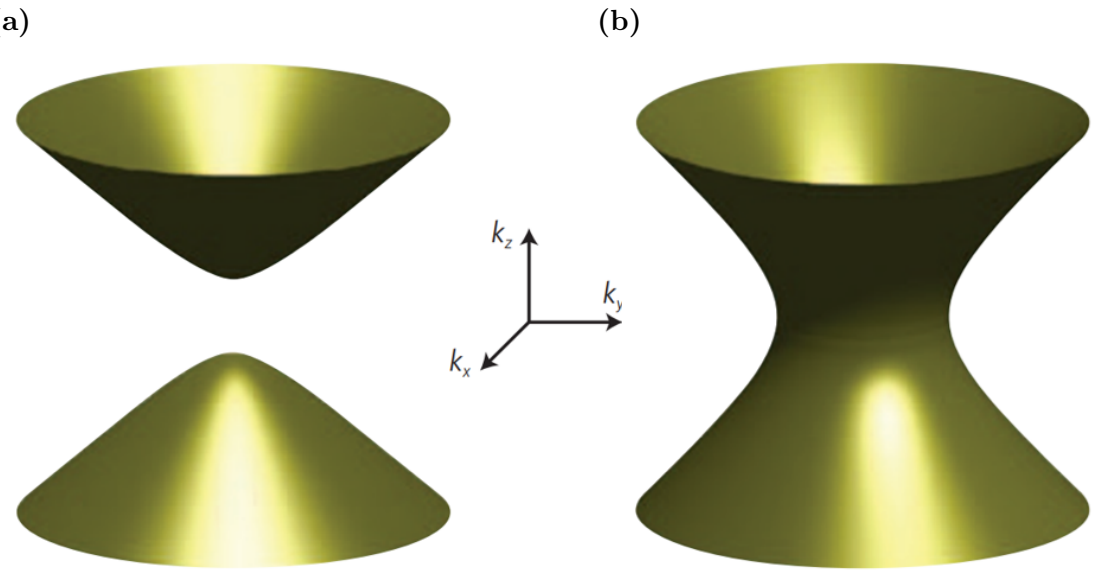

Figure 1.5. Iso-frequency curves of HMM. (a) Type I HMM. (b) Type II HMM. Adapted from [83].

Since the focal stack encodes the 3D information of the scene, when combined with suitable algorithms, it can be used to track 3D objects [114], estimate depth maps [70, 62, 91, 33] and reconstruct light fields [18, 54, 39].

1.2 Emerging nanomaterials and new imaging approaches

1.2.1 Hyperbolic metamaterial

Metamaterials are artificial inhomogeneous structured media with the scale of inhomogeneity that is much smaller than the wavelength of interest; through engineering their structure, special optical properties not existing in natural materials can be realized. In the effective medium approximation, the response of most metamaterials can be characterized by macroscopic effective permittivity ϵ and permeability μ tensors.

Hyperbolic metamaterial (HMM) is one type of metamaterial that exhibits hyperbolic dispersion, i.e., when the real parts of two different primary components of either the dielectric permittivity (electric HMM) or magnetic permeability (magnetic HMM) tensors have opposite signs. Electric HMM (magnetic HMM) can be either classified as type I if $\epsilon_\perp > 0$ and $\epsilon_\parallel < 0$ ($\mu_\perp > 0$ and $\mu_\parallel < 0$) or type II if $\epsilon_\perp < 0$ and $\epsilon_\parallel > 0$ ($\mu_\perp < 0$ and $\mu_\parallel > 0$), where $\epsilon_\perp$, $\mu_\perp$ represent the tensor component perpendicular to the optical axis and $\epsilon_\parallel > 0$, $\mu_\parallel > 0$ represent the tensor component parallel to the optical axis.

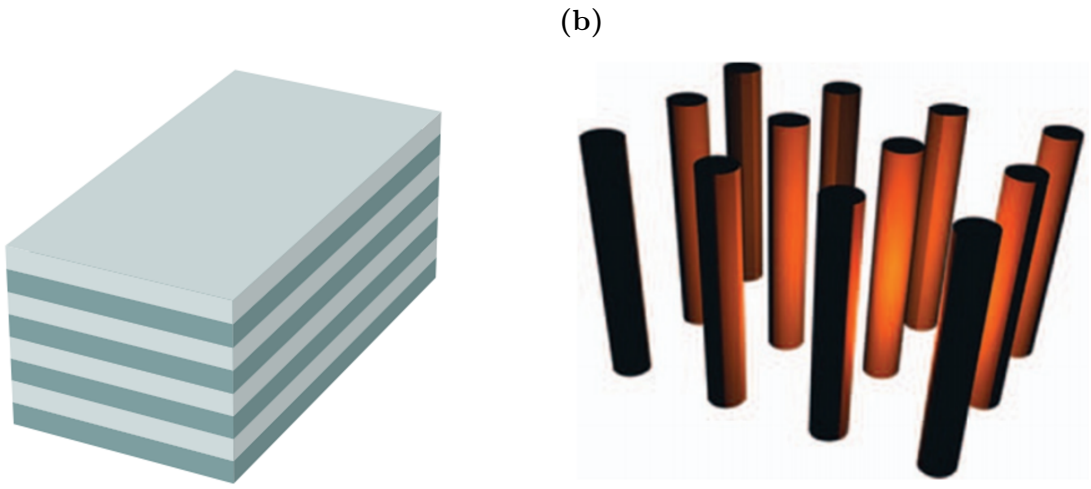

Figure 1.6. Example HMM structure. (a) metal-dielectric layered structure. (b) Wire array structure. Adapted from [83].

Electric HMM can be realized either using a metal-dielectric layered structure or a metallic wire array structure, as shown in Fig. 1.6. Since the permittivity of metal is negative below the plasma frequency, by constraining the electrons in 2D (Fig. 1.6(a)) and 1D (Fig. 1.6(b)), the material will have anisotropic electric response and hyperbolic dispersion regime can be reached via proper design.

The most attractive feature of HMM is that it can support high-k modes, which leads to many novel super resolution imaging applications. Hyper-lens [57] made from HMM was proposed to magnify high spatial frequency features and convert them to low spatial frequency features. These low spatial frequency features can then be imaged by conventional optics. Narimanov proposed to use HMM for hyperstructured illumination [67]. By measuring the phase and amplitude of electric field distribution in the far field, the Motti projection [64] of the object with deep subwavelength resolution can be reconstructed.

1.2.2 Graphene

Graphene is a monolayer of carbon atoms arranged in a two-dimensional (2D) hexagonal honeycomb lattice. Fig. 1.7(a) shows the graphene lattice structure. A unit cell of graphene consists of a pair of carbon atoms $\boldsymbol{A}$ and $\boldsymbol{B}$. Repeating the unit cell along lattice vector $\mathbf{a_1}$ and $\mathbf{a_2}$ forms the entire graphene lattice. Fig. 1.7(b) shows the graphene's reciprocal lattice in k space, which is also hexagonal. The two sets of equivalent points, $\boldsymbol{K}$ and $\boldsymbol{K'}$, which are called Dirac points.

According to the tight-binding model, the graphene has a band structure of the

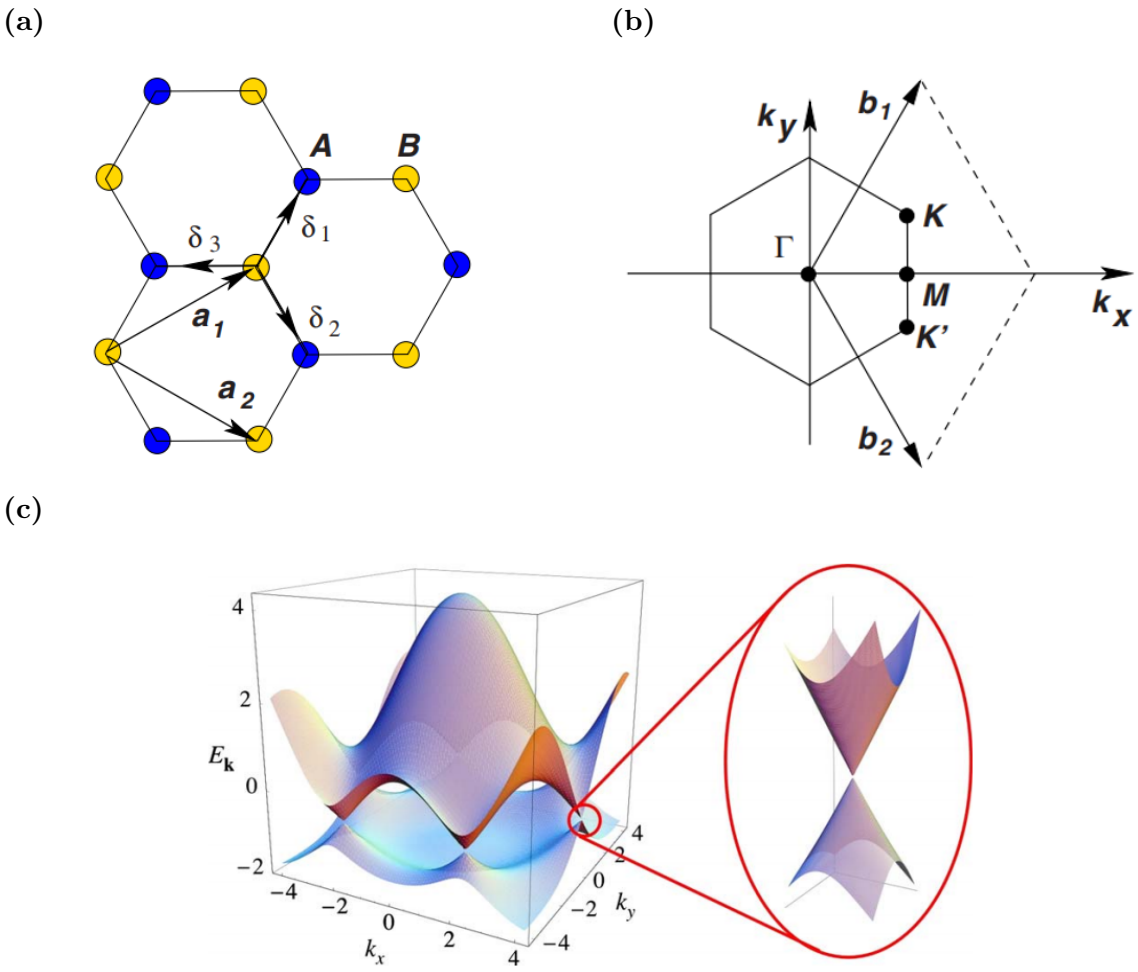

Figure 1.7. (a) Graphene lattice. (b) Graphene reciprocal lattice in k-space. (c) Graphene energy band structure. Adapted from [72].

form:

$$E_{\pm}(\boldsymbol{k}) = \pm t\sqrt{3 + f(\boldsymbol{k})} - t' f(\boldsymbol{k}), \tag{1.4}$$

$$f(\boldsymbol{k}) = 2\cos\left(\sqrt{3}k_y a\right) + 4\cos\left(\frac{\sqrt{3}}{2}k_y a\right)\cos\left(\frac{3}{2}k_x a\right) \tag{1.5}$$

where the plus sign corresponds to the energy of conduction band electrons and the minus sign corresponds to the energy of valence band electrons, t ($\approx 2.8eV$) is the nearest-neighbor hopping energy and t' (in range $[0.02t, 0.2t]$, depending on the tight-binding parameterization) is the next nearest-neighbor hopping energy in the same sub-lattice, a is the carbon-carbon distance (1.42 Å). Fig. 1.7(a) shows the calculated band structure. Around the Dirac points, the electron dispersion relation is approximately linear and is given by:

$$E_{\pm}(\boldsymbol{k}) = \pm v_F \boldsymbol{k}, \tag{1.6}$$

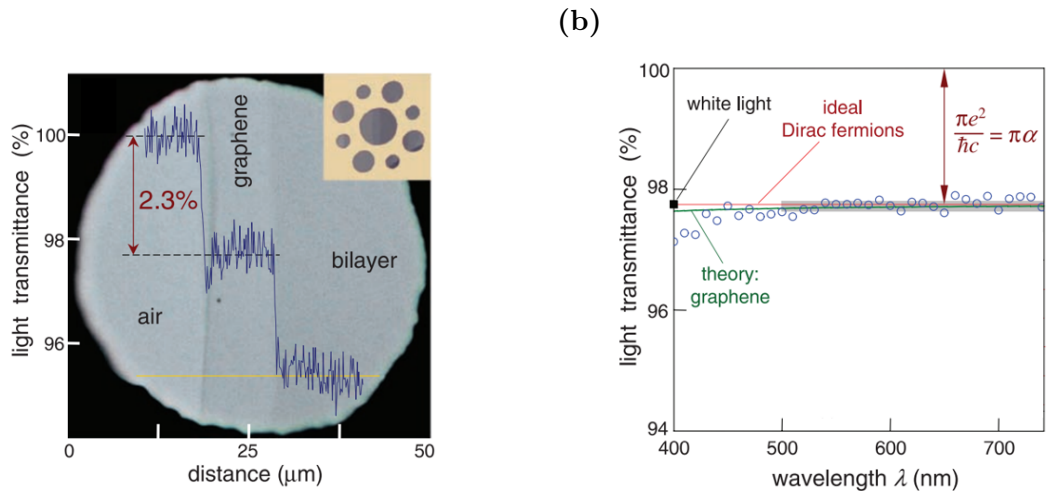

Figure 1.8. (a) Photograph of an aperture partially covered by graphene. The line scan profile shows the intensity of transmitted white light along the yellow line. (b) Transmittance spectrum of single-layer graphene (open circles). The red line is the transmittance expected for 2D Dirac fermions. The green curve takes into account a nonlinearity and triangular warping of graphene's electronic band structure. Adapted from [66].

where $v_F = 3\text{ta}/2 \approx 1 \times 10^6 m/s$ is called the Fermi velocity.

Due to this linear dispersion relationship, graphene has a flat absorption spectrum from visible to the THz frequency, with an absorbance of $\pi e^2/\hbar c \approx 2.3\%$, where $\hbar$ is the reduced Planck's constant. Fig. 1.8 (a) shows a photograph of graphene sample, highlighting its high transparency. Fig. 1.8 (b) shows the measured transmittance spectrum of the graphene (open circles) and its comparison with theoretical calculation.

Numerous methods have been developed to synthesize graphene. Common methods include mechanical exfoliation [75], epitaxial growth [7, 8, 24], and chemical vapor deposition (CVD) growth methods [53]. Graphene obtained from exfoliation method has high quality, but is hard to scale up the production. The epitaxial growth method is scalable and compatible with Si technology. On the other hand, graphene synthesized from CVD method has a lower quality, but is low-cost and have large-area device applications.

Due to the graphene's ultra-broadband absorption spectrum and high carrier mobility, it has great potential to be used for high-speed broadband photodetection. Many graphene-based photodetectors have been proposed in the recent years. Mueller designed a interdigitated metal-graphene-metal photodetector, which achieves a maximum external photoresponsivity of 6.1 mA/W at a wavelength of 1.55 um [65] and an operation speed of 16 GHz can be achieved. Furchi et al. [28] proposed to use a

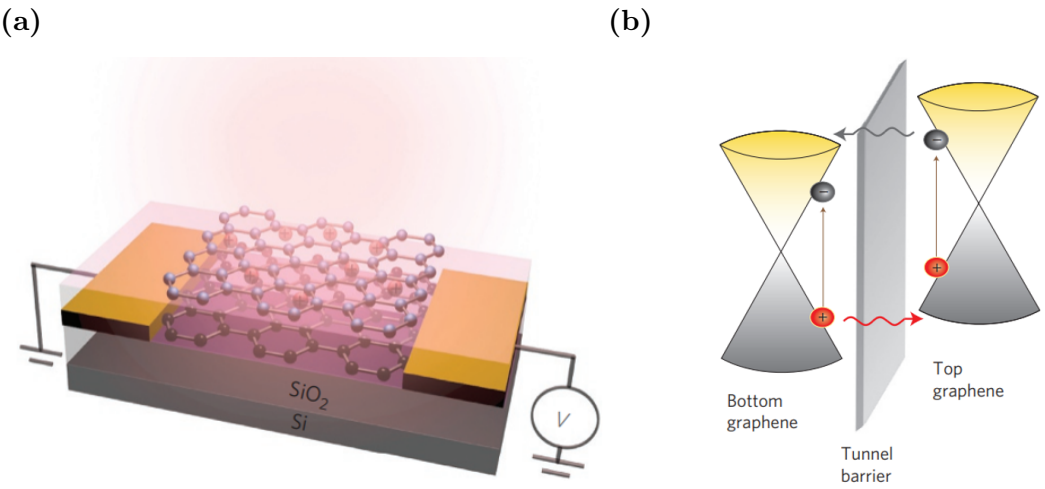

Figure 1.9. (a) Device structure of the graphene phototransistor. (b) Schematic of band diagram and photoexcited hot carrier transport under light illumination. Electrons and holes are represented by grey and red spheres, respectively. Vertical arrows represent photoexcitation, and lateral arrows represent tunnelling of hot electron (grey) and hole (red). Adapted from [55].

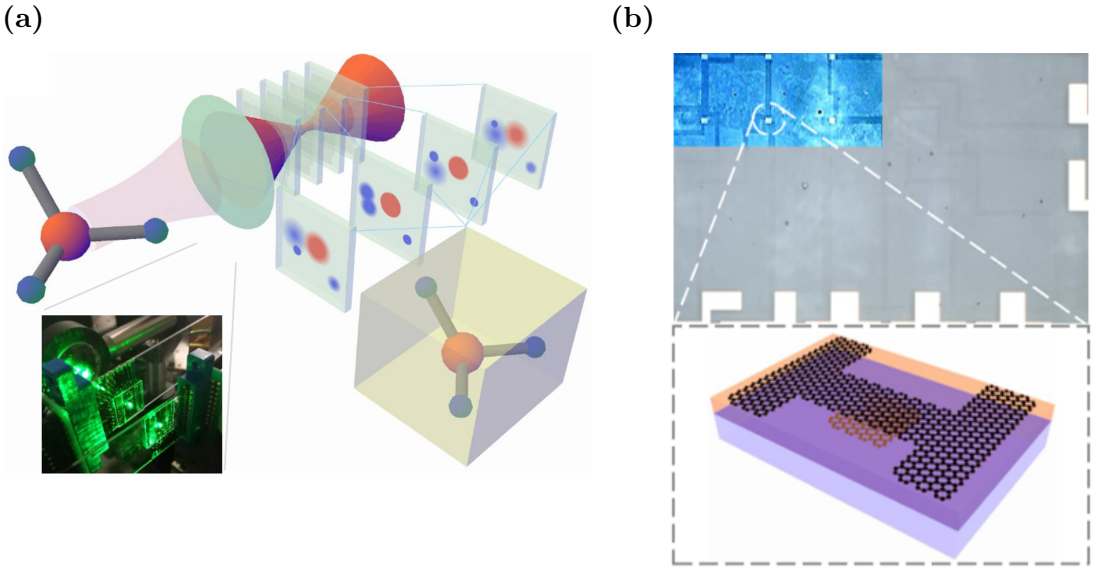

Figure 1.10. (a) Schematic showing simultaneous capture of multiple images of a 3D object on different focal planes using focal stack camera. Inset: photograph of focal stack camera used in experiments with two transparent focal planes. (b) Upper panel: optical image of a 4×4 transparent graphene photodetector array, Lower panel: schematic of the all-graphene phototransistor design. It includes a top graphene layer as transistor channel and a bottom graphene patch as floating gate, separated by a silicon tunneling barrier (purple).

microcavity to enhance the absorption of the graphene and achieves a responsivity of 21 mA/W at the cavity resonance frequency of 864 nm. More recently, research teams at University of Michigan designed a ultra-broadband graphene-based phototransistor with high responsivity [55]. Fig. 1.9 shows the device structure and the working principle. The photodetector consists of two graphene layers sandwiching a thin tunnel barrier. Upon light illumination, charge accumulation on the top graphene layer due to quantum tunnelling leads to strong photogating effect, which results in high photoconductive gain.

In the follow-up works, after replacing the Si substrate by glass and replacing the metal interconnect by graphene, highly transparent graphene photodetectors are demonstrated [54, 114]. A novel focal stack camera, based on such highly transparent graphene photodetectors, was introduced [54, 114]; this optical system can capture a focal stack in a single exposure. Note that this is not possible for conventional focal stack photography, in which sensor movement or lens focus adjustment are needed. The system is illustrated in Fig. 1.10. Such graphene-transistor based image sensor is highly transparent (90 %-95 % transmission), while still maintaining high responsivity. This novel focal stack camera can be applied to applications including light field reconstruction, depth estimation and secure imaging, as will be presented in the following chapters.

CHAPTER II

Nanoscale Fingerprinting with Hyperbolic Metamaterials

2.1 Introduction

To resolve the objects beyond the diffraction limit is of great interests, especially in biology. Fluorescence based super-resolution methods [9, 37] requires sequential image captures and fluorescent labeling, making them inapplicable to real-time dynamic imaging applications. This is also true for near-field approach to super-resolution imaging [10], in which spatial scanning of the sample is needed.

Metamaterials, having extraordinary material properties not found in nature, provide another direction to super-resolution imaging. Superlens [79] is able to enhance the evanescent waves of the object, but it can be subsequently focused and brought to far-field using conventional optics. On the other hand, the later introduced hyperlens [44], was designed to propagate near field signals and magnify it spatially, which can be then imaged by conventional optics. However, both methods have high loss due to the typical use of metallic materials in these metamaterials. Such high loss necessitates a strong illumination to the sample in order to have a good signal to noise ratio in the signal. This is not practical for many applications in biology, where the cells would be damaged by strong intensity illumination.

This chapter proposes an alternative method for discriminating nanoscale objects. Importantly, our proposed method does not capture an image of the object. Instead, it captures a spectral fingerprint of the object and allow us to discriminate nanoscale objects with different spatial separations and/or made from different materials. Contrary to the hyperlens approach where the sample has to be strongly illuminated, the sample in our methods are illuminated by light that is already attenuated by the metamaterial. Our proposed method provides an alternative way for discriminating

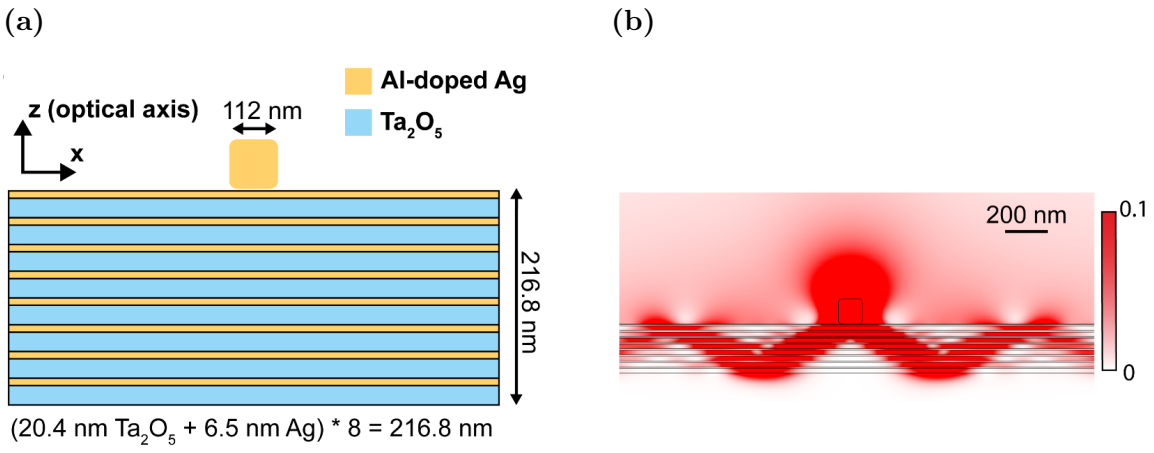

Figure 2.1. (a): Structure design of the HMM. (b): The norm square of the scattered electric field for the TM wave incident from top at wavelength 1200 nm.

nanoscale objects and may find applications in the cases where other methods are not suitable. The result of this project is published in *APL Photonics* [40].

2.2 HMM device design and working principle

In this project, we designed a HMM structure shown in Fig. 2.1(a) and showed that, using HMM, a spectral scan of the scattered intensity can be used to determine the positions of sub-wavelength sized objects. We take advantage of the deep subwavelength resolution from the highly localized beams and we are able to obtain the nanoscale object's spatial and material information by matching the measured spectral characteristics to known records, which we refer to as "fingerprinting."

If the incident field illuminates an object present near an HMM, the scattered field excites Volume Plasma Polariton (VPP) modes [43] inside the HMM; these modes have a highly localized field pattern and propagate along a well-defined direction, as illustrated in Fig. 2.1(b). This highly localized beam has subwavelength width, and its propagation direction is strongly wavelength dependent. By using wavelength-dependent intensity only measurement, it is then possible to resolve the subwavelength structure. We used a 2D finite element analysis (COMSOL 5.2) to demonstrate the feasibility of the proposed method (see appendix section 2.4 for COMSOL implementation details).

In our simulations, the type II HMM device consists of 8 pairs of alternating 20.4 nm Ta_2O_5 and 6.5 nm Al-doped Ag layers [113, 112], as shown in Fig. 2.1(a). An Al-doped Ag scatterer is placed 1 nm above the HMM and is illuminated with a normally incident plane wave. The small distance between the scatterer and the HMM

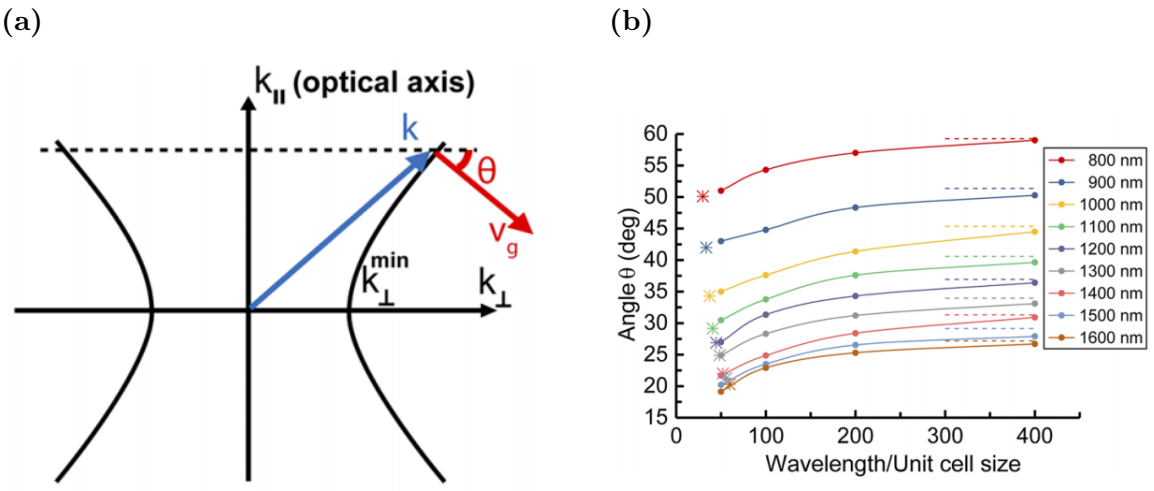

Figure 2.2. (a): Iso-frequency curve of the type II HMM for the TM wave, illustrating how the localized beam propagation angle θ can be determined. Wavevector k (blue arrow); group velocity v_g (red arrow). (b): Localized beam angle versus wavelength/unit cell size for different wavelengths using exact simulation (solid lines with filled circles). Asterisks: The beam angle of the HMM structure shown in (a) at corresponding wavelengths. The dashed line indicates the beam angle using EMT at corresponding wavelengths.

enables coupling of the scattered field into VPP modes inside the HMM. The layered structure is uniaxially anisotropic, and under the effective medium theory (EMT) approximation [90], the structure has $\mathrm{Re}\,(\varepsilon_\perp) < 0$ and $\mathrm{Re}\,(\varepsilon_\parallel) > 0$ for wavelengths larger than 647 nm and behaves as the type II HMM (a calculation of the effective dielectric constants within the EMT approximation is given in appendix section A.2). We note that the use of a type II HMM is necessary in this scheme: for the type I HMM, the direct transmission of the incident field is high and will act as background noise in the experimental measurement. On the contrary, for the type II HMM, most of the incident field is reflected back toward the light source and does not contribute to the measurement noise. This is evident by considering the iso-frequency curve shown in Fig. 2.2(a): there is no mode with $k_\perp = 0$ (normal incident) supported by HMM. Fig. 2.1(b) shows a typical scattered field distribution when the HMM device is illuminated by a plane wave (with an out-of-plane magnetic field) at a wavelength of 1200 nm. Note in the figure that two localized beams (VPP modes) with subwavelength beam width are generated in the HMM. The angle between the beam propagation direction and the x axis in the EMT limit is given by:

$$\theta(\lambda) = \tan^{-1}\left(\sqrt{\frac{\varepsilon_\parallel(\lambda)}{|\mathrm{Re}\,\varepsilon_\perp(\lambda)|}}\right), \tag{2.1}$$

For a small scatterer, the scattered light will have mostly large $k_\perp$ components, so we may approximate $k_\perp$ as $k_\perp \gg \frac{\omega}{c}$, where ω is the angular frequency and c is the speed of light; in that case, eqn. 2.1 can be derived by noting that the group velocity direction is perpendicular to the iso-frequency curve, as illustrated in Fig. 2.2(a) (Further details are provided in appendix section A.1). In Fig. 2.2(b), the dashed lines show the beam angle obtained from EMT and a strong wavelength dependence is clearly apparent. The exact simulation [asterisks in Fig. 2.2(b) also shows a similar wavelength dependence, and the difference between the two will be discussed in section 2.5.

This wavelength dependence, as we will see, is central to our nanoscale fingerprinting concept. The physical processes illustrated in Fig. 2.1(b) are summarized as follows: The top scatterer generates a scattered field under excitation. The propagating components and some evanescent components $\left(k_\perp < k_\perp^{\min}\right)$ of the scattered field are totally reflected at the top HMM/air interface (see appendix section A.1), while the remaining evanescent components $\left(k_\perp \geq k_\perp^{\min}\right)$ of the scattered field from the top scatterer are coupled to the propagating VPP modes in the HMM and form the highly localized beams. The propagation and reflection of the beams at the HMM/air interfaces leads to a zig-zag field pattern. At the bottom HMM/air interface, the beams couple to evanescent modes in air, which gives rise to a highly localized field distribution and can be scattered by nanoscale objects positioned nearby. This highly localized field distribution is wavelength dependent and enables nanoscale fingerprinting; The next section describes two possible configurations (shown in Fig. 2.3) for the proposed HMM device.

2.3 Operating configurations

In the configuration of Fig. 2.3(a), a target object to be identified (with refractive index n = 1.73) is placed below the HMM device, with a certain spacing relative to the top scatterer. By sweeping the wavelength of the incident light, the beam propagation direction θ will change, as indicated by asterisks in Fig. 2.2(b), and the target scatters strongly only at the wavelength for which the beam is localized close to the scatterer. Fig. 2.4(a) shows the scattering strength, defined as scattered power with target scattered power without target, measured as a function of wavelength for different spacing (see appendix section A.3 for calculation of the scattering strength). It can be seen in the curve that targets located at different spacing differ in their peak scattered wavelength and the peak position shifts to longer wavelength monotonically

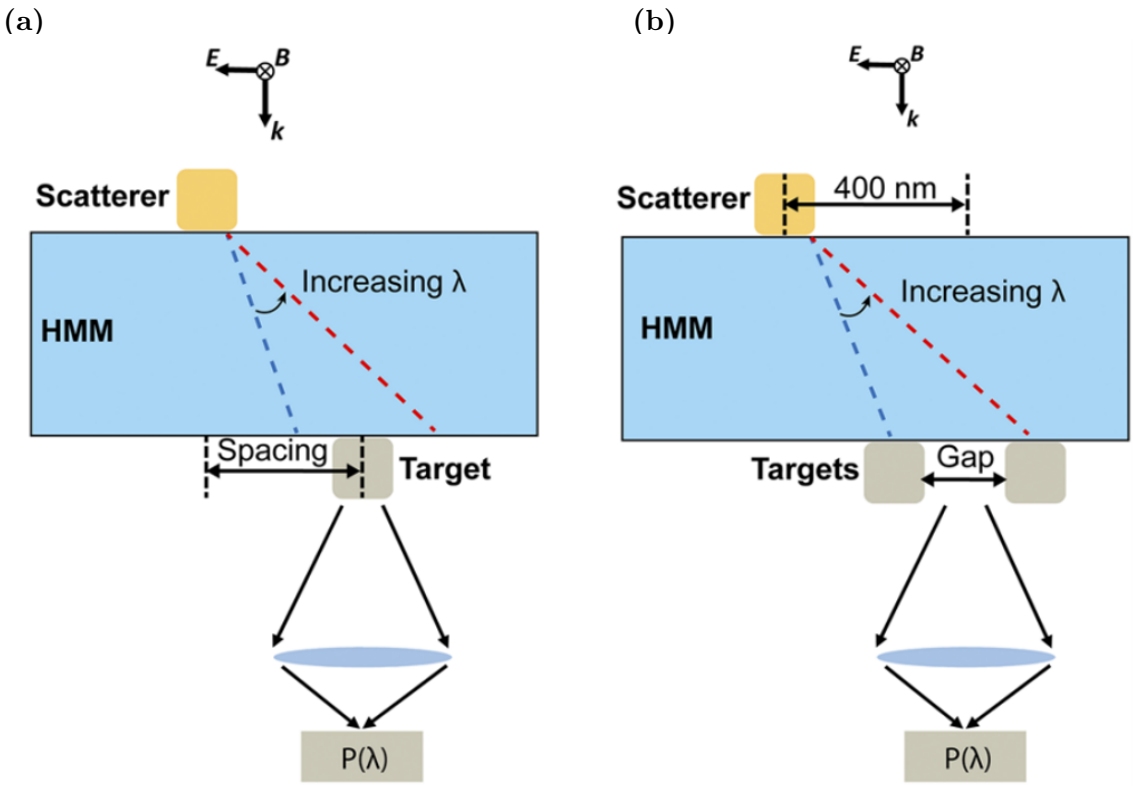

Figure 2.3. Two possible device configurations. Dashed lines show the change in the beam direction as the wavelength is increased from shorter (blue) to longer (red). A photodetector measures the scattered power P(λ).

as the spacing is increased, as expected from the wavelength dependence of the beam angle. Since objects at different spacing differ in their peak position, this spectral curve serves as the "fingerprint" for us to identify the target's location.

Next consider the case in which two targets (again with the refractive index n = 1.73) are placed below the HMM with a gap between them, as shown in Fig. 2.3(b). Fig. 2.4(b) shows the scattering strength versus wavelength for three different gap sizes. For a gap size of 60 nm, two peaks are clearly visible and the peak at shorter wavelength is due to the scattering from the target object closer to the top scatterer. For a gap size of 20 nm, two targets are close enough that they are just barely resolved. For a gap size of 0 nm, only one peak is present as expected. Again, the spectral shape encodes the bottom targets' spatial position and serves as the "fingerprint" for us to determine the gap, which is deep-subwavelength in size.

We next consider the case in which two bottom targets in the configuration of Fig. 2.3(b) consist of different materials. Since the scattered power increases as the refractive index contrast of the target to the air is increased (see additional discussion

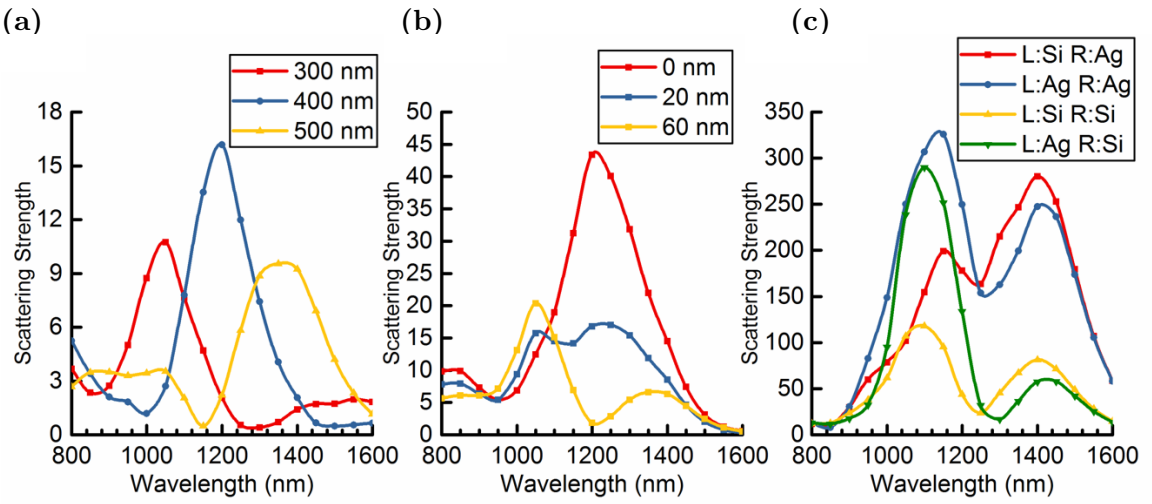

Figure 2.4. (a): Scattering strength versus wavelength for different nanoparticle spacing [see configuration in Fig. 2.3(a)]. n = 1.73 for the bottom target. (b): Scattering strength versus wavelength for different gap sizes [see configuration in Fig. 2.3(b)]. n = 1.73 for both the targets. (c) Scattering strength versus wavelength for four different target material combinations at gap = 100 nm [see configuration in Fig. 2.3(b). L, left target; R, right target].

in appendix section A.4), the magnitude of the spectral peak contains material information about the targets and helps to identify the target's material composition, in addition to their spatial information during the "fingerprinting" process. Four material combinations using Ag and Si are simulated at a fixed gap size of 100 nm to show that a calibrated measurement can in principle distinguish their material composition information in addition to the spatial information, as shown in Fig. 2.3(c). Two peaks can be seen for each material combination in the plot, indicating that there is a resolvable gap (100 nm) between two bottom targets. In addition, each curve has its unique spectral shape, which can be used for "fingerprinting" to determine material information. Finally, we show in the appendix section A.5 that changing the bottom target shape does not change the scattering strength significantly. As a result, the proposed "fingerprinting" process is robust to unintended small structure variations of the bottom target.

2.4 COMSOL implementation

We used the COMSOL for the device simulation and the details are described below. The scattered field of our device is calculated in a two step manner: In the first step, a simulation is done without the top scatterer and bottom target(s) (i.e.

only the HMM structure is present) to get the background field. In this step, periodic boundary condition is used on left/right boundaries. The port condition is used on the top/bottom boundaries. In the second step, the top scatterer and bottom target(s) are introduced into the model. The background field obtained from the first step is used as excitation to calculate the scattered field. In this step, absorbing boundary condition (referred to as the scattering boundary condition in COMSOL) is used in all boundaries.

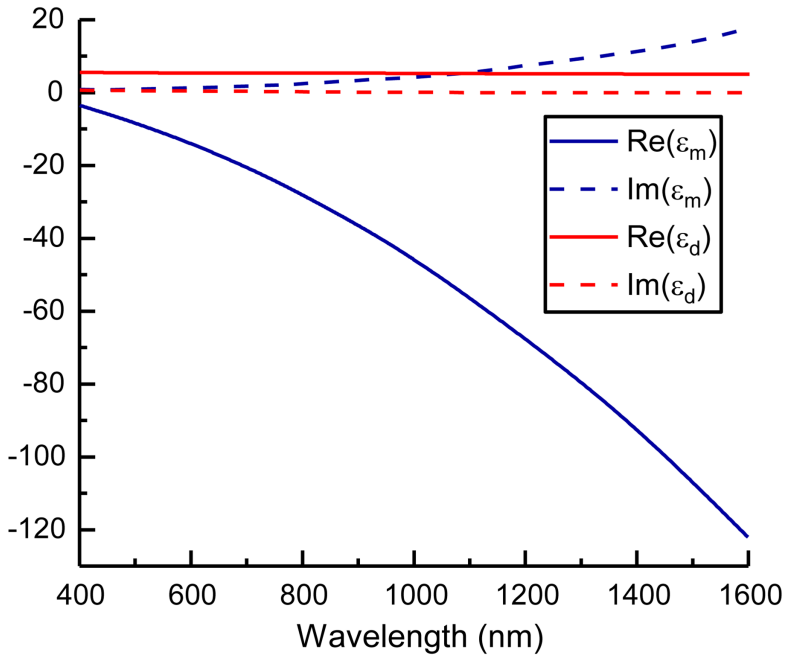

Figure 2.5. The permittivity values of the metal (ε_{m}) and dielectric (ε_{d}) layer

The permittivity values of the metal (Al-doped Ag) and dielectric (Ta_2O_5) layers used in simulation are acquired through spectroscopic ellipsometry characterization of Al-doped Ag and Ta_2O_5 films. The Al-doped silver film was prepared by Al and Ag co-sputtering, whose details are described in previous publications [113, 112]. The Ta_2O_5 layer was prepared by radio frequency (RF) magnetron sputtering of Ta_2O_5.

The measured permittivity values are plotted in Fig. 2.5. The permittivity values

fitted by polynomial functions are given by:

$$\begin{aligned}
&\mathrm{Re}\,(\varepsilon_m) = -5\times10^{-5}\lambda^2 - 0.0047\lambda + 5.7457,\\
&\mathrm{Im}\,(\varepsilon_m) = 6\times10^{-9}\lambda^3 - 4\times10^{-6}\lambda^2 + 0.0027\lambda - 0.1546,\\
&\mathrm{Re}\,(\varepsilon_d) = -6\times10^{-15}\lambda^5 + 3\times10^{-11}\lambda^4 - 6\times10^{-8}\lambda^3 + 6\times10^{-5}\lambda^2 - 0.0253\lambda + 9.8605,\\
&\mathrm{Im}\,(\varepsilon_d) = 2\times10^{-10}\lambda^3 + 2\times10^{-7}\lambda^2 - 0.0015\lambda + 1.2267,
\end{aligned} \tag{2.2}$$

where ε_m and ε_d are respectively the permittivity of metal and dielectric layers, λ is the wavelength of light in unit of nm.

2.5 Deviation from effective medium approximation

We will now return to the cause of the beam angle difference between the EMT calculation and the exact simulation, as observed in Fig. 2.2(b). This difference is due to the finite thickness of the metal/dielectric layers composing the HMM, which leads to a deviation in the exact iso-frequency curve from the EMT case, where the structure is assumed to be homogeneous. This deviation due to the finite thickness of the composing layers has been previously reported [68]. As the ratio wavelength/unit cell size of the HMM structure becomes smaller, the EMT approximation should improve. This may be seen in Fig. 2.2(b), in which the beam angle versus wavelength /unit cell size using exact simulation at different wavelengths is plotted (solid lines). Note that the beam angle obtained via exact simulation (solid lines) approaches the value obtained from the EMT (dashed lines) as the unit cell size is reduced, and it is very close to the EMT result when an extremely small unit cell size (wavelength/unit cell size = 400) is used. This also reveals that the device operates in a very large $k_\perp$ regime: the excited high k-modes in HMM require a very small unit cell size for the EMT to be accurate. It is worth noting that in the proposed 2D structure, the scattered field is transverse-magnetic (TM) polarized and couples to the extraordinary wave (e-wave) of hyperbolic dispersion in HMM under TM excitation. There is no transverse electric (TE) polarized component coupling to the ordinary wave (o-wave) in HMM. However, in a more realistic 3D geometry, the scattered field will have both TE and TM components and the transmission of TE components through HMM may reduce the signal-to-noise ratio on the measurement due to an increased background [27]. We give an estimate of the noise magnitude compared to the signal in the 3D geometry in the next section.

2.6 3D geometry noise analysis

In the more practical applications with 3D geometry, the transmission of TE polarized components through HMM will act like noise in the measurement. In this section, we estimate the noise magnitude compared to the signal. In a 3D geometry, the TE polarized scattered field from the top scatter will couple to ordinary modes (o-wave) in the HMM, which have dispersion relation: $k_\perp^2 + k_\parallel^2 = \varepsilon_\perp \frac{\omega^2}{c^2}$ ($\varepsilon_\perp < 0$). Here $k_\parallel$ and $k_\perp$ are respectively the components of wave vector parallel and perpendicular to optical axis, ω is the angular frequency of wave and c is the speed of light. Since $\varepsilon_\perp < 0$, all ordinary modes in HMM are decaying. Denoting the ordinary mode electric field E $\sim e^{ik_\perp r_\perp} e^{ik_\parallel z} = e^{ik_\perp r_\perp} e^{-\alpha z}$, then $\alpha^2 = k_\perp^2 - \varepsilon_\perp \frac{\omega^2}{c^2}$, according to the dispersion relation. It can be seen from the expression that how fast these modes decay (how large α is) depends on the magnitude of $k_\perp$, which can be divided into two groups, low-$k_\perp$and high-$k_\perp$:

1. Low-$k_\perp$ o-waves: From $\alpha^2 = k_\perp^2 - \varepsilon_\perp \frac{\omega^2}{c^2}$, Low-$k_\perp$ o-waves have small α and hence are not attenuated much as their decay length ($1/\alpha$) is larger than the thickness of the HMM. For a subwavelength scatterer on top of the proposed structure, however, the total power of the low-$k_\perp$ waves is only a small fraction of the total power of all o-waves. So this is a small fraction of the scattered power and can be tolerated as a low-power noise, with the total power that is $\sim \frac{a}{\lambda_0}$ fraction of the signal power, where a is the characteristic subwavelength scale of the target and λ_0 is the free space wavelength (assuming the HMM is surrounded by free space).

2. High-$k_\perp$ o-waves: Most of the scattered power among o-waves are from high-$k_\perp$ o-waves. Initially (i.e., before propagation through the HMM), their total power is comparable to the total power of light scattered into the extraordinary modes (e-wave). However, for these high-$k_\perp$ o-waves, their decay length L is on the order of $1/k_\perp$ (since $\alpha^2 = k_\perp^2 - \varepsilon_\perp \frac{\omega^2}{c^2} \approx k_\perp^2$ for high $k_\perp$), which is much smaller than the thickness of the HMM. Therefore, the transmission of these high-$k_\perp$ o-waves will be negligible.

Considering both i) and ii), we find only small noise background due to the o-waves in 3D geometry, which is much smaller than our signal, with the effect of the o-waves on the SNR $\sim \frac{a}{\lambda_0}$.

2.7 Summary

In summary, we have demonstrated a novel concept based on HMM that is able to discriminate nanostructures. The proposed "fingerprinting" process can be used to determine the location of a single target with deep-subwavelength accuracy, resolve a nanoscale gap between two targets, and obtain information on the target material. Similar devices working in other wavelength ranges can be achieved by appropriate modifications of the HMM design.

This work could potentially find applications in metrology or in biomolecular measurements. For example, determining the separation between two biomolecules with nanoscale separation is important and can be used to study interaction between proteins, such as dimerization of motor proteins $soppina2014dimerization$ or association of regulator proteins [60]. Foster resonance energy transfer [82] (FRET) methods are commonly used for this purpose; however, they are typically applicable when the separation is in the range $1 - 10$ nm, and it is challenging to provide an absolute distance estimation between molecules in FRET because the energy transfer and fluorescence process are sensitive to the environment and orientation of the dyes [89] and photobleaching of the dye molecules hinders distance estimation for a long period of time. On the other hand, our proposed method does not have the aforementioned limitations and works best in the separation range of tens of nanometers, which is hard to reach using FRET.

CHAPTER III

Learning Based Light Field Reconstruction

3.1 Introduction

This chapter presents the application of using the focal stack camera for light field reconstruction. Although there already exists light field cameras that directly capture a light field of a scene (chapter I), the use of the micro-lens array in such light field cameras for multiplexing inevitably leads to a spatial-angular resolution trade-off in the captured light field. For a 4D light field $l(x, y, u, v)$, where x, y are the spatial coordinates and u, v are the angular coordinates, one sees that for a fixed resolution image sensor, a higher image spatial resolution then forces a lower angular resolution and vice versa. For example, the light field by Lytro Illum camera has a resolution of $376 \times 541 \times 14 \times 14$ [104] and the low spatial resolution is due to the above spatial-angular resolution trade-off. Motivated by this, it is worth investigating whether it is possible to reconstruct the light field of a scene from its focal stack measurement, in which case there is no such sptial-angular resolution trade-off.

Reconstructing a signal from its indirect measurement is called inverse problem and has been studied extensively in medical imaging. An example is CT scan, where X-ray sensor measures the transmitted X-ray with known intensity through the patient body. The process is repeated with different X-ray directions and from the collected sensor measurements, algorithms can be used to reconstruct the 3D internal structure of the patient body.

In many cases, the inverse problem can not be solved directly and exactly because the inverse problem is ill-posed or the measurement is corrupted heavily by noise. Instead, an model-based image reconstruction (MBIR) is a more suitable choice that has a better reconstruction quality, at the cost of increased computation time due to its iterative nature. MBIR typically have five components [26], which are outlined as follows:

- A object model that expresses the continuous signal to be reconstructed in a basis with finitely many unknown coefficients.
- A system model that relates the signal to the "ideal" measurement in the abscence of noise. This is typically in the form of a linear model as: $y = Ax$, where A is the system model matrix, x is the signal to be reconstructed, y is the measurement.
- A statistical model that describes how the noisy measurement vary around the expectation. Often a Gaussian noise or Poisson statistics are assumed.
- A cost function to be minimized to find the unknown signal coefficients.
- A algorithm, typically iterative, for minimizing the cost function.

For light field reconstruction, Lien et al. has already built the system model and successfully demonstrated light field reconstruction from focal stack using MBIR with 4D Edge Preserving (EP) regularizer [54]. Since the light field contains low rank structure as exhibited in its EPI, Blocker et al. proposed to use a low-rank plus sparse regularization term in the cost function to improve the reconstruction quality. However, the regularizer in these methods are all hand-crafted, which is not guaranteed to be optimal. With the recent advancement in deep learning, there have been many works incorporating deep learning techniques into the image reconstruction task, achieving improved performance [98, 3, 16, 13, 19, 20]. This motivates us to develop a learning based method that is suitable for light field reconstruction, which will be described next. The work in section 3.2 is published in *IEEE Transaction of Pattern Recognition and Machine Intelligence* [18] and the work in section 3.3 is published in *The international Conference on Acoustics, Speech, and Signal Processing* [39].

3.2 Iterative Neural Networks for Light Field Reconstruction

Light field reconstruction is to estimate a 4D light field x from focal stack measurement y, which can be done by solving the following optimization problem:

$$\underset{x}{\operatorname{argmin}} \; F(x;y), \quad F(x;y) \triangleq \frac{1}{2}\|Ax-y\|_2^2 + R(x), \tag{3.1}$$

where A is the imaging model matrix, $\frac{1}{2}\|Ax-y\|_2^2$ is the data fidelity term that ensures the reconstructed signal x is consistent with the measurement y, and $R(x)$ is

the regularization term that contains prior information about x. This optimization problem is under-determined and a good regularizer is critical to ensure good reconstruction. Conventional methods use some handcrafted regularizers, such as total variation regularization and edge-preserving regularization [11]. However, these handcrafted regularizers are by no means optimal, which lead to recent works of learning based image reconstruction.

Iterative Neural Network (INN) method is one way to incorporate deep learning into image reconstruction. It combines denoising neural networks with an unrolled iterative MBIR algorithm [98, 16, 20]. Compared to methods that directly use a neural network to regress from the raw measurement to reconstructed image [122], it reduces the risk of network overfitting issue, by balancing the data-fitting term and the reguarlization term in the MBIR.

ADMM-Net [98] was the first to propose unrolling an iterative optimization algorithm (ADMM [12]) to solve inverse problems. It replaces the tunning parameters in ADMM with learnable parameters. It is trained end to end and shows significantly improves the reconstruction accuracy over baseline ADMM method. Similarly, PD-Net [3] proposed to unroll the primal-dual algorithm [123] and BCD-Net [20] is obtained by unrolling the block coordinate descent algorithm [77]. However, the convergence and acceleration of these methods remains to be a challenge.

3.2.1 Momentum-Net structure

This section proposes Momentum-Net, a fast and convergent iterative neural network for inverse problems. It's constructed by unrolling the iterative block proximal extrapolated gradient method (BPEG-M [17]) and incorporating learned regularizer into it. To accelerate the convergence, it contains an extrapolation step (momentum term) in the iterative updating process and hence its name, Momentum-Net.

Fig. 3.1 and Algorithm 1 show the structure of the Momentum-Net. Each iteration of Momentum-Net consists of 1) image refining, 2) extrapolation, and 3) Model-Based Image Reconstruction (MBIR) modules. At the i-th iteration, Momentum-Net performs the following three processes:

- *Refining*: The i-th image refining module gives the refined image $z^{(i+1)}$, by applying the i-th refining Neural Network (NN) , $R_{\theta^{(i+1)}}$, to an input image at the i-th iteration, $x^{(i)}$ (i.e., image estimate from the $(i-1)$-th iteration). We apply ρ-relaxation with $\rho \in (0, 1)$; The parameter ρ controls the strength of inference from refining NNs, but does not affect the convergence guarantee of

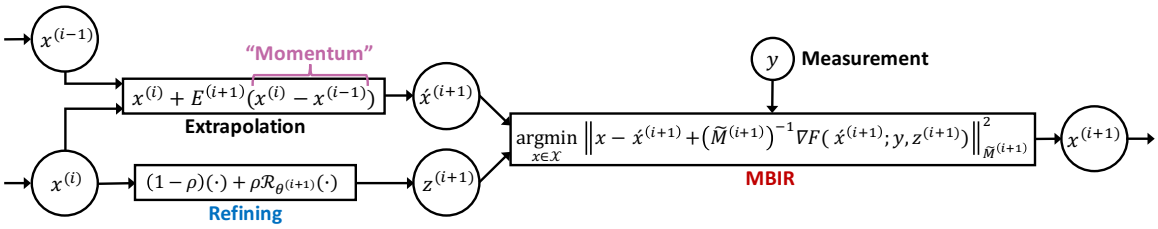

Figure 3.1. The architecture of the Momentum-Net, showing its updating rules at the i-th iteration.

Algorithm 1: Momentum-Net for Light Field Reconstruction

Require: $\{\mathcal{R}_{\theta^{(i)}} : i = 1, \ldots, N_{\text{iter}}\}, \rho, \delta \in (0,1), \gamma > 0, \beta^0 = 1, x^{(0)} = x^{(-1)}, y$

for $i = 0, \ldots, N_{iter-1}$ **do**

Calculate $\widetilde{M}^{(i+1)}$ by (3.5) and $E^{(i+1)}$ by (3.2) and (3.3)

Image refinining:

$$z^{(i+1)} = (1-\rho)x^{(i)} + \rho \mathcal{R}_{\theta^{(i+1)}}\left(x^{(i)}\right)$$

Extrapolation:

$$\acute{x}^{(i+1)} = x^{(i)} + E^{(i+1)}\left(x^{(i)} - x^{(i-1)}\right)$$

MBIR:

$$x^{(i+1)} = \text{Prox}^{\widetilde{M}^{(i+1)}}_{\mathbb{I}_{\mathcal{X}}}\left(\acute{x}^{(i+1)} - \left(\widetilde{M}^{(i+1)}\right)^{-1} \nabla F\left(\acute{x}^{(i+1)}; y, z^{(i+1)}\right)\right)$$

end

Momentum-Net. Proper selection of ρ can improve MBIR accuracy.

- *Extrapolation*: The i-th extrapolation module gives the extrapolated point $\acute{x}^{(i+1)}$, based on *momentum* terms $x^{(i)} - x^{(i-1)}$;. Intuitively speaking, momentum provides information from previous updates to amplify the changes in subsequent iterations. The extrapolation coefficient is given by:

$$E^{(i)} = \delta^2 m^{(i)}, \tag{3.2}$$

and the momentum coefficients $m^{(i)}$ are updated by the following formula [17]:

$$m^{(i+1)} = \frac{\beta^{(i)} - 1}{\beta^{(i+1)}}, \quad \beta^{(i+1)} = \frac{1 + \sqrt{1 + 4\left(\beta^{(i)}\right)^2}}{2}, \tag{3.3}$$

- *MBIR*: This step solves a *majorized* version of the following MBIR problem at

the extrapolated point $\acute{x}^{(i+1)}$:

$$\min_x F(x; y, z^{(i+1)}), \tag{3.4}$$

with F defined in 3.1, using the majorization matrix:

$$\widetilde{M}^{(i+1)} = \text{diag}(A^T A1) + \gamma I. \tag{3.5}$$

This step gives a reconstructed image $x^{(i+1)}$ and is used as the input to the next Momentum-Net iteration.

3.2.2 Benefits of Momentum-Net

Momentum-Net has several benefits over existing INNs:

1. *Benefits from refining module*: The image refining module can use iteration-wise image refining NNs $\{\mathcal{R}_{\theta^{(i+1)}} : i \geq 0\}$: those are particularly useful to reduce overfitting risks by reducing dimensions of their parameters $\theta^{(i+1)}$ at each iteration [20]. Iteration-wise refining NNs require less memory for training, compared to methods that use a single refining NN for all iterations, e.g., [30].

2. *Benefits from extrapolation module*: The extrapolation module uses the momentum terms $x^{(i)} - x^{(i-1)}$ that accelerate the convergence of Momentum-Net. In particular, compared to the existing gradient-descent-inspired INNs, e.g., TNRD [16], Momentum-Net converges faster.

3. *Benefits from MBIR module*: The MBIR module does not require multiple inner iterations for a wide range of imaging problems and has both theoretical and practical benefits. Different from the existing BCD-Net-type methods [98, 13, 87, 116] that can require iterative solvers for their MBIR modules, MBIR module of Momentum-Net can have practical close-form solution, and its corresponding convergence analysis can hold stably for a wide range of imaging applications. Second, combined with extrapolation module, noniterative MBIR modules lead to faster MBIR, compared to the existing BCD-Net-type methods that can require multiple inner iterations for their MBIR modules for convergence. Third, Momentum-Net guarantees convergence even for nonconvex MBIR cost function $F(x; y, z)$ or nonconvex data-fit $f(x; y)$ of which the gradient is Lipschitz continuous, while existing INNs overlooked nonconvex $F(x; y, z)$

or $f(x;y)$. Detailed analysis on convergence of Momentum-Net can be found in [18].

3.2.3 Momentum-Net experimental setup

To reconstruct a light field, the MBIR problem considers a data fidelity term $f(x;y)=\frac{1}{2}\|y-Ax\|_2^2$ and a box constraint $\mathcal{X}=[0,1]$, where A is the system matrix of light field imaging system. In general, a light field photography system using a focal stack is extremely under-determined, because the system matrix A is a wide matrix.

To avoid the inverse crime [107], our imaging simulation used higher-resolution synthetic light field dataset [38] (we converted the original RGB sub-aperture images to grayscale ones by the `rgb2gray` function in MATLAB, for simplicity and smaller memory requirements in training). We simulated $n_{\mathrm{F}}=5$ focal stack images of size 255×255 with 40 dB Additive White Gaussian Noise (AWGN) that models electronic noise at sensors. The sensor positions were chosen such that five sensors focus at equally spaced depths; specifically, the closest sensor to scenes and farthest sensor from scenes focus at two different depths that correspond to $\texttt{disp}_{\min}+0.2$ and $\texttt{disp}_{\max}-0.2$, respectively, where $\texttt{disp}_{\max}$ and $\texttt{disp}_{\min}$ are the approximate maximum and minimum disparity values specified in [38]. We reconstructed 4D light fields of resolution $255\times255\times9\times9$.

We used Convolutional Neural Networks (CNN) as the refining neural networks in the *Image refining* step of Algorithm 1. We used either a shallow 3-layer CNN (sCNN) or a deep 6-layer CNN (dCNN), with a residual connection [36, 115] to refine the input light field. We chose to refine the light field in the EPI domain. Specifically, the input light field is sliced to two sets of horizontal EPIs and vertical EPIs. Each EPI was refined by the CNN refiner. We then took the average of two light fields that were permuted back from refined horizontal and vertical EPI sets. We found that refining the light field in the EPI domain performs better than refining the light field in the subaperture image domain. This is likely because the image structure in the EPI domain is much simpler, hence easier to refine.

We trained image refining NNs at the $(i+1)$-th iteration $\mathcal{R}_{\theta^{(i+1)}}$ by minimizing the following loss:

$$L=\frac{1}{2S}\sum_{s=1}^{N}\left\|x_s-\mathcal{R}_{\theta^{(i+1)}}\left(x_s^{(i)}\right)\right\|_2^2, \tag{3.6}$$

where x_s is the ground truth signal to be reconstructed, $x_s^{(i)}$ is the reconstructed signal by the Momentum-Net at the (i)-th iteration, N is the total number of training

samples. The networks were trained in Pytorch using Nvidia GTX 1080 Ti. We set the hyperparameters of Momentum-Net as: $N_{\text{iter}} = 100$, $\gamma = 1$, $\rho = \delta = 1 - \varepsilon$, where ε is the machine epsilon. The initial reconstruction $x^{(0)}$ is set to $A^T y$ rescaled to the interval $[0, 1]$ (i.e., dividing by its max value). The parameters of the first refiner network, $\mathcal{R}_{\theta^{(1)}}$, was initialized with Kaiming uniform initialization [35]; Refiner networks in the later iteration, i.e., at the i-th INN iteration, for $i \geq 2$, are initialized from those learned from the previous iteration, i.e., $(i-1)$-th iteration.

We tested trained INNs to three samples of which scene parameter and camera settings are different from those in training samples (all training and testing samples have different camera and scene parameters). We evaluated the reconstruction quality using peak signal-to-noise ratio (PSNR). In addition, we compared the trained Momentum-Net to MBIR method using the state-of-the-art non-trained regularizer, 4D EP introduced in [54]. (The low-rank plus sparse tensor decomposition model [11] failed when inverse crimes and measurement noise are considered.) We finely tuned its regularization parameter to achieve the lowest RMSE values.

We further investigated impacts of the light field MBIR quality on a higher-level depth estimation application, by applying the robust Spinning Parallelogram Operator (SPO) depth estimation method [117] to reconstructed light fields.

3.2.4 Momentum-Net results

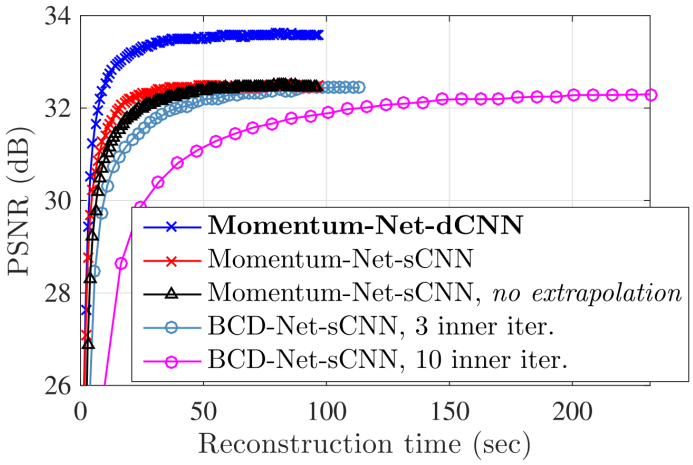

Figure 3.2. PSNR maximization comparisons between different INNs (Light field photography system with $n_{\text{F}} = 5$ detectors obtain a focal stack of light fields consisting of $S = 81$ sub-aperture images; averaged PSNR values across three test reconstructed images).

Fig. 3.2 compares the proposed Momentum-Net with existing INN method, i.e.,

BCD-Net [20]. It shows that to reach the 32 dB PSNR value in light field reconstruction from a focal stack, the proposed Momentum-Net using sCNN (red) decreases MBIR time by 36.5% and 61.5%, compared to Momentum-Net without extrapolation (black) and BCD-Net using three inner iterations (light blue), respectively. It also shows that using a deep CNN model (blue) performs better than a shallow CNN model (red). Note that using dCNN refiners instead of sCNN refiners has a negligible effect on total run time of Momentum-Net, because reconstruction time of MBIR modules (in CPUs) dominates inference time of image refining modules (in GPUs).

Fig. 3.3 shows the comparison of the light field reconstruction quality, using state-of-the-art non-trained 4D EP regularizer introduced in [11] (Fig. 3.3(b)), using Momentum-Net with shallow 3-layer sCNN (Fig. 3.3(c)), and using Momentum-Net with deeper 6-layer dCNN (Fig. 3.3(d)). Proposed Momentum-Net with deep CNN as refining network achieves the best reconstruction quality. We further used the reconstructed light field for downstream depth estimation task and evaluated their depth RMSE. Fig. 3.4 shows the proposed Momentum-Net also leads to the best final depth estimation accuracy.

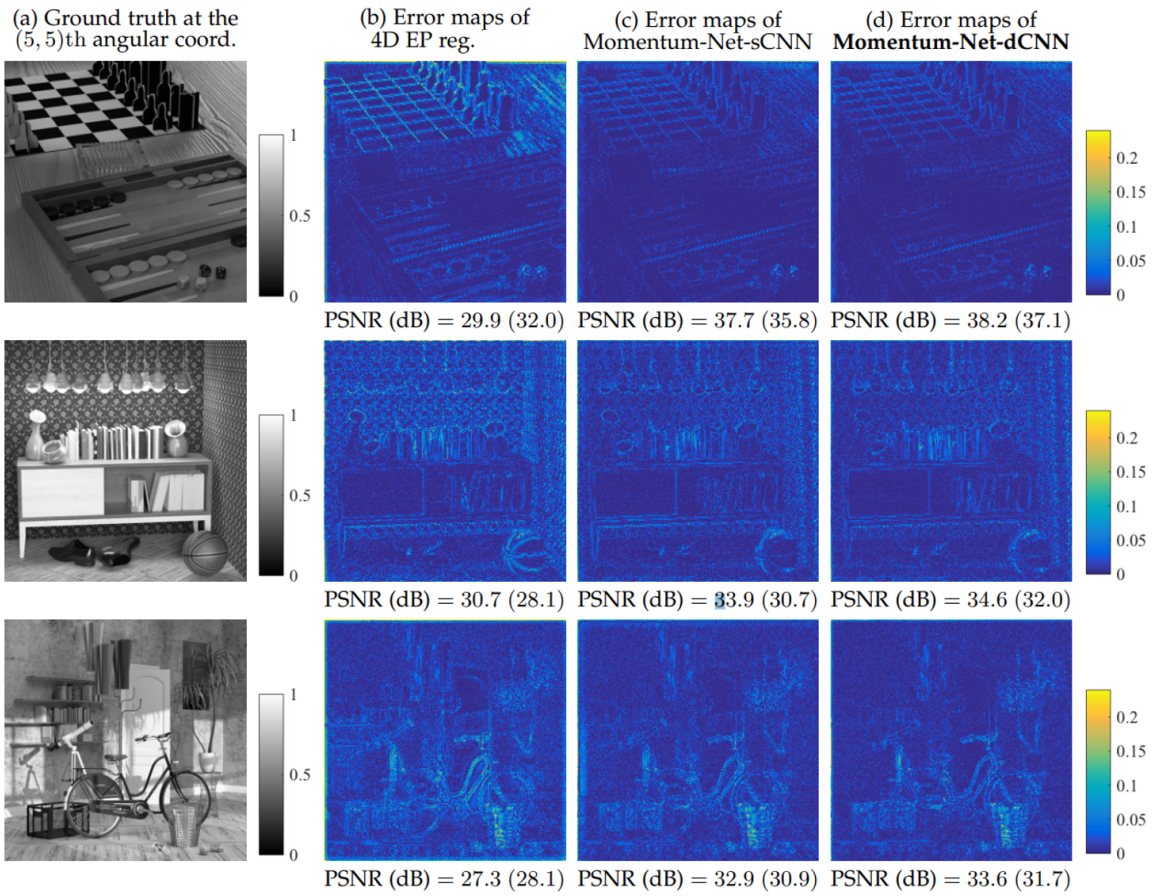

Figure 3.3. Error map comparisons of reconstructed sub-aperture images (at the angular coordinate $(5, 5)$) from different MBIR methods. The PSNR values in parenthesis were measured from reconstructed light fields.

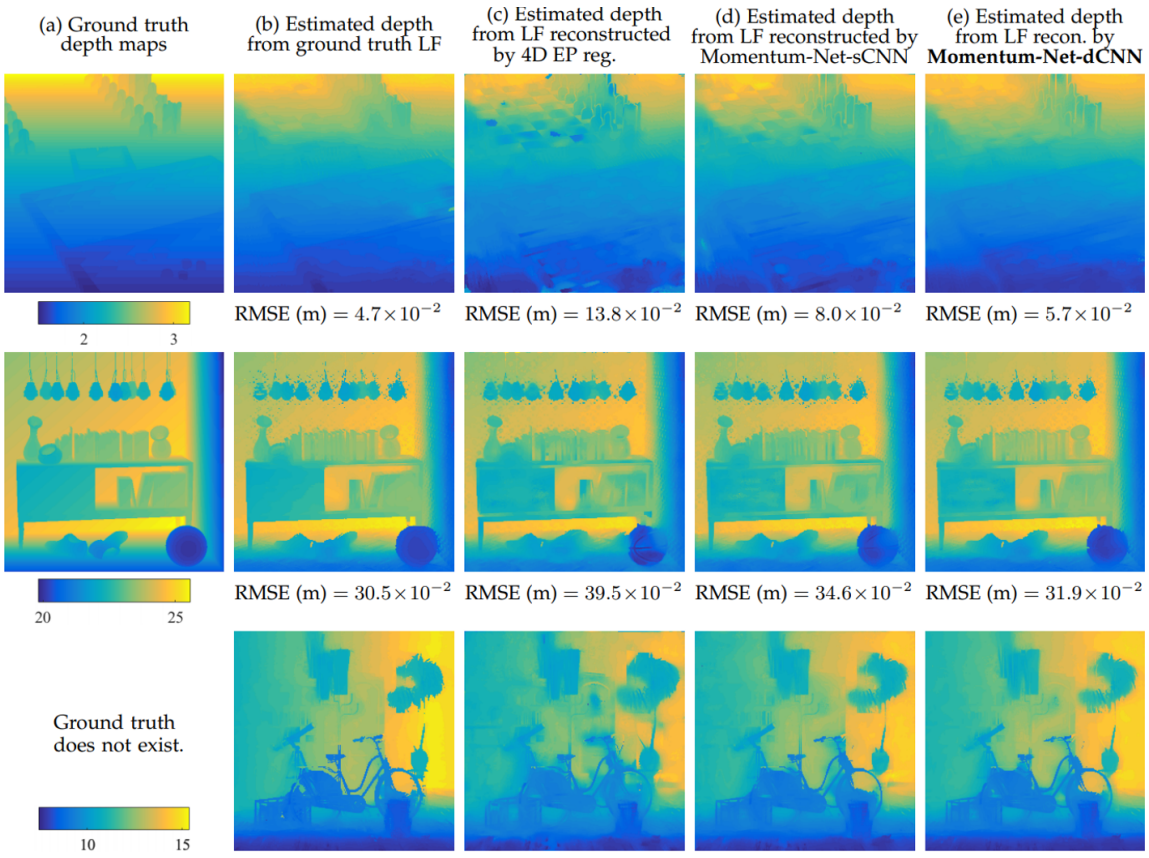

Figure 3.4. Comparisons of estimated depths from light fields reconstructed by different MBIR methods. SPO depth estimation [117] was applied to reconstructed light fields

3.3 Non-iterative neural networks for light field reconstruction

The light field reconstruction methods described in the previous section, are all iterative in nature, which are typically slow. This section proposes a non-iterative light field reconstruction and depth estimation method based on sequential CNNs.

CNN methods are rapidly emerging as a powerful tool for various image processing and computer vision tasks due to their ability to model complicated functions and short inference time. Prior works have applied CNN method to light field view synthesis [46, 97, 101]. The most relevant work is from Srinivasan et al [97]. They proposed a sequential CNN approach that reconstructs a light field from a single all-in-focus image. Their pipeline consists of a CNN that estimates ray depth of the scene from the input all-in-focus image, a rendering module that renders a Lambertian light field using the estimated ray depth, and a second CNN that corrects the artifacts in the rendered Lambertian light field. As their method uses ray depth to render a light field, light field reconstruction quality largely depends on the quality of the estimated ray depth. However, depth estimation from single image is challenging as it lacks reliable depth cues. On the other hand, depth from focal stack images is more accurate and could greatly benefit light field reconstruction.

Motivated by [101], our proposed method uses three sequential CNNs. The first CNN estimates an all-in-focus image from focal stack images; the second CNN estimates 4D ray depth from focal stack images and the estimated all-in-focus image; a rendering module renders a Lambertian light field with the estimated all-in-focus image and ray depth, and the third CNN subsequently refines the rendered light field and provides the final reconstructed light field. Numerical experiments show that the proposed method significantly improves light field reconstruction accuracy, compared with a state-of-the-art sequential CNN approach using a single all-in-focus image [97], conventional MBIR using 4D edge-preserving (EP) regularizer (from a focal stack) [11], and direct regression CNN from a focal stack. In addition, the proposed method considerably reduces light field reconstruction time compared with MBIR using EP regularizer.

3.3.1 Algorithm for non-iterative light field reconstruction

The proposed approach uses four steps to reconstruct light fields, as illustrated in Fig. 3.5. In the first step, an "all-in-focus image synthesis" neural network (NN) synthesizes an all-in-focus image from a focal stack (section 3.3.1.1). In the second step,

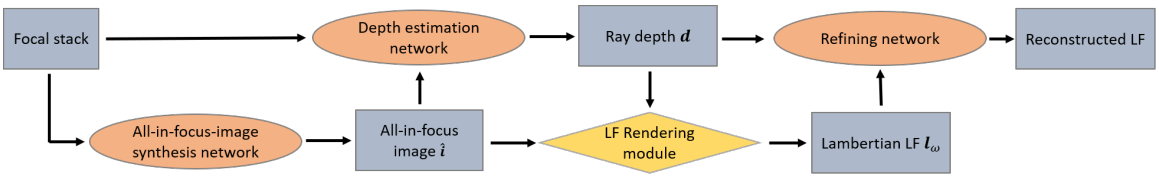

Figure 3.5. Proposed CNN-based method for light field reconstruction and depth estimation using focal stack.

a depth estimation NN estimates 4D ray depth $\boldsymbol{d}$ (depth maps for every view point) from the estimated all-in-focus image $\hat{\boldsymbol{i}}$ and focal stack images $\boldsymbol{f}$ (section 3.3.1.2). The third step renders a Lambertian light field $\boldsymbol{l}_w$ by backward warping the all-in-focus image $\hat{\boldsymbol{i}}$, using the estimated 4D ray depth $\boldsymbol{d}$ (section 3.3.1.3). Because the rendered light field is Lambertian and may contain artifacts around occlusions, we use a refining NN to further refine $\boldsymbol{l}_w$ and obtain a final light field $\boldsymbol{l}^\star$ (section 3.3.1.4). The following subsections describe details of each step.

3.3.1.1 All-in-focus image synthesis NN

We first estimated an all-in-focus image, given the focal stack images; this process is called focal stacking. There are several focal stacking approaches, e.g., edge detection, Fourier analysis, and CNN. Among these, we choose CNN-based method – specifically, U-Net [88] with modified input and output channel numbers – due to its good image mapping capability. We forward pass reshaped focal stack images (from the size $C \times N_F \times H \times W$ to $(C \cdot n_{\mathrm{F}}) \times H \times W$, where C is the number of color channel, n_{F} is the number of focal planes in the focal stack, H and W are the image height and width, respectively) through the modified U-Net. To squeeze the output all-in-focus image to be within the interval $[0, 1]$, we put a differentiable nonlinear function $g(\cdot) = (\tanh(\cdot)+1)/2$ at the end of the U-Net. We train the modified U-Net, $\mathcal{A}_{\theta_a}(\boldsymbol{f})$, having parameter set θ_a, by minimizing the ℓ_1 loss:

$$\min_{\theta_a} \sum_n \|\mathcal{A}_{\theta_a}(\boldsymbol{f}_n) - \boldsymbol{i}_n\|_1,$$

where $\{\boldsymbol{f}_n : \forall n\}$ are training focal stack images, and $\{\boldsymbol{i}_n : \forall n\}$ are the ground truth all-in-focus images. We use the center sub-aperture images of the ground truth light fields for $\{\boldsymbol{i}_n\}$, because sub-aperture images of light fields have small enough aperture such that all regions of the image are well in focus.

3.3.1.2 Depth estimation NN

The light field rendering (section 3.3.1.3) uses both an all-in-focus image and a 4D ray depth $d(\boldsymbol{x}, \boldsymbol{\nu})$, i.e., a collection of 2D disparity maps, one for each angular coordinate $\boldsymbol{\nu}$. We modified the CNN architecture in [97] to estimate 4D ray depth using focal stack images and the all-in-focus image from $\mathcal{A}_{\theta_a}$. We reshape the input focal stack in the same way as described in section 3.3.1.1. We use dilated convolution layers [110] to have exponentially growing receptive field without losing resolution. At the end of the NN, a tanh scaling layer squeezes the estimated disparity within the range $[-1, 1]$. We jointly train the depth estimation NN and the refining NN (section 3.3.1.4); see training loss in section 3.3.2.

3.3.1.3 Light field rendering

Given the estimated 4D ray depth $\boldsymbol{d}$ and the estimated all-in-focus image $\hat{\boldsymbol{i}}$ via trained $\mathcal{A}_{\theta_a^\star}(\cdot)$, we render a Lambertian light field $\boldsymbol{l}_w$ by backward warping $\hat{\boldsymbol{i}}$ as follows [97]:

$$\begin{aligned} l_w(\boldsymbol{x}, \boldsymbol{\nu}) &= l_w(\boldsymbol{x} + \boldsymbol{\nu} d(\boldsymbol{x}, \boldsymbol{\nu}), \mathbf{0}) = \hat{i}(\boldsymbol{x} + \boldsymbol{\nu} d(\boldsymbol{x}, \boldsymbol{\nu})) \\ &=: \mathcal{W}(\hat{\boldsymbol{i}}, \boldsymbol{d}). \end{aligned} \tag{3.7}$$

We use bilinear interpolation to calculate the values of $\hat{i}(\boldsymbol{x} + \boldsymbol{\nu} d(\boldsymbol{x}, \boldsymbol{\nu}))$ in the warping. As the rendering at a viewpoint $\boldsymbol{\nu}$ given by (3.7) is essentially a sampling of the pixel values at the center view, the rendered light field $\boldsymbol{l}_w$ will be approximately Lambertian and can have artifacts around the occlusion regions.

3.3.1.4 Refining NN

Because the rendered light field $\boldsymbol{l}_w$ from section 3.3.1.3 does not model the non-Lambertian effect and occlusion effect, we use an additional refining NN (see its architecture in [97]) to remove these artifacts and get a final reconstructed light field $\hat{\boldsymbol{l}}$. We use a residual connection [36] for the NN to learn the difference between the Lambertian light field $\boldsymbol{l}_w$ and true light field $\boldsymbol{l}$. We input both estimated 4D ray depth $\boldsymbol{d}$ and Lambertian light field $\boldsymbol{l}_w$ to the NN; in particular, $\boldsymbol{d}$ is useful for predicting occluded region and to refine $\boldsymbol{l}_w$.

3.3.2 Training of depth estimation NN and refining NN

We jointly train the depth estimation NN and the refining NN, similar to [97]. By using differentiable bilinear interpolation for the light field rendering, the loss gradient can be back-propagated from the refining NN, through the light field rendering module, and to the depth estimation NN. Specifically, we jointly train a depth estimation NN, $\mathcal{D}_{\theta_d}(\boldsymbol{f}, \hat{\boldsymbol{i}})$, and a refining NN, $\mathcal{R}_{\theta_r}(\boldsymbol{d}, \boldsymbol{l}_w)$ having parameters θ_d and θ_r, respectively, by minimizing the following loss function:

$$\begin{aligned}\min_{\theta_d,\theta_r} \sum_n \Big\| \mathcal{W}\big(\hat{\boldsymbol{i}}_n, \mathcal{D}_{\theta_d}(\boldsymbol{f}_n, \hat{\boldsymbol{i}}_n)\big) - \boldsymbol{l}_n \Big\|_1 + \\ \Big\| \mathcal{R}_{\theta_r}\Big(\mathcal{D}_{\theta_d}(\boldsymbol{f}_\theta, \hat{\boldsymbol{i}}_n), \mathcal{W}\big(\hat{\boldsymbol{i}}_n, \mathcal{D}_{\theta_d}(\boldsymbol{f}_n, \hat{\boldsymbol{i}}_n)\big)\Big) - \boldsymbol{l}_n \Big\|_1 + \\ \lambda_{\mathrm{c}}\psi_{\mathrm{c}}\Big(\mathcal{D}_{\theta_d}(\boldsymbol{f}_n, \hat{\boldsymbol{i}}_n)\Big) + \lambda_{\mathrm{tv}}\psi_{\mathrm{tv}}\Big(\mathcal{D}_{\theta_d}(\boldsymbol{f}_n, \hat{\boldsymbol{i}}_n)\Big)\,, \end{aligned} \tag{3.8}$$

where the training data consists of focal stack images $\{\boldsymbol{f}_n\}$, estimated all-in-focus images $\{\hat{\boldsymbol{i}}_n = \mathcal{A}_{\theta_a^\star}(\boldsymbol{f}_n) : \forall n\}$ and ground truth light fields $\{\boldsymbol{l}_n : \forall n\}$. In (3.8), ψ_{c} and ψ_{tv} are 4D ray depth consistency and total variation regularizer, respectively, designed to make the estimated ray depth $\boldsymbol{d}$ reasonable [97]. The regularizers are defined by [97]

$$\psi_{\mathrm{c}}(\boldsymbol{d}) := \sum_{\boldsymbol{x},\boldsymbol{\nu}} |d(\boldsymbol{x}, \boldsymbol{\nu}) - d(\boldsymbol{x} + d(\boldsymbol{x}, \boldsymbol{\nu}), \boldsymbol{\nu} - \mathbf{1})| \tag{3.9}$$

$$\psi_{\mathrm{tv}}(\boldsymbol{d}) := \|\nabla_{\boldsymbol{x}} \boldsymbol{d}\|_1. \tag{3.10}$$

As the ray depth consists of depth maps at different viewpoints, these depth maps should be consistent with each other. Specifically, they are related by the equality

$$d(\boldsymbol{x}, \boldsymbol{\nu}) = d(\boldsymbol{x} + \boldsymbol{\Delta} D(\boldsymbol{x}, \boldsymbol{\nu}), \boldsymbol{\nu} - \boldsymbol{\Delta}) \tag{3.11}$$

that is similar to the relation in (3.7). Note that the relation in (3.7) corresponds to a special case of $\boldsymbol{\Delta} = \boldsymbol{\nu}$; choosing $\boldsymbol{\Delta} = \mathbf{1}$ leads to the ray depth consistency regularizer in (3.9), which encourages depths maps at neighboring views to be consistent. On the other hand, the total variation regularizer in (3.10) ensures the estimated depth maps are spatially smooth.

3.3.3 Experimental setup

We compared the proposed method with following three methods: *1)* a state-of-the-art sequential CNN method that estimates 4D ray depth from a single image

and then reconstructs a light field [97]; *2)* a conventional 4D EP MBIR method that reconstructs a light field from focal stack (see, e.g., [11, 18]); *3)* a direct regression CNN from focal stack – we chose a U-Net architecture [88]. For *3)*, a sufficient number of network parameters is chosen such that further increasing the parameter doesn't give better performance.

For all experiments in the paper, we used the light field dataset in [97] that consists of 3343 RGB light fields of flowers and plants taken with Lytro Illum camera. To avoid an inverse crime, we simulated 185×269 focal stack images with number of focal planes $N_F = 7$, from high spatial resolution light fields consisting of 370×538 sub-aperture images on (central) 7×7 angular ($\boldsymbol{\nu}$-) grid. The locations of the seven sensors were chosen to focus at equally spaced disparities in the interval $[-1, 0.3]$. We reconstructed light fields consisting of 185×269 RGB sub-aperture images on the 7×7 $\boldsymbol{\nu}$-grid.

We used the Adam optimizer [48] to train all the NNs compared in the paper. We set the default learning rate as 3×10^{-4}; for training the direct regression CNN, we used 5×10^{-4}. In training the all-in-focus synthesis NN in section 3.3.1.1, we used a batch size of 2 and 40 epochs. We used learning rate scheduling to stabilize the training: the learning rate decays by 0.5 at epochs 3, 6, 10, and 20. For joint training of depth estimation and refining NNs, we used a batch size of 1 and 50 epochs. We chose the regularization parameters in (3.8) as $\lambda_{\text{c}} = \ 0.005$ and $\lambda_{\text{tv}} = 0.01$.

For evaluating the performance of conventional MBIR with 4D EP regularizer, we used the hyperbola penalty function, selected the regularization and hyperpola penalty parameter as 1.6×10^5 and 0.38, respectively, and used conjugate gradient descent method with 30 iterations. We reconstructed each color channel of the light field independently.

3.3.4 Results

Fig. 3.6(a-c) shows an example of reconstructed light field and intermediate estimated depth from the proposed method. The proposed method can reconstruct both ray depth and light field with good quality from a focal stack.

Fig. 3.6(c-d) shows ray depth estimated by the proposed method using focal stack (c) and by sequential CNN using a single image (d). The proposed method can improve depth estimation. As expected, better depth estimate benefits subsequent light field reconstruction: Table 3.1 shows that the proposed method achieves a 4.8 dB peak signal-to-noise ratio (PSNR) improvement over the state-of-the-art sequential CNN using a single image [97]. In addition, the proposed method significantly im-

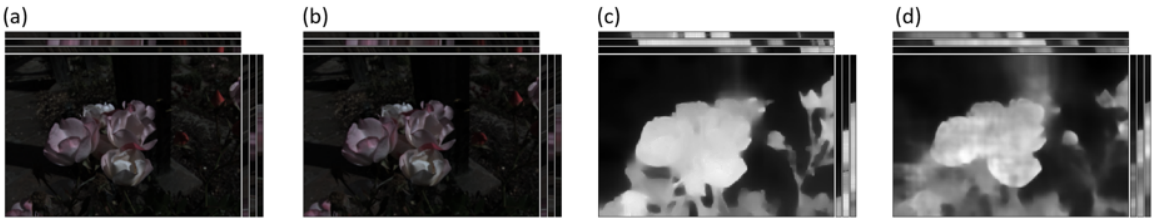

Figure 3.6. Sub-aperture images and epipolar slices of the reconstructed light field and the estimated 4D ray depth. (a) Ground truth light field visualized at the corner view. (b) Reconstructed light field via the proposed method at the corner view (PSNR = 42.23 dB). (c) Estimated center view depth via the proposed method. (d) Estimated center view depth via single image sequential CNN [97].

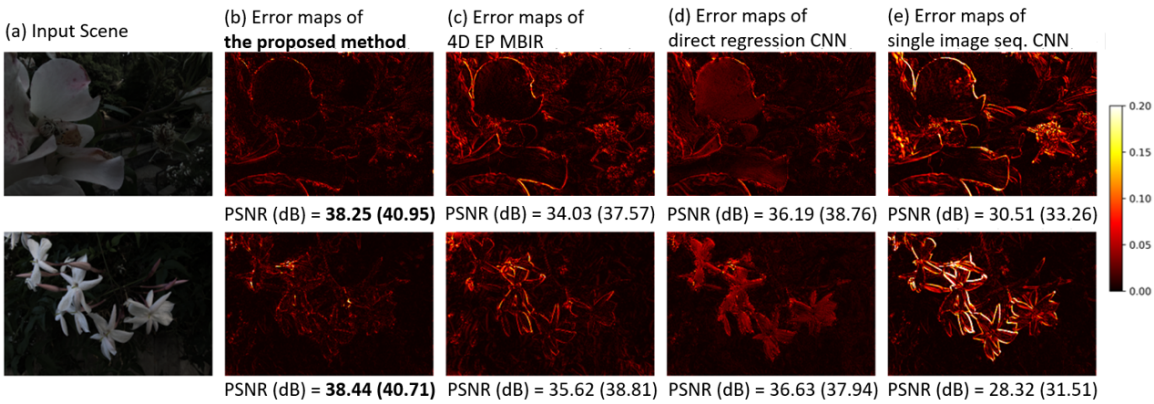

Figure 3.7. Error maps of the reconstructed light field sub-aperture view ($u = -1, v = 3$). The PSNR values shown in parenthesis are calculated from reconstructed light fields.

proves light field reconstruction accuracy compared to other light field reconstruction methods using focal stack images: the proposed method achieves 2.85 dB and 1.97 dB PNSR improvements, over the conventional 4D EP MBIR method and direct regression CNN, respectively; see Table 3.1. Fig. 3.7 shows sub-aperture view error maps of two test light field for all the methods. The error maps of proposed method show significantly reduced error.

The second column of Table 3.1 includes the timing comparison between the proposed method and other methods. In particular, it shows that the proposed method significantly reduces computation time compared to 4D EP MBIR.

3.4 Summary

We have presented learning based methods for light field reconstruction, which shows improved reconstruction performance over traditional MBIR methods. Momen-

Methods	PSNR (dB)	Time (sec.)
Proposed method	**39.76**	4.14 (6.2×10^{-2})
Single image sequential CNN [97]	34.96	3.96 (4.6×10^{-2})
4D EP MBIR	36.91	152 (n/a)
Direct regression CNN	37.79	**0.23** (1.7×10^{-3})

Table 3.1. Average PSNR of the reconstructed light field and reconstruction time (on CPU/GPU) for 100 test samples. Values in parenthesis are GPU reconstruction times.

tum-Net, an iterative NN based approach was first presented, which incorporates both the physical model and learned regularizer in an unrolled optimization framework. The light field is reconstructed in an iterative way. In each iteration, the reconstruction is refined by a NN refiner and followed by MBIR. This soft-refiner approach increases the reliability of the reconstruction module as the knowledge of the physical model is incorporated through the system matrix in MBIR step. It could potentially have better generalization ability across different light field datasets. Then we presented a sequential-NN based method for light field reconstruction. This approach reconstructs the light field in a non-iterative way. It is a data-driven approach without the need of system matrix at test time and has the benefits of much faster reconstruction speed over the iterative reconstruction methods.

CHAPTER IV

Unsupervised Depth Estimation from Focal Stack

4.1 Introduction

This chapter proposes a learning based methods for depth estimation from focal stack. Since collecting a focal stack dataset with ground truth depth map could be time-consuming and challenging, a method without requiring depth ground truth is of great interest. The method described in section 3.3 is already one method that is able to estimate depth with network trained only using light field as the supervision signal. Here we propose another method to estimate depth, without any supervision: using an input focal stack, an all-in-focus image is estimated using a focus measure and a depth map is estimated using a CNN. Then using a differentiable focal stack synthesis module, a focal stack is reconstructed from the all-in-focus image and the estimated map. The network is trained using the focal stack reconstruction loss without the need of any supervision (self-supervised). The chapter is organized as follows: section 4.2 describes related work on unsupervised depth estimation. Section 4.3 describes our proposed method for unsupervised depth estimation from focal stack. Section 4.4 describes the experimental setup and presents the results of the proposed method.

4.2 Related work

Training a network without supervision is useful when the training labels are hard to obtain. Garg et al. [29] proposed a CNN based unsupervised monocular depth estimation method, which only requires stereo image pairs for training. The network estimates a disparity map for an input left-view image and warps the right-view image using the estimated disparity to reconstruct the left-view image. The network is trained end-to-end by minimizing the photometric reprojection loss. Later work in MonoDepth [31] achieved better performance by exploiting the left-right consistency.

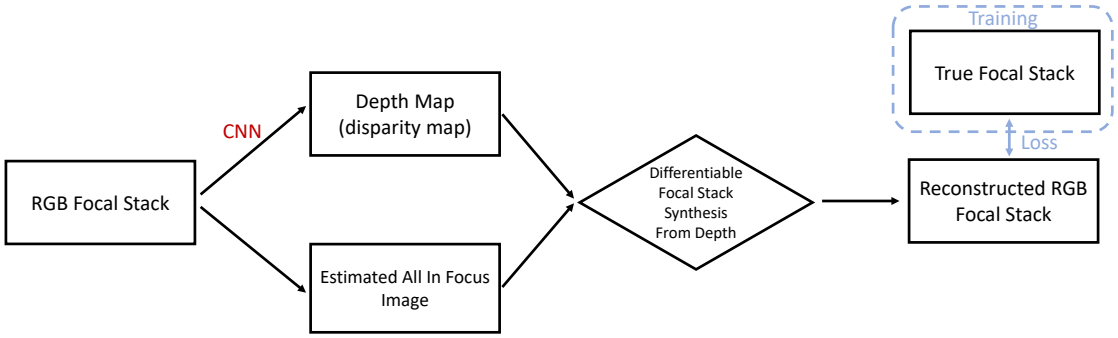

Figure 4.1. Flow chart of the proposed unsupervised depth from focus method.

Srinivasan et al. [96] proposed to use a defocused image as the supervision and trains a depth estimation network end-to-end to predict the scene depths that best explain the defocused image. Similarly, [32] proposed to use a defocused focal stack, instead of single image, as the supervision to train the network. However, both methods accept a single input image as the input and unsupervised depth estimation from focus stack has not been explored, which will be the topic of this chapter.

4.3 Method

Fig. 4.1 illustrates the pipeline of unsupervised depth estimation framework. The input focal stack is passed into a CNN and estimates a depth map. Then a differentiable focal stack rendering module (section 4.3.2) takes in the estimated depth map and an estimated all-in-focus (section 4.3.1) image as inputs and reconstruct the focal stack. A photometric reconstruction error is used as the loss and the gradient can be back-propagated to train the CNN. The details of each step are described in the following subsections.

4.3.1 All-in-focus image estimation

To estimate the all-in-focus image from a focal stack, we first convert all images to gray. Then the images are filtered by 2D gaussian with $\sigma = 1.1$ to reduce the noise. Next, a laplacian operator is applied to each image to measure the focus sharpness. The RGB pixel value of the final estimated all-in-focus image at each pixel location is then taken from the sensor plane with maximum laplacian response. Fig. 4.2 visualizes the result of all-in-focus image estimation.

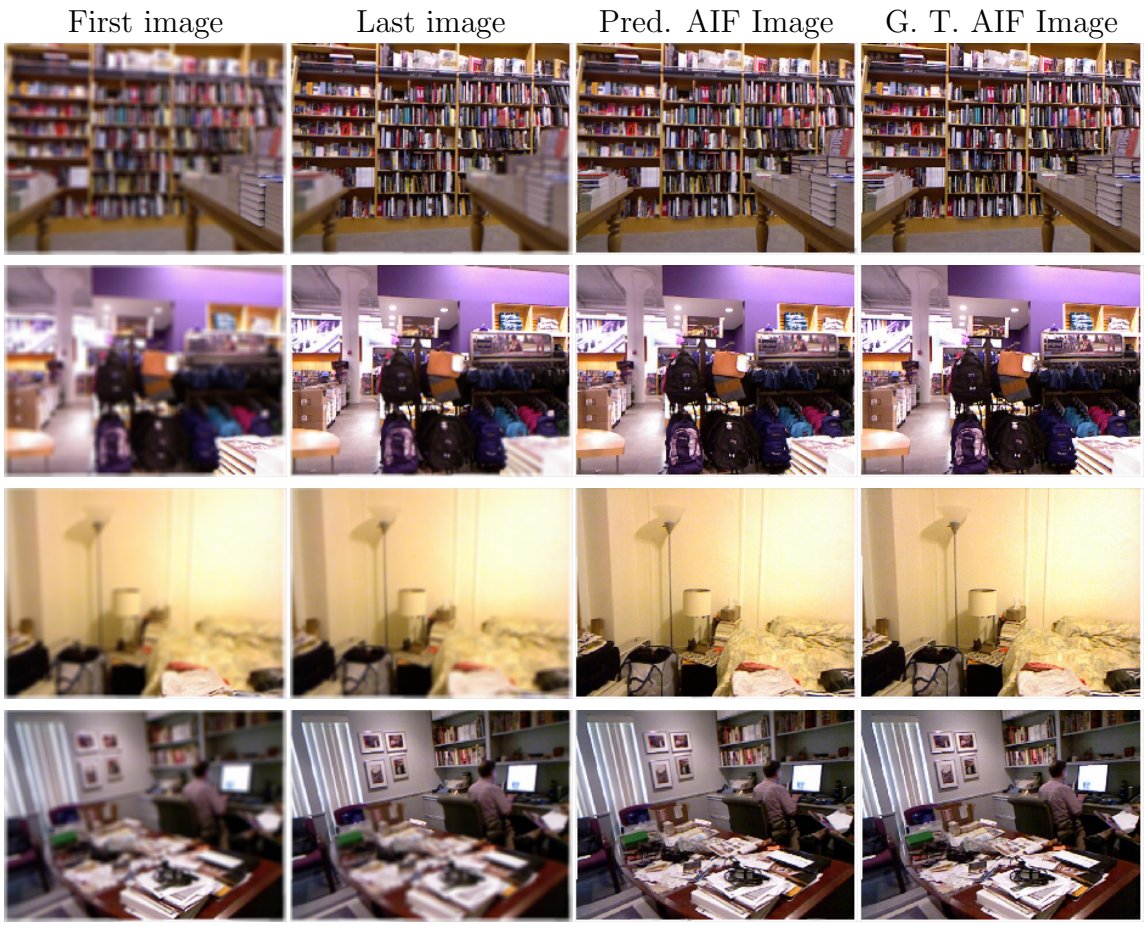

Figure 4.2. Visualization of all-in-focus (AIF) image estimation. First and last image in the focal stack sequence are shown in the first two columns. 3rd column: estimated AIF images. 4th column: ground truth AIF images.

4.3.2 Differentiable focal stack synthesis

Here we will describe the differentiable focal stack synthesis from depth in more detail. According to the thin lens model, the out-of-focus object will be blurred in the captured image I_{blur}. If denoting an all-in-focus image (sharply focused everywhere) as I_{AIF} and the captured defocused image as I_{blur}, the imaging process modeled as follows:

$$I_{\text{blur}} = I_{\text{AIF}} * \text{PSF}(r; R), \tag{4.1}$$

where '$*$' is the convolution operation with spatially varying kernel and $\text{PSF}(r; R)$ is the position-dependent point spread function. We model the $\text{PSF}(r; R)$ as a 2D

Gaussian function with radius R given by:

$$R = A\frac{|O-D|}{O}\frac{f}{D-f}, \tag{4.2}$$

where O is the distance between an object and the lens, D is the focusing distance of the lens, A is the radius of the lens aperture and f is the focal length. With a known camera configuration and an estimated depth map of the scene, we can then calculate a 2D pixel-wise R map using equation 4.2 and generate the defocused image I_{blur} using equation 4.1. Importantly, this process is differentiable with respect to the input depth map, which allows the gradient of reconstruction error in I_{blur} to backprogate to the depth estimation network.

4.3.3 Network training

We used the same depth estimation network as described in section 3.3. The following loss L is used to train the network:

$$L = \lambda_1 L_{\text{rec}} + \lambda_2 L_{\text{smooth}} + \lambda_3 L_{\text{sharp}}, \tag{4.3}$$

$$L_{\text{rec}} = \frac{1}{N}\sum \alpha\frac{1-SSIM(\hat{I}_{\text{blur}}, I_{\text{blur}})}{2} + (1-\alpha)\|\hat{I}_{\text{blur}} - I_{\text{blur}}\|_1, \tag{4.4}$$

$$L_{\text{smooth}} = \frac{1}{N}\sum |\partial_x \hat{O}|\exp^{-|\partial_x I_{\text{AIF}}|} + |\partial_y \hat{O}|\exp^{-|\partial_y I_{\text{AIF}}|} \tag{4.5}$$

$$L_{\text{sharp}} = \|S(\hat{I}_{\text{blur}}) - S(I_{\text{blur}})\|_1, \tag{4.6}$$

where $S(I)$ is a sharpness measure described in [59], and ' $\hat{}$ ' indicates the quantity is from estimation.

4.4 Experimental setup and results

We trained and evaluated the proposed method using NYU-v2 dataset [69]. It contains 120k RGB images of indoor scene in depth range [0.7 m, 10 m], with corresponding depth maps captured using Microsoft Kinect. Following previous works [32, 105], we used 654 images from a subset of 1449 aligned RGB-depth pairs for testing. Since the NYU-v2 dataset only contains sharp in-focus images, we generated a focal stack dataset with $n_{\text{F}} = 6$ from sharp in-focus images using equation 4.1. Following the method in [120], we set their focus distances to be [0.8 m, 1 m, 1.2 m, 1.6 m, 2.4 m, 5 m]. This particular focus distance setting ensures the depth of field of neighboring imaging sensors are contacting with each other, but with no overlap. We trained the

network for 170k iterations using a batch size of 2, a learning rate of 2×10^{-5} using Adam optimizer [48].

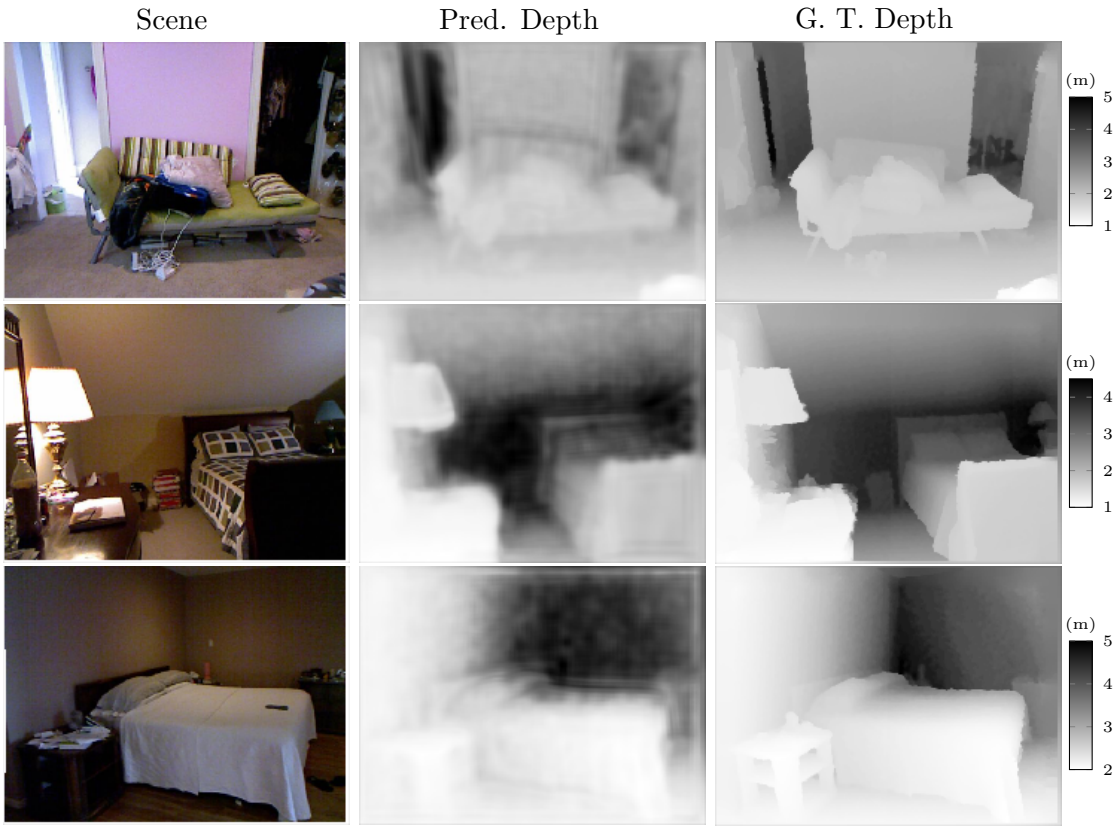

Figure 4.3. Visualization of the depth estimation result.

Fig. 4.3 visualizes depth estimation results on the test samples. It shows the proposed method can estimate the depth of the scene with good quality using the focal stack, without depth supervision during training. Table 4.1 compares the depth estimation result of several methods. Comparing with prior work of single image based unsupervised depth estimation in [32] (2nd row), our proposed method (3rd row) achieves much lower RMSE and better δ accuracy, demonstrating the advantage of focal stack for 3D sensing purpose. We also compared the depth estimation accuracy using the proposed method, with either all-in-focus image estimated from focal stack (3rd row) or with ground truth all-in-focus image (4th row). It indicates that with a better all-in-focus image estimation can further improve the depth estimation accuracy. This could be achieved using a deep-learning based all-in-focus image estimation and is the direction of our future work.

	Supervision	RMSE (m)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Depth from single image [32]	FS	0.546	0.797	0.951	0.987
Focal stack with syn. I_{AIF}	FS	0.310	0.959	0.990	0.997
Focal stack with g.t. I_{AIF}	FS	0.244	0.955	0.985	0.997
Supervised depth from focus	Depth	0.174	0.983	0.997	0.999

Table 4.1. Result of unsupervised depth from focus.

4.5 Summary

This chapter presents an unsupervised depth from focal stack method. The proposed method estimates a depth map from input focal stack. By using a differentiable focal stack synthesis module and a focal stack reconstruction loss, the network can be trained end to end without depth supervision. Numerical experiments shows that the proposed method achieves good depth estimation accuracy and outperforms existing single-image based unsupervised depth estimation method.

CHAPTER V

Focal Stack Based 3D Tracking

5.1 Introduction

Chapter III have described 4D light field imaging and reconstruction from a focal stack. However, some optical applications, e.g., ranging and tracking, do not require computationally expensive 4D light field reconstruction. The question naturally arises as to whether the focal stack geometry will allow optical sensor data to provide the necessary information for a given application, without reconstructing a 4D light field or estimating a 3D scene structure via depth map.

To this end, this chapter demonstrates how combinations of focal stacks obtained by transparent sensor arrays and machine learning algorithms enable 3D object tracking, without the need for light-field reconstruction. Experimental results illustrate that the implemented neural networks using focal stack data can achieve accurate 3D object tracking efficiently (millisecond inference time using a conventional GPU computing power). This work demonstrates a transparent focal stack imaging system that is capable of tracking single and multiple point objects in 3D space. The proof-of-concept experiment is demonstrated with a vertical stack of two 4×4 (16-pixel) graphene sensors and feedforward neural networks that have the form of a multilayer perceptron (MLP). The imaging schematic is illustrated in Fig. 1.10. We also acquired focal stack data sets using a conventional CMOS camera with separate exposures for each focal plane. The simulations demonstrate the capability of future higher-resolution sensor arrays for tracking extended objects. Our experimental results show that the graphene-based transparent photodetector array is a scalable solution for 3D information acquisition, and that a combination of transparent photodetector arrays and machine learning algorithms can lead to a compact camera design capable of capturing real-time 3D information with high resolution. This type of optical system is potentially useful for emerging technologies such as

face recognition, autonomous vehicles and unmanned aero vehicle navigation, and biological video-rate 3D microscopy, without the need for an integrated illumination source. Graphene-based transparent photodetectors can detect light with a broad bandwidth from visible to mid-infrared. This enables 3D infrared imaging for even more applications. This work has been published in *Nature Communications* [114].

5.2 Focal stack imaging with transparent sensors

The concept of focal stack imaging was demonstrated using two vertically stacked transparent graphene arrays. As shown in Fig. 5.1(a), two 4×4 sensor arrays were mounted vertically along the optical axis, separated at a controlled distance, to form a stack of imaging planes. This double-focal-plane stack essentially serves as the camera of the imaging system. A convex lens focuses a 532 nm laser beam, with the beam focus serving as a point object. The focusing lens was mounted on a 3D-motorized stage to vary the position of the point object in 3D. The AC photocurrent is recorded for individual pixels on both front and back detector arrays while the point object is moving along the optical axis. Fig. 5.1(b) shows a representative set of images captured experimentally by the two detector arrays when a point object is scanned at different positions along the optical axis (12 mm, 18 mm, 22 mm) respectively, corresponding to focus shifting from the back plane toward the front plane (Fig. 2(c)). The grayscale images show the normalized photoresponse, with white (black) color representing high (low) intensity. As the focus point shifts from the back plane toward the front plane, the image captured by the front plane shrinks and sharpens, while the image captured by the back plane expands and blurs. Even though the low pixel density limits the image resolution, these results nevertheless verify the validity of simultaneously capturing images at multiple focal planes.

5.3 3D tracking of point objects

While a single image measures the lateral position of objects as in conventional cameras, differences between images captured in different sensor planes contain the depth information of the point object. Hence focal stack data can be used to reconstruct the 3D position of the point object. Here we consider three different types of point objects: a single-point object, a three-point object, and a two-point object that is rotated and translated in three dimensions.

First, we consider single-point tracking. In this experiment, we scanned the point

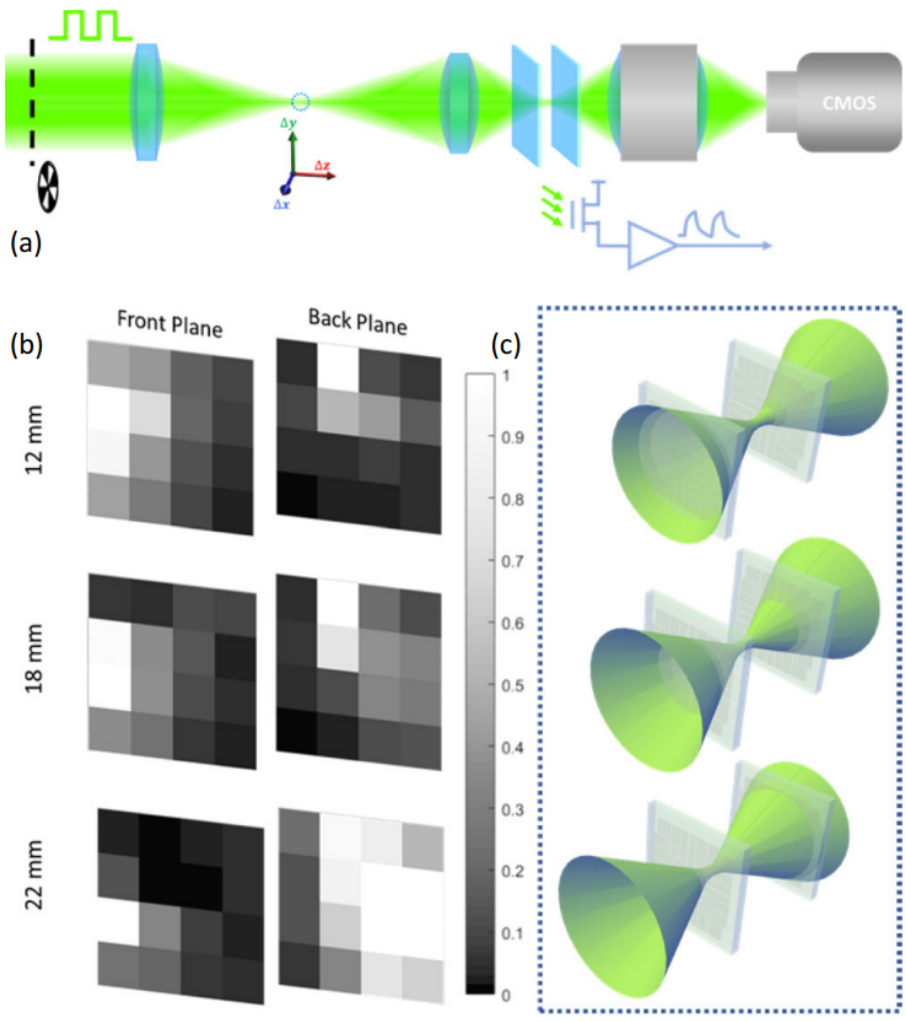

Figure 5.1. Experimental demonstration of focal stack imaging using double stacks of graphene detector arrays. (a) A schematic of measurement setup. A point object (dotted circle) is generated by focusing a green laser beam (532 nm) with the lens. Its position is controlled by a 3D motorized stage. Two detector arrays (blue sheets) are placed behind the lens. An objective and CCD camera are placed behind the detector array for sample alignment. A chopper modulates the light at 500 Hz and a lock-in amplifier records the AC current at the chopper frequency. (b) Images captured by the front and back photodetector planes with objects at three different positions along the optical axis (12 mm, 18 mm, 22 mm respectively). The grayscale images are generated using responsivities for individual pixels within the array, normalized by the maximum value for better contrast. The point source is slightly off-axis in the image presented, leading to the shift of spot center. (c) The illustrations of the beam profiles corresponding to the imaging planes in (b). The focus is shifting from the back plane (top panel) toward the front plane (bottom panel).

source (dotted circle in Fig. 5.1(a)) in a 3D spatial grid of size 0.6 mm $\times$ 0.6 mm

(x, y axes) × 20 mm (z axis, i.e., the longitudinal direction). The grid spacing was 0.06 mm along the x, y axes, and 2 mm along the z axis, leading to 1,331 grid points in total. For each measurement, two images were recorded from the graphene sensor planes. We randomly split the data into two subsets, training data with 1131 samples (85% of total samples) and testing data with 200 samples (15% of total samples); all experiments used this data splitting procedure. To estimate three spatial coordinates of the point object from the focal stack data, we trained three separate MLP neural networks (one for each spatial dimension) with mean-square error (MSE) loss. The results (Fig. 5.2(a)(b)) show that even with the limited resolution provided by 4×4 arrays, and only two sensor planes, the point object positions can be determined very accurately. We used RMSE to quantify the estimation accuracy on the testing dataset; we obtained RMSE values of 0.012 mm, 0.014 mm, and 1.196 mm along the x, y, and z directions, respectively.

Given the good tracking performance with the small-scale (i.e., 4×4 arrays) graphene transistor focal stack, we studied how the tracking performance scales with array size. We determined the performance advantages of larger arrays by using conventional CMOS sensors to acquire the focal stack data. For each point source position, we obtained multi-focal plane image stacks by multiple exposures with varying CMOS sensor depth (note that focal stack data collected by CMOS sensors with multiple exposures would be comparable to that obtained by the proposed transparent array with a single exposure, as long as the scene being imaged is static), and down-sampled the resolution of high resolution (1280×1024) images captured by CMOS sensor to 4×4, 9×9, and 32×32. We observed that tracking performance improves as the array size increases; results are presented in appendix B.5.

We next considered the possibility of tracking multi-point objects. Here, the object consisted of three point objects, and these three points can have three possible relative positions to each other. We synthesized 1,880 3-point objects images as the sum of single-point objects images from either the graphene detectors or the CMOS detectors (see details of focal stack synthesis in appendix B.2). This synthesis approach is reasonable given that the detector response is sufficiently linear and it avoids the complexity of precisely positioning multiple point objects in the optical setup. To estimate the spatial coordinates of the 3-point synthetic objects, we trained a MLP neural network with MSE loss that considers the ordering ambiguity of the network outputs (see appendix B.4). We used 3-point object's data synthesized from the CMOS-sensor readout in the single-point tracking experiment (with each CMOS image smoothed by spatial averaging and then down-sampled to 9×9). We found

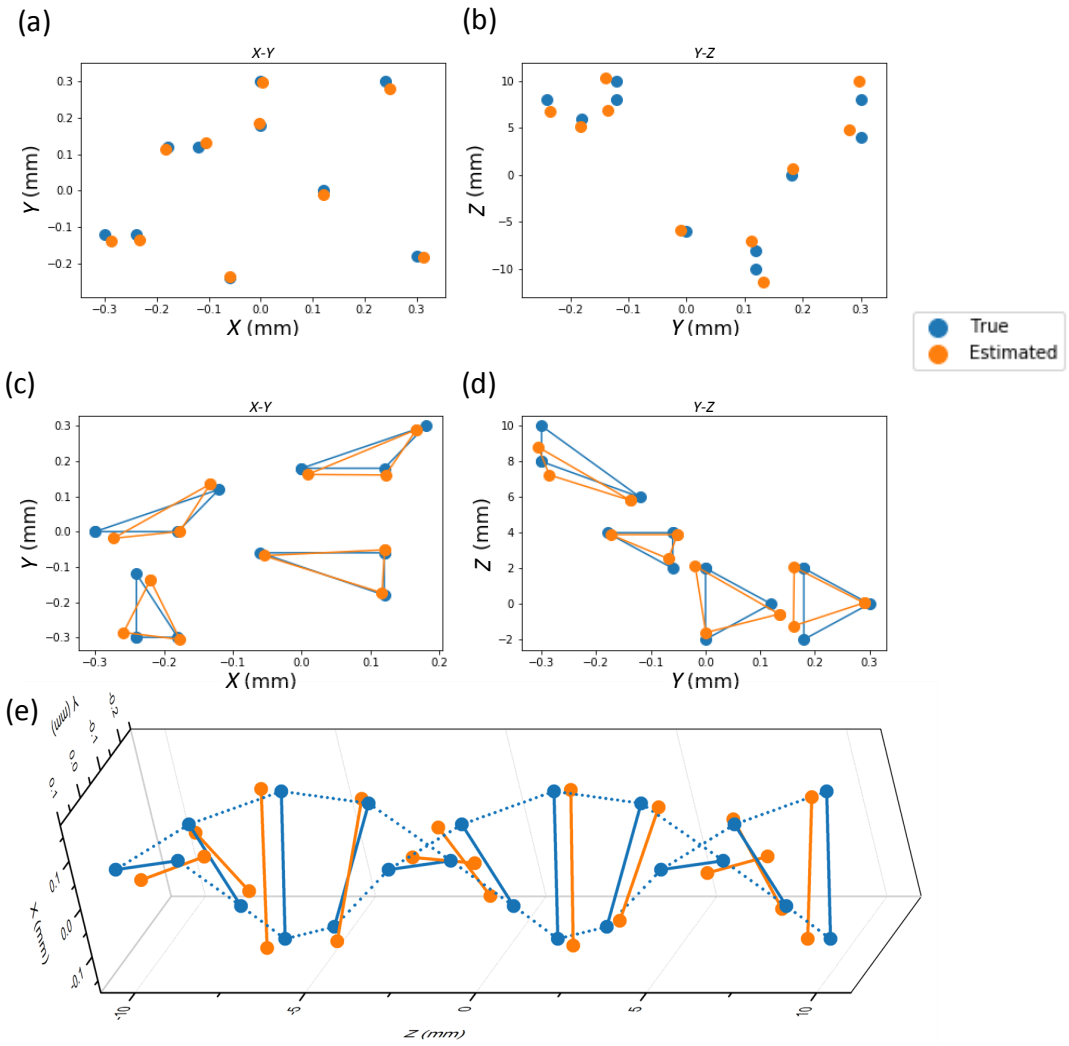

Figure 5.2. (a-b) Tracking results for single point object. Results are based on images captured with the graphene photodetector arrays. (c-d) Tracking results for three points objects. Results are based on data synthesized from multi focal-plane CMOS images (downsampled to 9×9) of single point source. (e): Tracking results for rotating two-point objects on one testing trajectory. Results are based on data synthesized from single point source images captured with graphene photodetector arrays.

that the trained MLP neural network can estimate a multi-point object's position with remarkable accuracy; see Fig. 5.2(c-d). The RMSE values calculated from the entire test set are 0.017 mm, 0.016 mm, 0.59 mm, along x-, y-, z-directions, respectively. Similar to the single-point object tracking experiment, the multi-point object tracking performance improves with increasing sensor resolution (see appendix B.5).

Finally, we considered tracking of a two-point object that is rotated and translated in three dimensions. This task aims to demonstrate 3D tracking of a continuously

moving object, such as a rotating solid rod. Similar to the 3-point object tracking experiment, we synthesized a 2-point object focal stack from single-point object focal stacks captured using the graphene transparent transistor array. The two points are located at the same x-y plane and are separated by a fixed distance, as if tied by a solid rod. The rod is allowed to rotate in the x-y plane and translate along the z-axis, forming helical trajectories, as shown in Fig. 5.3(e). We trained a MLP neural network with 242 training trajectories using MSE loss to estimate the object's spatial coordinates and tested its performance on 38 test rotating trajectories. Fig. 5.2(e) shows the results of one test trajectory. The neural network estimated the orientation (x- and y-coordinates) and depth (z-coordinate) of test objects with good accuracy: RMSE along x-, y-, and z-directions for the entire test set are 0.016 mm, 0.024 mm, 0.65 mm, respectively. Appendix B.4 gives further details on the MLP neural network architectures and training.

5.4 3D extended object tracking

The aforementioned objects consisted of a few point sources. For non-point-like (extended) objects, the graphene 4×4 pixel array fails to accurately estimate the configuration, given the limited information available from such a small array. To illustrate the possibilities of 3D tracking of a complex object and estimating its orientation, we used a ladybug as an extended object and moved it in a 3D spatial grid of size 8.5 mm $\times$ 8.5 mm $\times$ 45 mm. The grid spacing was 0.85 mm along both x- and y-directions, and 3 mm along z-direction. At each grid point, the object took 8 possible orientations in the x-z plane, with 45° angular separation between neighboring orientations (see experiment details in appendix B.3). We acquired 15,488 high-resolution focal stack images using the CMOS sensor (at two different planes) and trained two convolutional neural networks (CNNs), one to estimate the ladybug's position and the other for estimating its orientation, with MSE loss and the cross-entropy loss, respectively. Fig. 5.3 shows the results for five test samples. The CNNs correctly classified the orientation of all five samples and estimated their 3D position accurately. For the entire test set, the RMSE along x-, y-, and z-directions is 0.11 mm, 0.13 mm, and 0.65 mm, respectively, and the orientation is classified with 99.35% accuracy. We note that at least two imaging planes are needed to achieve good estimation accuracy along depth (z)-direction: when the sensor at the front position is solely used, the RMSE value along z-direction is 2.14 mm, and when the sensor at the back position is solely used, the RMSE value along z-direction is 1.60

mm.

Appendix B.4 describes the CNN architectures and training details.

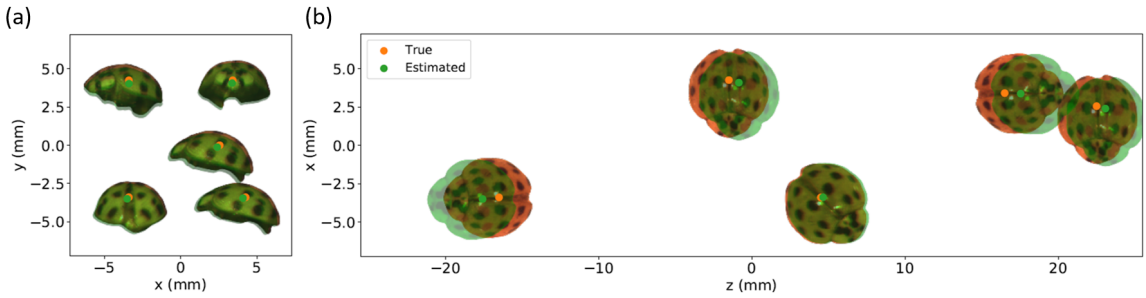

Figure 5.3. 3D extended-object tracking and its orientation estimation using focal stack data collected by a CMOS camera, in (a) the x-y-plane perspective and (b) in the x-z-plane perspective. The estimated (true) ladybug's position and orientation are indicated by green (orange) dots and green (orange) overlaid ladybug images.

5.5 Summary

In conclusion, we designed and demonstrated a focal stack imaging system enabled by graphene transparent photodetector arrays and the use of feedforward neural networks. Even with limited pixel density, we successfully demonstrated simultaneous imaging at multiple focal planes, which can be used for 3D tracking of point objects with high speed and high accuracy. Our computer model further proves that such an imaging system has the potential to track an extended object and estimate its orientation at the same time. Future advancements in graphene detector technology, such as higher density arrays and smaller hysteresis enabled by higher quality tunnel barriers, will be necessary to move beyond the current proof-of-concept demonstration. We also want to emphasize that the proposed focal stacking imaging concept is not limited to graphene detectors alone. Transparent (or semi-transparent) detectors made from other 2D semiconductors and ultra-thin semiconductor films can also be implemented as the transparent sensor planes within the focal stacks. The resulting ultra-compact, high-resolution, and fast 3D object detection technology can be advantageous over existing technologies such as LiDAR and light-field cameras. Our work also showcases that the combination of nanophotonic devices, which is intrinsically high-performance but nondeterministic, with machine learning algorithms can complement and open new frontiers in computational imaging.

CHAPTER VI

Focal Stack Camera Depth Estimation Performance Comparison and Design Exploration

6.1 Introduction

Previous chapters have presented 3D sensing applications of the focal stack camera, including light field reconstruction (chapter III), depth estimation (chapter III, IV), and 3D tracking (chapter V). Despite these successful demonstrations of focal stack camera applications, the dependence of the focal stack camera design on its 3D sensing performance has not yet been explored. It is also unknown what the performance trade-offs might be when comparing the focal stack and light field camera approaches.

This chapter addresses these questions via a set of numerical experiments. Specifically, we focus on depth estimation performance evaluation, using deep learning based methods. Using focal stacks that are either computed from publicly available light field datasets [38, 33, 81] or captured experimentally, we train convolutional neural network (CNN) models to estimate depth maps from the input focal stack and study the dependence of the camera parameters, including number of sensor planes, aperture size and sensor resolution, on the depth estimation accuracy. We further compare the system performance with the light field camera and show that focal stacks achieve comparable performance.

This chapter is organized as follows: Section 6.2.1 and 6.2.2 describe the background and methods for focal stack and light field depth imaging. Section 6.2.3 describes the network structure we used for estimating the depth from the focal stack and from the light field. Section 6.2.4 describes the datasets we used for performance evaluation. Section 6.3 contains the experiment results and analysis. This work has been submitted to *OSA Optics Express* for peer review.

6.2 Methods

6.2.1 Focal stack depth imaging

Several approaches have been developed to estimate depth maps from a focal stack. Nayar et al. [70] used a sum-modified-Laplacian to measure the focus sharpness and fit the focus sharpness by a gaussian distribution to obtain accurate depth. Moeller et al. [62] cast the depth estimation as a nonconvex optimization problem that includes a data fidelity term and a regularization term, which is solved by linearized alternating directions method of multipliers (ADMM) [12]. Sakurikar et al. [91] used a composite focus measure that is a weighted combination of standard focus measures to measure the focus sharpness and showed that it achieves better performance than those using a single individual focus measure. Hazirbas et al. [33] trained a deep neural network for depth estimation from focal stack.

6.2.2 Light field depth imaging

As one of our goals in this paper is to compare the depth estimation performance of the focal stack camera with the light field camera, this section describes some related works on the light field based depth sensing.

Since light field is essentially a multi-view image set, identifying the pixel correspondence between different views in the light field allows one to estimate the depth. Chen et al. [14] proposed a bilateral consistency metric to evaluate the surface camera light field and then apply a stereo matching algorithm to estimate the depth. Shin et al. [93] trained a neural network, EPINet, to process sub-aperture views along horizontal, vertical, and diagonal directions to regress a depth map. Tsai et al. [100] computed a 4D disparity cost volume and employed an attention mechanism to scale a feature map from each sub-aperture view by its importance and then estimate the depth. As illustrated in chapter I, since EPIs of a light field contains stripe-like structure with slope indicating the depth, Zhang et al. [117] designed a spinning parallelogram to estimate the slope of lines in the EPIs of the light field.

6.2.3 Network structure

This section describes the neural networks used for estimating the depth from a focal stack and from light field images. Fig. 6.1 shows the neural network structure we used for estimating depth from a focal stack. The input RGB focal stack contains n_{F} images that are concatenated along the color dimension for a total of $3n_{\mathrm{F}}$ input

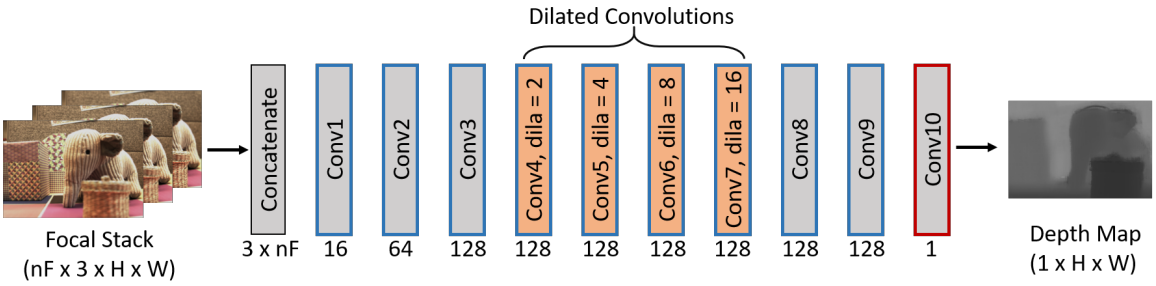

Figure 6.1. Network structure for depth estimation from focal stack. All convolutions have filter size of 3×3, stride 1, and the output channel number for each layer is indicated beneath. Blue border around a layer indicates that Batch Normalization and leaky ReLU are applied to the output. Red border indicates tanh non-linearity is applied to the output. n_{F} is the number of images in the focal stack.

channels. The network consists of 10 convolution layers with no spatial pooling or up-sampling operations, to preserve fine-details in the final output. Dilated convolutions [110] are used to ensure a large receptive field without significant computation cost. The output from the last convolution layer (after tanh nonlinearity) is further scaled and offset by dataset-dependent constant α and β, respectively to constrain the output to a plausible range.

We use the EPI-Net [93] for estimating the depth from the light field image. The network has a four-branch structure, where each branch takes in sub-aperture images of the light field along a particular direction (horizontal, vertical, left-diagonal or right-diagonal). Features are extracted from each branch independently using 2D convolutions and then concatenated along the color dimension. Then additional convolutions are used to process the concatenated feature map to predict the final depth map. More details about the network structure can be found in [93].

6.2.4 Focal stack dataset

We generated focal stack data from three publicly available light field datasets: the HCI light field dataset [38], the DDFF dataset [34] and the CVIA dataset [81]. The HCI light field dataset contains 28 synthetic light fields of resolution $9 \times 9 \times 512 \times 512$, of which 16 light fields in the category 'additional' are used as the training data and the remaining 8 light fields are used as the testing data. We synthesized focal stacks using the add-shift algorithm [73], with images focusing at disparity planes evenly distributed in [-3,3]. The DDFF dataset contains 600 training and 120 testing realistic light fields of size $9 \times 9 \times 383 \times 552$ captured by a Lytro light field camera. 480 light fields from the original training data are used in our experiments for training,

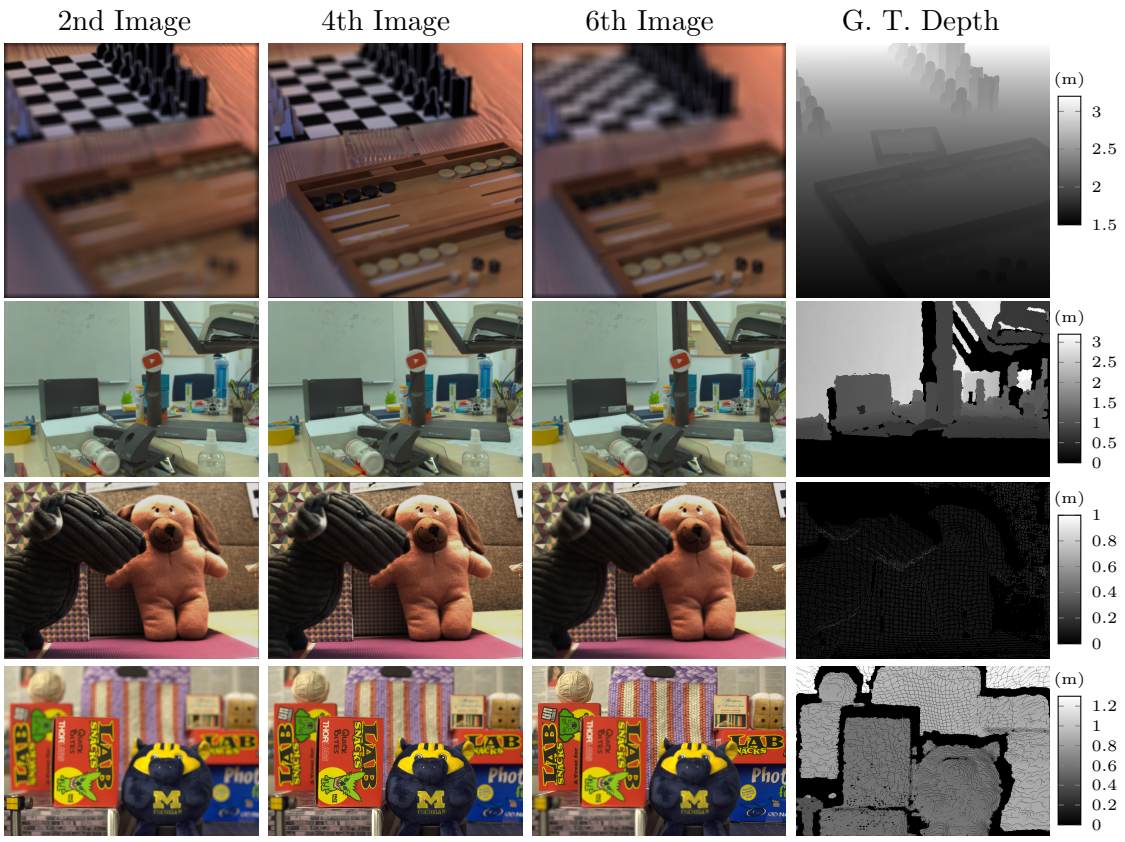

Figure 6.2. Example focal stacks showing the 2nd, 4th and 6th images in the stack sequence. Last column shows the ground truth depth maps. Rows correspond to HCI dataset, DDFF dataset, CVIA dataset and Nikon dataset, respectively.

with the remaining 120 light fields in the original training data used for testing. We synthesized focal stacks, each containing n_{F} images focusing at disparity planes evenly distributed in [0.020, 0.282]. The CVIA dataset contains 40 light fields of resolution $15 \times 15 \times 434 \times 625$ in a distance range of 0.2 to 1.6m using a Lytro camera, of which 32 are used for training, and 8 for testing. We synthesized focal stack with images focusing at disparity planes evenly distributed in [-0.44, 0.17]. All datasets above contain ground truth depth maps for evaluation, either from its synthetic 3D models (HCI synthetic light field), or from depth sensors (DDFF dataset and CVIA dataset).

In addition to using focal stacks generated from the existing light field datasets, we also collected an additional focal stack dataset, which we named as Nikon dataset. Unlike all the above datasets, where the focal stacks are synthesized from the light fields, it consists of focal stacks captured directly using a DLSR camera (Nikon D7200, 35 mm lens) with focal stacking function provided in camera control software 'controlMyNikon'. As such, the focal stacks in the Nikon dataset resemble most closely

the focal stack one would capture using the focal stack imaging system shown in Fig. 1.10. We set the step size of the focal stacking in the software to 200, and size of the focal stack n_{F} to 7, which covers a depth range of approximately 0.4 m to 1.3 m. We repeat the focal stack collection process for 4 aperture size settings (f/3.2, f/5, f/10, f/22). Each setting contains 40 focal stacks of resolution 854×1280 after resizing, of which 32 are used as the training data and the remaining 8 are used as the test data. We additionally form Nikon datasets with $n_{\mathrm{F}} = 2$, from the $n_{\mathrm{F}} = 7$ datasets, by using only the 2nd and 6th focus position images, which are used to study the dependence on number of sensor planes. Since the raw captured focal stack exhibits a focus breathing effect due to the change of magnification when the 35-mm lens focus is changed, we additionally perform a focal stack alignment process to compensate the magnification change and align the images in the focal stack. We also capture ground truth depth maps for each focal stack, using an Intel RealSense D415 Depth Camera, and register the depth onto the RGB images. More details on the focal stack collection, focal stack alignment and depth registration can be found in the appendix D. Fig. 6.2 shows example focal stack images of the datasets we use.

6.3 Experiments and results

We trained separate networks (Fig. 6.1) to estimate depth, using focal stack datasets with varying camera parameters (number of sensor planes in focal stack, sensor resolution, aperture size), to study their dependence on the depth estimation accuracy. Finally, we compared the depth performance from the focal stack and the light field. The details of the training setup and experiments are described next.

6.3.1 Training setup

All networks were trained in Pytorch with L_1 loss using Adam optimizer [48] with learning rate 10^{-4}, batch size 4. The input focal stacks/light fields were randomly cropped in the spatial dimension to 125×125. Models were trained till convergence (80k epochs for HCI dataset, 5k epochs for DDFF dataset, 15k epochs for CVIA and Nikon datasets).

6.3.2 Sensor resolution dependence

Here we study how the sensor pixel resolution affects the depth estimation performance. Specifically, a down-sample rate of N means reducing the effective resolution

of the images in the focal stack by setting the pixel values in every $N \times N$ block to the value of the top left pixel. Fig. 6.3a illustrates the downsampling process that mimics the fact that the active sensing areas (individual pixels) of a low resolution sensor are not densely packed in the 2D plane. The first 3 rows of Fig. 6.3b show example focal stacks ($n_{\mathrm{F}} = 2$) with varying down-sample rates collected with f/3.2 aperture setting, along with the estimated depth maps, which indicate that higher resolution sensors lead to higher quality depth maps as one may intuitively expect. The left column of Fig. 6.4 shows how the RMSE of the depth estimates depend on the focal stack image resolution. Better resolution images lead to better performance on DDFF, CVIA and Nikon datasets. This trend can be understood as follows: degrading the resolution causes some defocus blur information to be lost (at the extreme of very low resolution, objects at all depths will be equally blurred). In addition, the $n_{\mathrm{F}} = 7$ result has lower RMSE than that from $n_{\mathrm{F}} = 2$, especially for a large down-sample rate, indicating that having more focal planes in the focal stack camera is helpful, as expected.

6.3.3 Aperture size dependence

We next study how the aperture size affects the depth estimation performance. According to Eq. 1.3, a larger aperture leads to a larger defocus blur, which could potentially affect the depth estimation performance. For focal stacks that are synthesized from a light field (HCI dataset, DDFF dataset, CVIA dataset), changing the aperture size can be realized by refocusing using only the sub-aperture images that are within the desired aperture window from the light field. For our Nikon datasets, we acquired separate focal stacks with different aperture sizes for each scene. Comparing the 1st and 4th row of Fig. 6.3b shows the effect of reducing the aperture size. The images in the focal stack become sharper as the aperture is reduced and the estimated depth becomes noisier. The right column of Fig. 6.4 shows quantitatively that decreasing the aperture size increases the RMSE error. This trend can be understood because in the limit of very small aperture size, all images in the focal stack would be the same image with every depth in focus. Comparing the results of $n_{\mathrm{F}} = 2$ and $n_{\mathrm{F}} = 7$ with changing aperture size, having more focal planes slightly improves the accuracy in this case.

6.3.4 Focal stack and light field camera comparison

Here we compare the performance between depth from light field and depth from focal stack on the HCI, DDFF and CVIA datasets. EPINet [93] is used to estimate

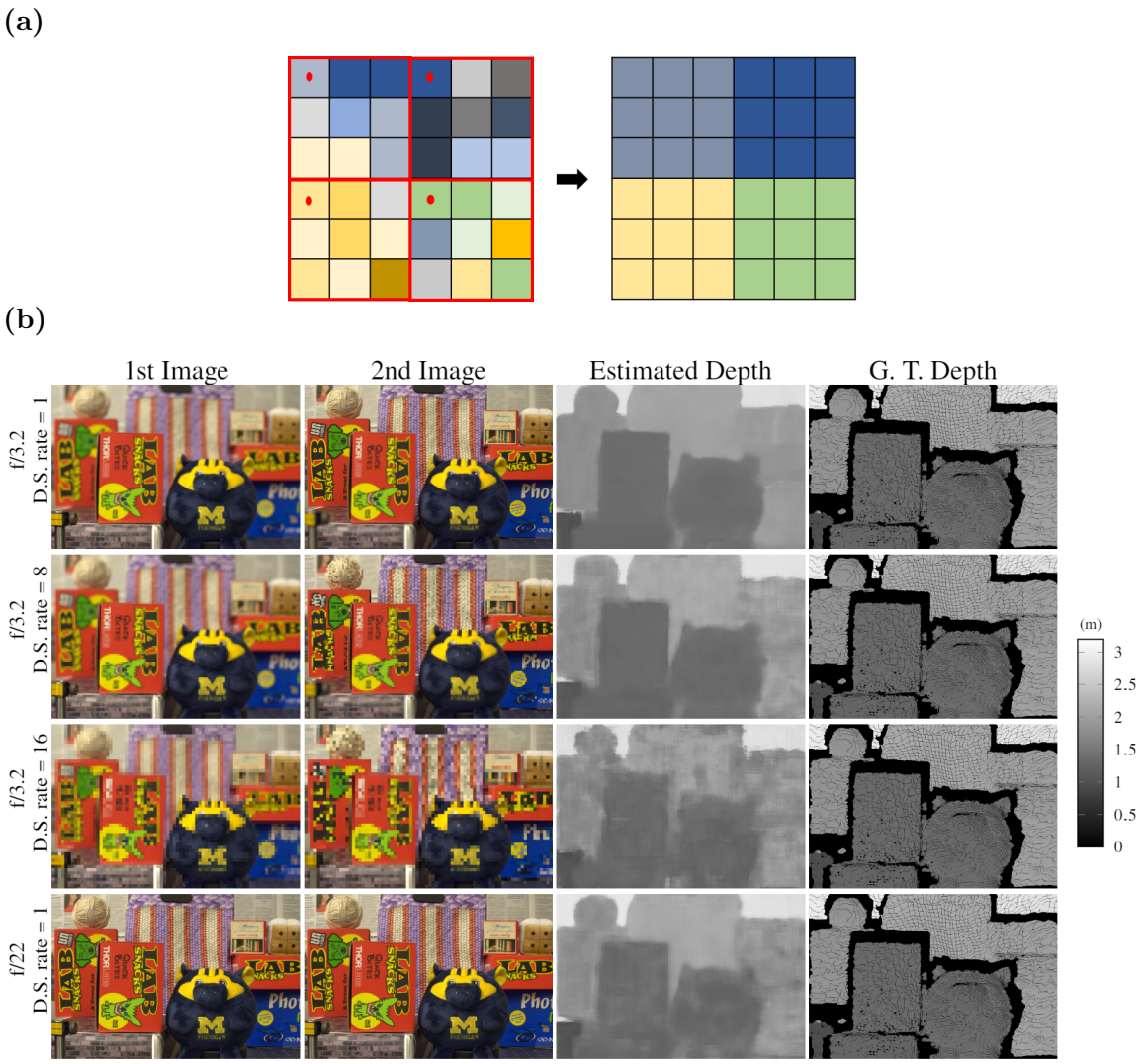

Figure 6.3. Example focal stacks with different camera parameters in Nikon dataset. (a) Schematic illustrating focal stack generation with down-sample rate = 3. (b) Focal stack examples ($n_F = 2$) captured with different down-sample rate and aperture setting. The depth estimated from the focal stack and the ground truth depth are also shown.

	DDFF	**CVIA**	**HCI**
Focal Stack	0.018	0.035	0.36
Light Field	0.027	0.042	0.17

Table 6.1. RMSE of depth map estimated from focal stack and light field. Focal stack of $n_F = 7$ is used. For DDFF and HCI, the RMSE is calculated on the disparity map with unit of pixel. For CVIA, the RMSE is calculated on the depth map with unit of meter. Largest possible aperture is used in all experiments.

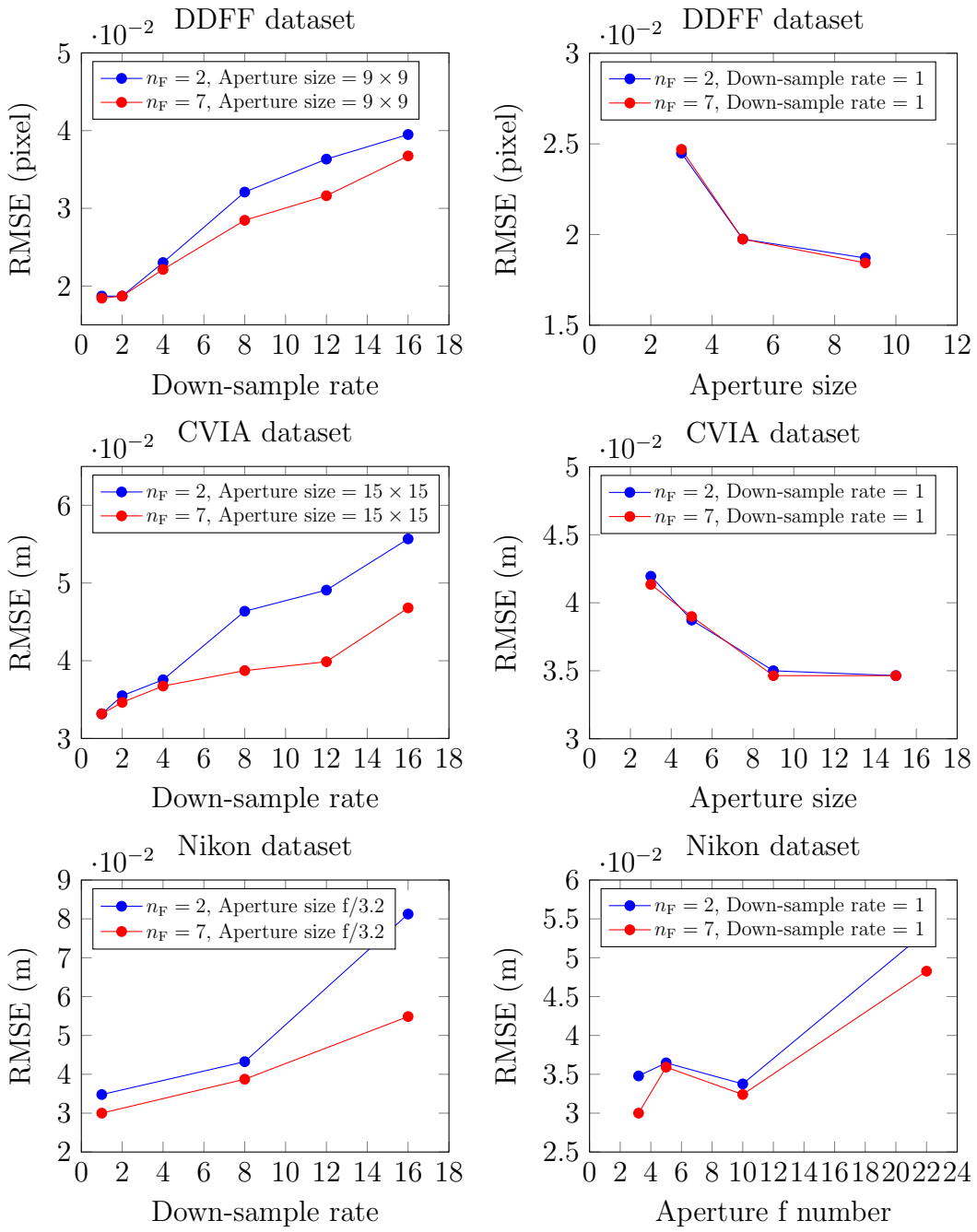

Figure 6.4. RMSE of the depth estimated from focal stack images on DDFF dataset, CVIA dataset and Nikon dataset as a function of resolution down-sample rate (left column), aperture size (right column) and number of sensor planes n_F.

the depth from light fields. Light fields and focal stacks with the largest possible aperture sizs are used for each dataset. We used $n_F = 7$ for the focal stack data and used no down-sampling of the focal stack/light field images. Table 6.1 shows that the depth estimation from focal stack has a disparity RMSE error of 0.018 pixel on

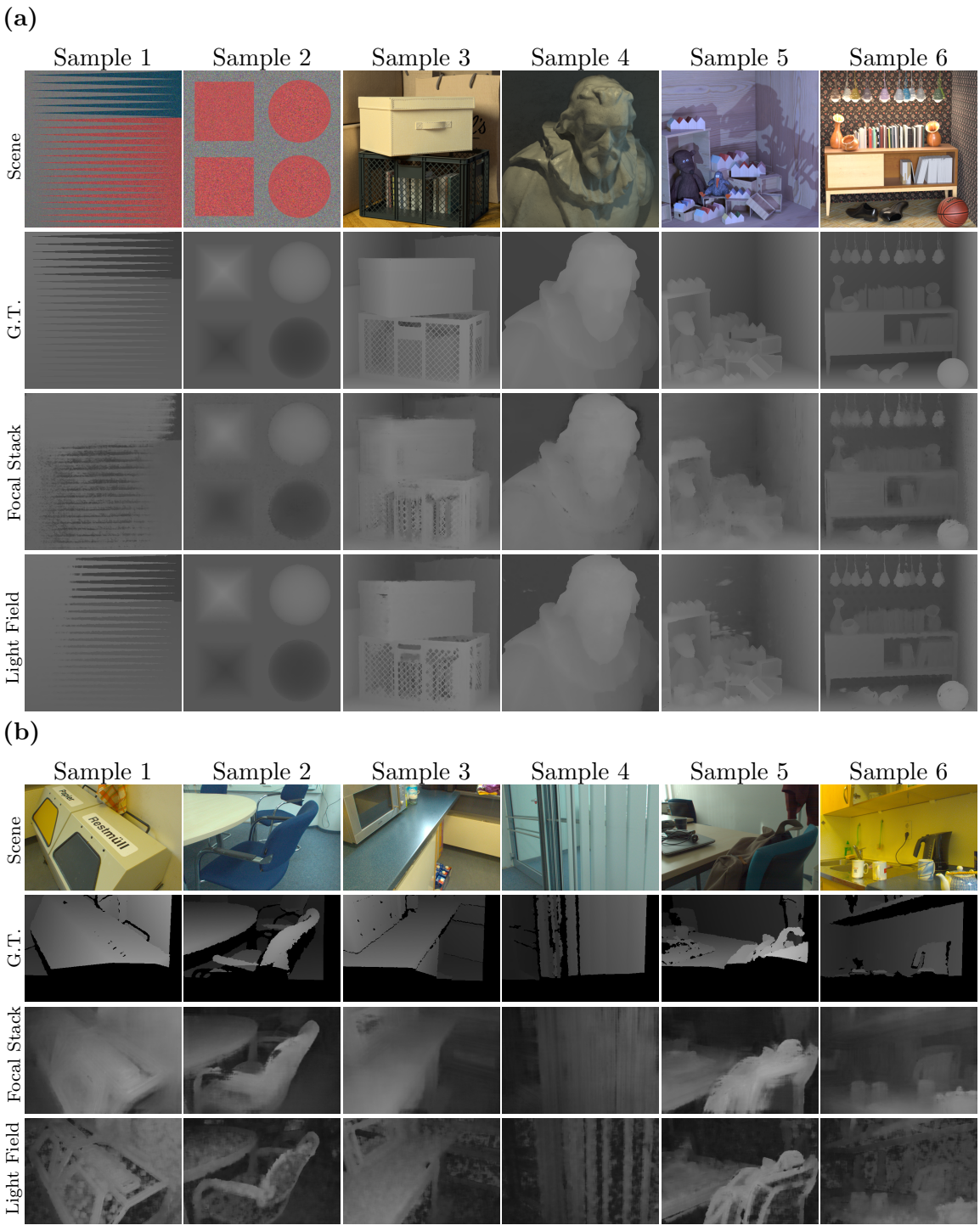

Figure 6.5. Qualitative disparity estimation results from light field data and focal stack data. (a) Results on HCI dataset. (b) Results on DDFF dataset.

the DDFF dataset, which is 33% lower compared to that from the light field. On the CVIA dataset, the focal stack based method also performs better than the light field based method, with 17% lower RMSE. However, the light field based depth estimation

performs better on the HCI dataset, with a disparity RMSE of 0.17 pixel, as opposed to 0.36 pixel for focal stack based method.

To better understand when the focal stack would perform better than a light field camera for depth estimation, Fig. 6.5 shows qualitative depth estimation results on the HCI dataset and DDFF datasets. On the HCI dataset (Fig. 6.5a), depth from light field can better resolve the fine structures compared to focal stack method, as can be seen, for example, by comparing the estimated depth maps of sample 1. This is likely because HCI dataset has a large disparity and hence the amount of defocus blur on the out-of-focus object is significant. Unless the object happens to be in focus on one of the image plane, it would be hard to precisely localize the object boundary using the focal stack. On the DDFF dataset, light field based method performs poorly and shows poor estimates on texture-less regions. This is because the maximum disparity in the DDFF dataset is small, and as a result the sub-aperture images in the light field become very similar. This makes it hard to estimate the depth from the light field. On the other hand, the focal stack based method is still able to produce smooth and good depth estimates in this case, by analyzing the small change in the focus sharpness, which is what a CNN excels at. This also suggests that more information is not always better, and the way the information is presented is also important: the light field, which has a larger data size and more information, may not perform better than focal stack on depth estimation, in the cases where the maximum disparity of the scene is small, e.g., small aperture camera, or far away objects. In such cases, it turns out that the more compact representation of the scene in the form of a focal stack is better suited for a neural network to estimate the depth.

6.4 Summary

This chapter explored the focal stack camera design parameter space, including the number of focal planes, size of the aperture and sensor resolution, and studied their effects on the depth estimation performance, using three public light field datasets and an experimentally acquired Nikon focal stack dataset. We further compared the focal stack camera performance with the light field camera and showed that which one is better for depth estimation depends on the maximum disparity of the scene. These findings can be helpful for future designs of focal stack cameras.

CHAPTER VII

Secure Imaging using Focal Stack Camera

7.1 Introduction

Previous chapters have presented 3D sensing applications of the focal stack camera. This chapter, in contrast, concerns the application of using focal stack camera for secure imaging applications. That is, to detect faked or manipulated images using a focal stack.

Digital images are convenient to store and share, but they are also susceptible to malicious manipulations. With common photo editing tools, little effort or expertise are needed to convincingly manipulate an image. With the advancement of deep learning, this issue becomes even more severe: Generative Adverserial Networks (GAN) are able to synthesize realistic non-existing images, change the style of an image, or inpaint an image to remove specific objects in it. Deepfakes can even seamlessly swap the face of one person with another in images [1, 49]. These malicious manipulated images could appear in the news, causing misleading opinions in the public or being provided in the court as evidence, with obvious serious consequences.

Verifying the integrity of multi-media has been a research topic for long time in the field of multi-media forensics [25, 21, 23, 95, 85, 58]. Traditional methods verify the integrity of a digital medium and detect traces of malicious manipulation by examining some signatures in the image, using either passive or active approaches. In the active approach, semi-fragile watermarks are pro-actively embedded into the image. The introduced watermark (which is visually imperceptible) is persistent after benign image operations such as brightness adjustment, resizing and compression, but gets destroyed after malicious editing. In the passive approach, imaging artifacts such as those due to lens distortion [45], color filtering [85], Photo Response Non-Uniformity (PRNU) [58], or compression are used to authenticate an image.

Each method has its own limitations, however. The passive approach, while being simple to implement, relies on weak traces that are likely to be destroyed after compression/resizing. PRNU fingerprint analysis, while being a popular forensic method, requires knowledge about the source camera's PRNU. On the other hand, the active watermarking approach is more robust against compression/resizing, but alters the original content due to the watermark embedding. More recently, deep learning based forensic detection methods have also been proposed [41, 102, 108, 52]. However, the ability to generalize data-driven models remains as a key challenge: these models perform well on images that are similar to the training data, but the performance can quickly degrade when the models are fed with images that differ too much from the training data distribution [118, 22].

Most existing image forgery detection methods assume a standard camera and attempt to determine the image authenticity by analyzing features present in a given 2D image file. Adding security features directly on the hardware side can improve forgery detection. Motivated by this possibility, this paper proposes a new way to prevent and detect malicious image manipulation by enriching the information carried by the digital images and videos. Specifically, we propose to use a *focal stack*, instead of a single image, for secure media sharing, where the entire focal stack image file is shared publicly. Fig. 7.1 illustrates the idea: images in the focal stack contain depth dependent defocus blur. Since generating physically realistic content with defocus blur that is consistent across the focal stack is extremely challenging, we show that detecting image manipulation is much easier for a focal stack compared to a single image, by using such inter-focal stack consistency cues. This approach leads to a much more secure media format. Someone attempting to manipulate the image would have to do it for every image in the focal stack, and it would be extremely challenging to accomplish this in a way where the consistencies of the content and the defocus blur are maintained across the focal stack.

To demonstrate the advantage of focal stack image sets over single 2D images as a tamper-evident image file, we limit our scope to inpainting types of image manipulation. We generated inpainted focal stacks using several CNN-based methods [71, 106, 111]; we then trained inpainting region localization CNNs to detect regions in the focal stack that are inpainted. We show that the focal stack based method achieves significantly better detection performance and generalization ability, compared to single image based methods. We further study how detection performance depends on the number of images in the focal stack and also whether the performance gain of using a focal stack might be mainly due to increased total pixel

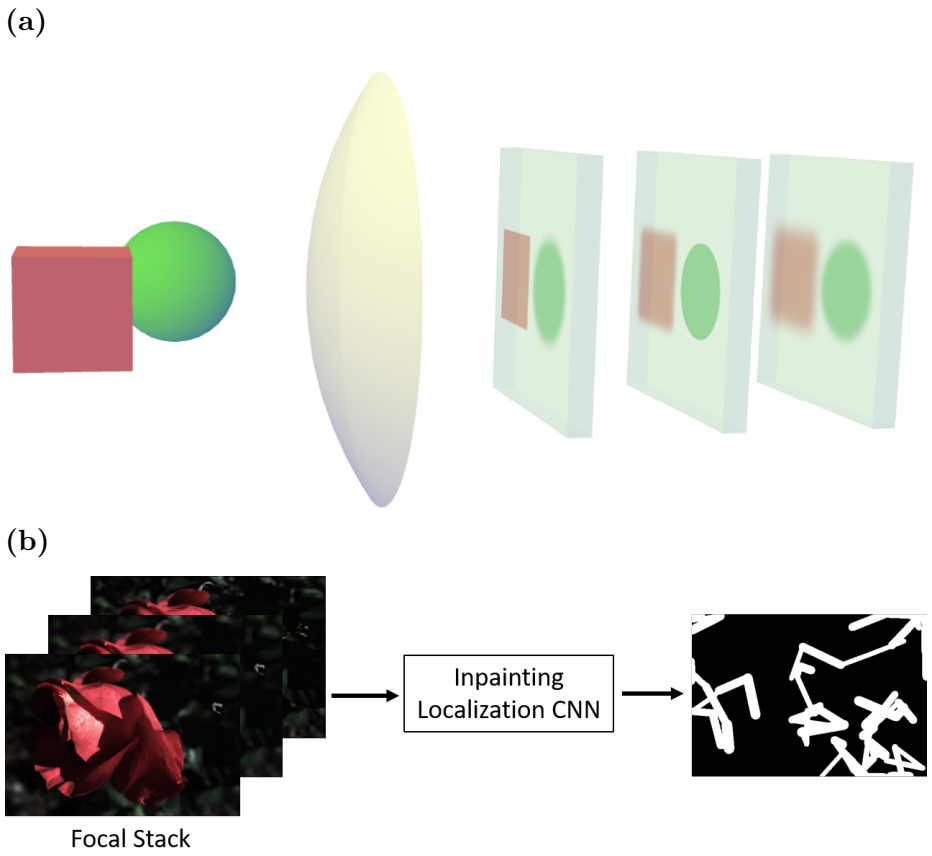

Figure 7.1. Focal stack system for inpainting region localization. (a) Imaging system schematic showing depth dependent defocus blur of a cube-ball object. (b) Inpainting localization CNN estimates inpainting regions from a focal stack.

number.

This chapter is organized as follows: section 7.2 describes related work on image inpainting, forgery localization and focal stack cameras. Section 7.3 describes the method we used to generate inpainted focal stacks and the method to localize inpainted regions. Section 7.4 presents multiple numerical experiments and results. This work has been submitted to *IEEE Transactions on Image Processing* for peer review.

7.2 Related work

7.2.1 Image inpainting

Traditional image inpainting methods work well on highly textured or patterned regions, but fail on inpainted regions with rich context and semantic meaning, such as natural scenes and human faces. Simakov et al. proposed a bidirectional similar-

ity measure, a metric based on nearest neighbor patch search, to determine if two signals are similar and can be used as the objective function for image inpainting. PatchMatch [6] accelerated the patch matching process in the bidirectional similarity measure using random search and coherence propagation. Shift-Map [86] achieved inpainting by computing a shift-map, where the pixels in the inpainting region are sampled from a relative position indicated by the shift-map. The shift-map is estimated by a global optimization objective function that contains a data term and a smoothness term. The optimization is done in a hierarchical way to accelerate the computation, with low resolution shift-map estimated first and then refined by high-resolution one.

Deep learning based inpainting methods have better performance for inpainting complex objects and scenes due to their powerful capability for modeling the high level semantics presented in the image. The context encoder [78] is an early approach to image inpainting using deep learning methods. An encoder extracts semantic information from a masked input image, and a decoder reconstructs a full image with coherent contents filled in the inpainting region. Pixel-wise reconstruction loss and adversarial loss are used as the loss function to train the network. Later works typically follow this adversarial training to improve the fidelity of the inpainted region. GMCNN [106] used a multi-column network to inpaint missing regions at multiple-scales in parallel. A confidence driven pixel reconstruction loss is used to constrain filling boundary pixels more strictly, compared to those pixels that are far away from the boundary. A Markov Random Fields (MRF) type regularization promotes content diversity in the inpainting region. As standard convolution's response is conditioned on both valid pixels and also placeholder values in the inpainting region, it also leads to color discrepancies. To resolve this issue, Liu et al. [56] proposed partial convolution to reduce these artifacts by introducing a layer-wise binary valid mask to select out only valid pixels for convolution computation and to normalize the convolution output. Gated Convolution [111] further generalized the partial convolution by having a learnable gating mechanism to select only proper pixels for convolution. Nazeri et al. [71] divided the inpainting process into edge generation and colorization stages. In the first stage, the edges of the inpainting regions are first generated. Then the colorization network inpaints the region conditioned on the input image and also the edge map. Such proposed two-stage inpainting exhibits better details in the inpainting region. There has been continued progress on improving inpainting using deep learning methods. Li et al. proposed to use a recurrent feature reasoning module to improve the inpainting performance on large continuous holes. Yi et al. proposed

a contextual residual aggregation mechanism to inpaint ultra-high resolution images with good quality [109]. Peng et al. proposed to use a hierarchical vector quantized variational auto-encoder (VQ-VAE), to generate diverse inpainting results [80].

7.2.2 Forgery localization

Early attempts to localize manipulated regions in images relied on local anomalies of some signatures present in the image. Johnson et al. [45] analyzed the chromatic aberration presented in the image and identified the image regions where the chromatic aberrations are inconsistent with other regions in the image. Popescu et al. [85] showed that the color interpolation algorithm used for the color filter array in commercial cameras leads to periodic correlation patterns that can be revealed by Fourier analysis. They demonstrated that this signature can be used to localize tampered regions in an image. Assuming a known camera model or other reference images available, sensor pattern noise can also be used to localize a forged region by checking whether a region has such noise patterns [58]. In addition, splicing and copy-move forgery likely involves several post-processing steps, such as scaling/rotating the object and blurring the object/background boundary. These steps can generate re-sampling artifacts and can also be detected by spectral analysis [84].

Recent deep learning based methods, in contrast, learn discriminating forgery features from the data directly. Salloum et al. [92] trained a multi-task CNN (MFCN) for splicing localization. The network estimates both the splicing region and the splicing boundaries, with partially shared parameters between two tasks. Such multi-task design leads to better localization performance, compared to only estimating the splicing region. Huh et al. detected image splicing by training a classifier to determine whether two image patches have EXIF meta consistency [41]. Wang et al. [102] detected image warping manipulation by training a CNN on script-generated warped images in Photoshop. Wu et al. [108] proposed a two-branch CNN model (BusterNet) to localize copy-move forgery regions. Li et al. [52] localized inpainted regions by using a CNN model with the first few layers initialized as high-pass filters to enhance the inpainting traces. Despite these efforts, developing a well performing forgery detection method with good generalization ability remains as a challenge.

7.2.3 Focal stack

There are numerous applications of focal stack imaging as we have seen in previous chapters. However, to the best of our knowledge, there is no prior work using focal

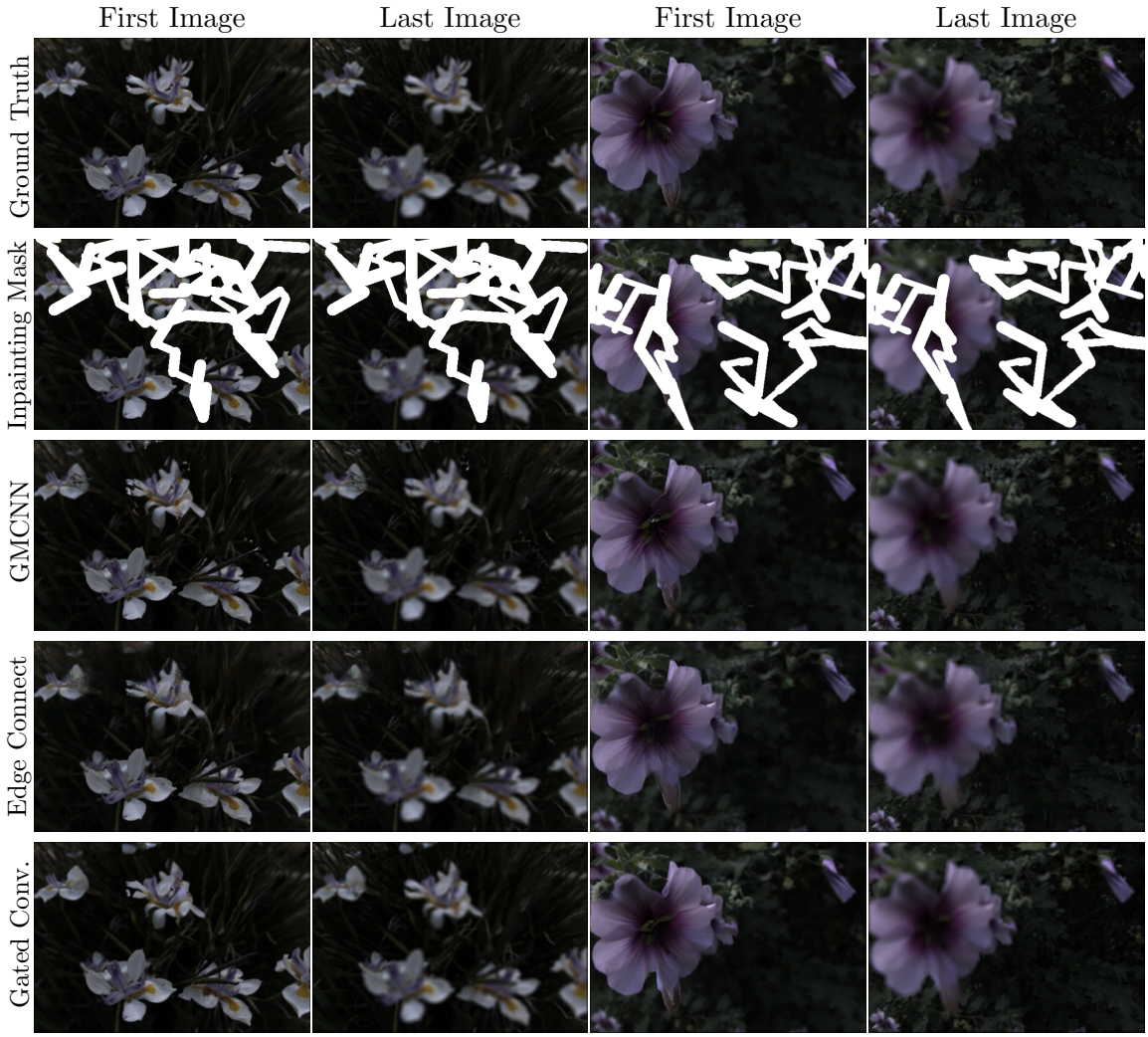

Figure 7.2. Example real and inpainted focal stacks. Only the first and the last image in each focal stack is shown. The region to be inpainted is shown as white in the second row.

stacks for image forensic related applications and this work is the first one to propose using focal stack imaging as a secure image format.

7.3 Method

To demonstrate the effectiveness of using focal stacks as a secure image format, we generated datasets containing manipulated focal stacks and trained a detection CNN to localize the forgery regions. The localization performance is then compared with single image based methods to show the advantage of focal stack over conventional images for image security applications. We focus on image inpainting forgery where

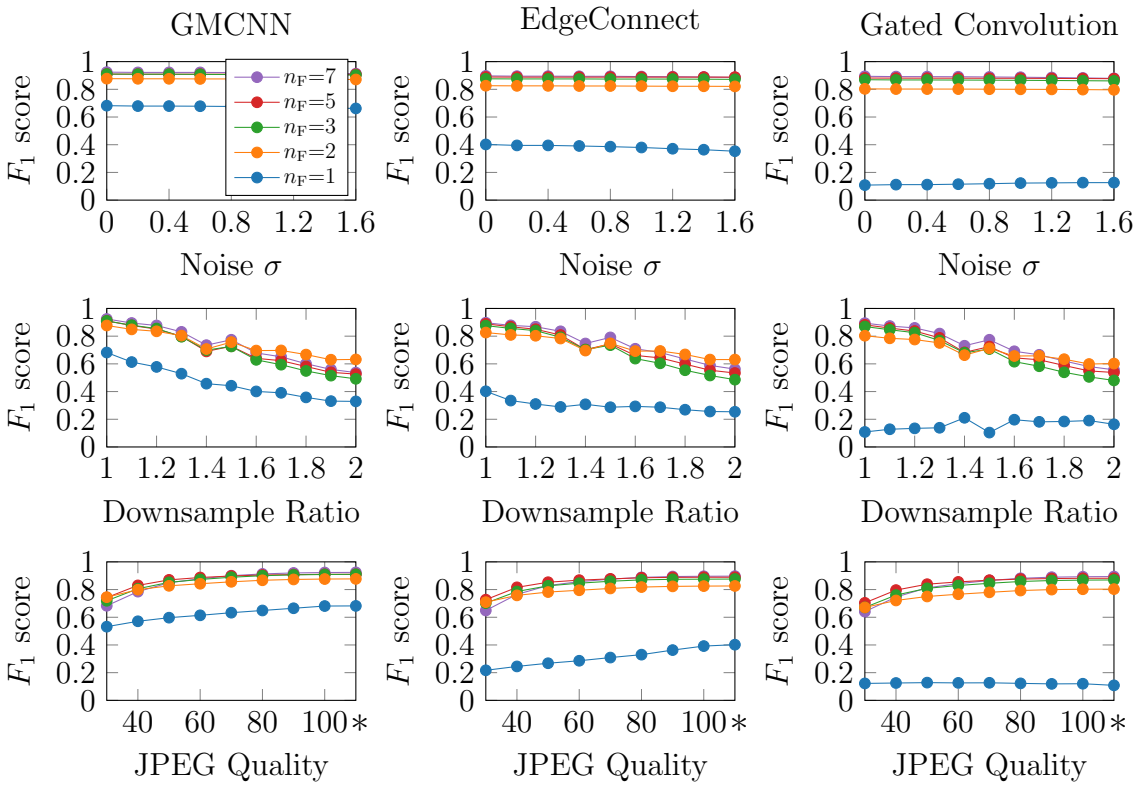

Figure 7.3. Localization F_1 scores for focal stack data with networks trained on GMCNN dataset with JPEG augmentation and tested on GMCNN data (1st column), EdgeConnect (2nd column) and Gated Convolution (3rd column) datasets. The robustness against Gaussian noise (1st row), resizing (2nd row) and JPEG compression(3rd row) are shown for each model. Symbol '*' on x-axis indicates the result without JPEG compression.

the inpainting is done by deep learning methods. Section 7.3.1 describes how we generate inpainted focal stacks using CNN methods. Section 7.3.2 describes how we localize inpainting regions in the manipulated focal stack.

7.3.1 Generating CNN inpainted focal stack

We first generated a set of authentic focal stacks from the Lytro flower light field dataset [97], using the add-shift algorithm [74]. The Lytro flower light field dataset contains 3343 light fields of flower scenes captured by Lytro Illum light field camera. Each light field has a size of $376 \times 541 \times 14 \times 14$, and following [97], we used only the central 8×8 sub-aperture images for focal stack generation. Each generated focal stack contains $n_F = 7$ images with differing focus positions. The focus positions are chosen to have their corresponding disparities evenly distributed in range [-1, 0.3], which covers roughly the entire possible object depth range. The first row of Fig. 7.2

shows example generated authentic focal stacks images.

Then we generated inpainted focal stack datasets, using three CNN based methods: GMCNN [106], EdgeConnect [71] and Gated Convolution [111]. GMCNN uses a multi-column network to extract features at different scale level. A special ID-MRF loss is designed to promote the diversity and realism of the inpainted region. EdgeConnect is a two-stage inpainting process. In the first stage, an edge generator generates edges for the inpainting region. In the second stage, an inpainting network fills the missing region with the help of the completed edges from the first stage. Gated Convolution [111] uses a learnable feature gating mechanism to solve the issue that a normal convolution treats both all pixels equally and inpaints the image following a two-stage coarse to fine process. We generated inpainted focal stacks using multiple methods to test the generalization ability of the network; we train the detection network using focal stacks inpainted by one method and then evaluate its performance on focal stacks inpainted by another method. This investigation mimics the more realistic scenario where the method used to inpaint the focal stack is unknown at the time of detection.

We generated random stroke-type regions to be inpainted for each focal stack. All images in the same focal stack shared the same spatial inpainting region. The goal of inpainting is typically trying to hide something in the original image and hence identical inpainting region across images in the same focal stack should be a reasonable assumption. Each image is then inpainted independently using one of the above CNN methods.

The CNN inpainting models were pre-trained on the places2 [119] dataset using their original implementation and fined tuned on the flower focal stack dataset. Fig. 7.2 shows example inpainted focal stacks.

7.3.2 Detecting CNN inpainted focal stack

The detection network we used for localizing inpainting region is based on DeepLabv3 [15]. DeepLabv3 was originally proposed for semantic segmentation and we repurposed it for region localization due to the similarity in these two tasks. The Atrous Spatial Pyramid Pooling (ASPP) layer in DeepLabv3 ensures large receptive field and fine detailed network output at the same time, which is beneficial for our inpainting region localization. We used ResNet-18 [36] as the backbone for feature extraction. A normal input image to the DeepLabv3 is a 3D tensor of shape (C, H, W), whereas focal stack is a 4D tensor of shape $(n_{\mathrm{F}}, C, H, W)$, so we reshaped the focal stack to be $(n_{\mathrm{F}} \times C, H, W)$ by concatenating images along the color channel. The network

outputs a pixel-wise probability map that indicates whether a pixel is inpainted and we train the network using binary cross-entropy loss.

Wang et al. [103] showed that proper data augmentations, such as applying JPEG compression, lead to a model with better generalization ability and robustness against common post-processing. Motivated by this, we followed their approach and trained our detection network with JPEG augmentation. Specifically, the training input focal stacks have a 50% probability of being JPEG compressed, with a JPEG quality factor of 70. For reference, we also trained models without JPEG augmentation; these models performed worse so the results are shown in the appendix C.

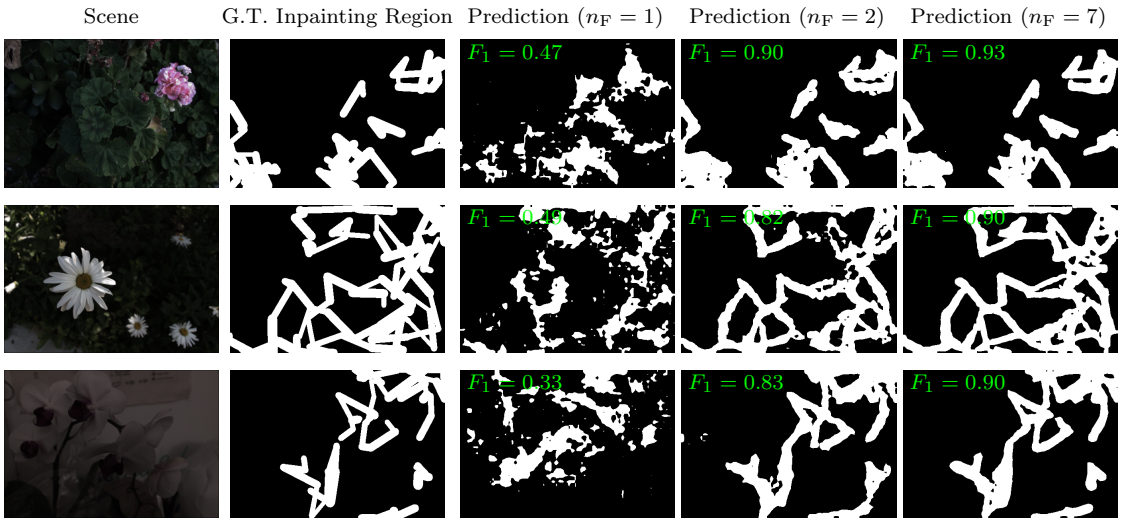

Figure 7.4. Example localization results of the model trained on GMCNN dataset and tested on Gated Convolution dataset. Probability threshold of 0.5 is used for classification. F_1 scores are indicated in green for each prediction.

7.4 Experiments and results

7.4.1 Implementation

The inpainted focal stack dataset generated from Lytro flower light fields contains 3343 focal stacks for each inpainting method (GMCNN, EdgeConnect, Gated Convolution). Each focal stack contains $n_F = 7$ images with changing focus depth and is associated with a ground truth inpainting region for training and evaluation. We used 2843 focal stacks for fine-tuning the inpainting networks and also training the detection network. The remaining 500 focal stacks are used for evaluating the inpainting localization performance.

We trained the detection network using Adam optimizer [48] with batch size 3. The models were trained for 110 epochs, with an initial learning rate 10^{-4} that was reduced to 10^{-5} after 70 epochs. We used data augmentation in the form of horizontal flipping with 50% probability, in addition to the JPEG compression augmentation described above.

We counted the true positive (TP), false positive (FP) and false negative (FN) predictions at the pixel level for each test sample, with the classification probability threshold set to 0.5. Then the F_1 scores, defined as $\frac{\mathrm{TP}}{\mathrm{TP}+\frac{1}{2}(\mathrm{FP}+\mathrm{FN})}$, were computed and averaged over all test samples to evaluate the network's inpainting localization performance.

We additionally tested the models' robustness against common post-processing methods including JPEG compression, gaussian noise, and resizing. Specifically, we added additive white gaussian noise with σ in range [0, 1.6] to test the robustness against noise. We downsampled test focal stacks using nearest neighbor interpolation with ratio in range [1, 2] to test the robustness against resizing. We JPEG compressed test focal stacks with JPEG quality in range [30, 100] to test the robustness against compression. Note that these post-processing processes are only applied to the test focal stacks; the models were trained using augmentation based only on horizontal flipping and JPEG compression with quality 70.

To study the dependence of the localization performance on the focal stack size n_F, we trained models using inpainted focal stack datasets with $n_\mathrm{F} = 1, 2, 3, 5, 7$. Specifically, the $n_\mathrm{F} = 7$ dataset is the one described at the begining of this section. We obtained the $n_\mathrm{F} = 1$ dataset by only using the 7th (last) image of each focal stack in $n_\mathrm{F} = 7$ dataset. Similarly, the $n_\mathrm{F} = 2$ dataset contains the 1st and 7th images, the $n_\mathrm{F} = 3$ dataset contains the 1st, 4th, 7th images, and the $n_\mathrm{F} = 5$ dataset contains the 1st, 3rd, 4th, 5th and 7th images.

7.4.2 Results

Fig. 7.3 shows the localization results trained on the GMCNN inpainted focal stack dataset and evaluated on testing focal stacks inpainted by GMCNN, EdgeConnect and Gated Convolution. The advantage of using focal stack ($n_\mathrm{F} \geq 2$) over single image ($n_\mathrm{F} = 1$) for inpainting region localization is apparent and significant for every test configuration. Taking the 1st row of Fig. 7.3 for example, training and testing both on the GMCNN dataset using $n_\mathrm{F} = 1$ has a F_1 score about 0.67 and using $n_\mathrm{F} = 2$ has a F_1 score about 0.87. The difference is even more dramatic when training is performed on the GMCNN dataset and testing is performed on the Gated

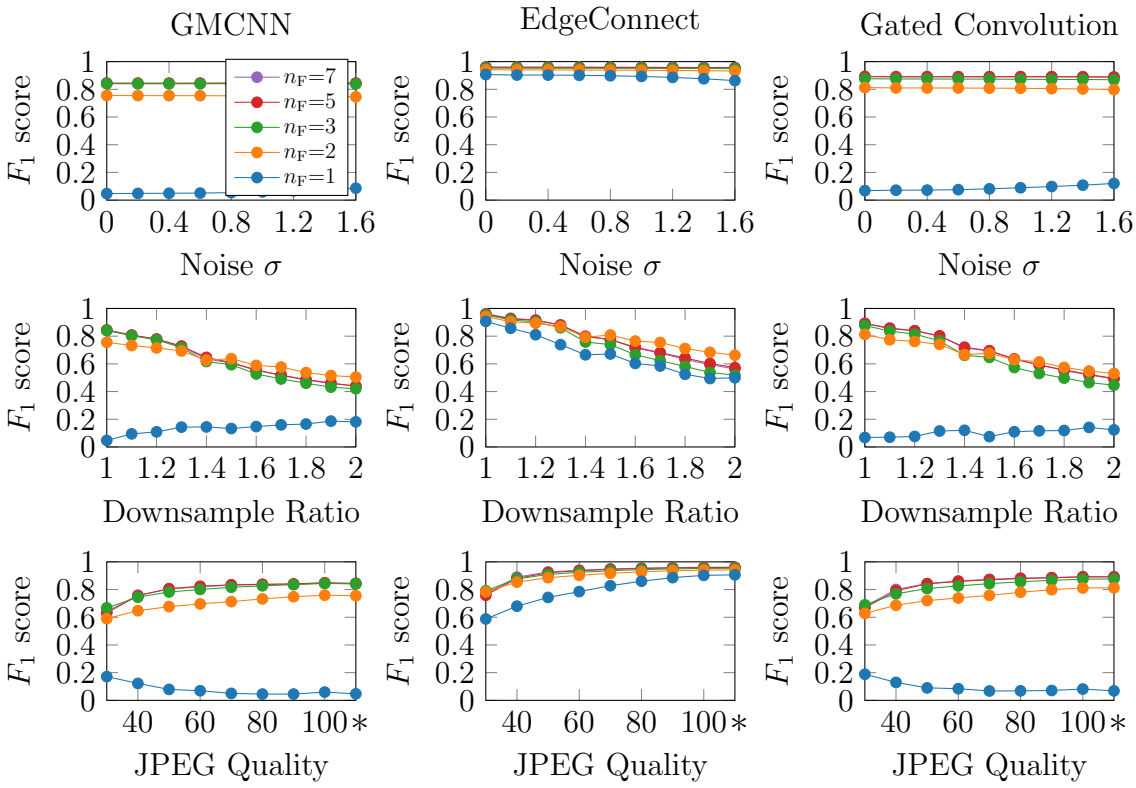

Figure 7.5. Localization F_1 scores for focal stack data with networks trained on EdgeConnect dataset with JPEG augmentation and tested on GMCNN (1st column), EdgeConnect (2nd column) and Gated Convolution (3rd column) datasets. Symbol '*' on x-axis indicates the result without JPEG compression.

Convolution dataset (top-right subplot): $n_F = 1$ has a F_1 score about 0.11 and using $n_F = 2$ has a F_1 score about 0.80. Increasing n_F further improves the F_1 score, though not significantly. Although the single image ($n_F = 1$) localization method performs fairly well when the testing data are generated by the same inpainting method as the training data, it performs poorly when the testing data are inpainted by a different method. On the other hand, *there is only a very small performance drop for the focal stack based method when testing on focal stacks inpainted by a method different from training*. These results show that the focal stack based method has a much better generalization ability across different inpainting methods. This benefit can be understood as follows: for single image based inpainting region localization, the network relies heavily on detecting inpainting method specific artifacts, such as checkerboard patterns produced by transpose convolutions [76] or unnatural transitions between inpainted and not inpainted regions, to determine whether a region is inpainted. However, these criteria cannot be universal for detecting inpainting because a differ-

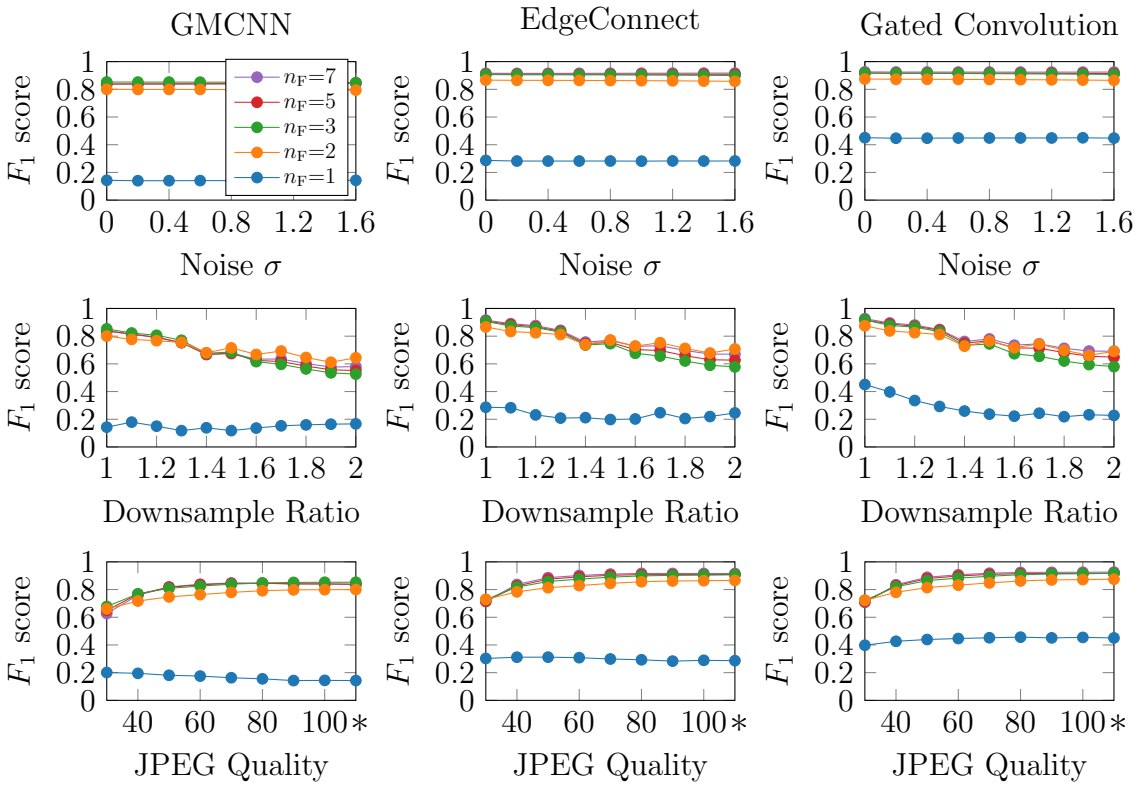

Figure 7.6. Localization F_1 scores for focal stack data with networks trained on Gated Convolution dataset with JPEG augmentation and tested on GMCNN (1st column), EdgeConnect (2nd column) and Gated Convolution (3rd column) datasets. Symbol '*' on x-axis indicates the result without JPEG compression.

ent method will likely have a different checker board pattern or a different transition artifact between inpainted and not inpainted region. On the other hand, the focal stack based method has a much more inpainting-method agnostic clue to determine whether a region is inpainted or not: it can check whether the content and the defocus blur across a focal stack in a region is physically and semantically consistent. Such consistency checks do not depend on the methods used for inpainting and hence it should better generalize across different inpainting methods.

Fig. 7.4 shows example predicted inpainting regions, using a model trained on GMCNN inpainted focal stacks and tested on Gated Convolution inpainted focal stacks. The single image based inpainting localization performs poorly, whereas using a focal stack of only $n_F = 2$ greatly improves the prediction and $n_F = 7$ model has the best performance.

We also trained models using EdgeConnect inpainted focal stacks, and using Gated Convolution inpainted focal stacks, to verify that the trends above are not specific to

the particular training dataset. Fig. 7.5 and Fig. 7.6 show the results. The general findings are similar as those from Fig. 7.3, with some minor differences: the advantage of a focal stack over a single image for the model trained and tested on EdgeConnect inpainted dataset is smaller, as shown in the middle column of Fig. 7.5. This is likely because the EdgeConnect inpainted images contain more visually apparent inpainting artifacts. Indeed, when we inspect closely some EdgeConnect inpainted regions, they tend to be darker, compared to non-inpainted regions. This makes inpainting localization using single image easier so the additional images in the focal stack do not help much. However, when the model is evaluated on the dataset inpainted by a method different from the training data, the single image localization performance degrades severely, as shown in the 1st and 3rd column of Fig. 7.5, while the focal stack based models retain high performance in these cases. This is again because the focal stack based method uses the more generalizable inter-focal stack consistency check to localize the inpainting region. For models trained on Gated Convolution, the single image based method performs poorly (3rd column of Fig. 7.6), even when tested on focal stacks inpainted by the same method. This is because the Gated Convolution inpainted images contain fewer artifacts and are more visually realistic. This makes the single image based method struggle to find discriminating forgery traces.

All results presented in Fig. 7.3, Fig. 7.6 and Fig. 7.5 demonstrate good robustness against several post-processing methods, including Gaussian noise (1st row), image resizing (2nd row) and JPEG compression (3rd row), showing that our proposed method would be useful in practical cases, such as in determining whether an internet image file is authentic or not, where these post-processing operations are common.

To verify that the advantage of a focal stack over a single image is not simply due to the increase in the number of total pixels, we trained additional models for $n_{\mathrm{F}} = 2$, using focal stacks downsampled by factors of $\sqrt{2}$ and 2. Fig. 7.7 shows the results. The $n_{\mathrm{F}} = 2$, downsampling ratio $= \sqrt{2}$ system has the same total number of pixels as $n_{\mathrm{F}} = 1$ system without downsampling, and $n_{\mathrm{F}} = 2$, downsampling ratio $= 2$ model has two times fewer total pixels, compared to the system of $n_{\mathrm{F}} = 1$, without downsampling. Fig. 7.7 shows that reducing the total pixel numbers in the focal stack system only slightly reduces the localization performance; the main performance gain of using a focal stack for inpainting localization is due to the multiple sensor plane nature of the focal stack system that encodes robust inter-focal stack consistency clues for forgery detection.

In practical applications, the testing focal stack to be authenticated may have a different focus setting than the training time focus setting. Thus, in Table 7.1 we

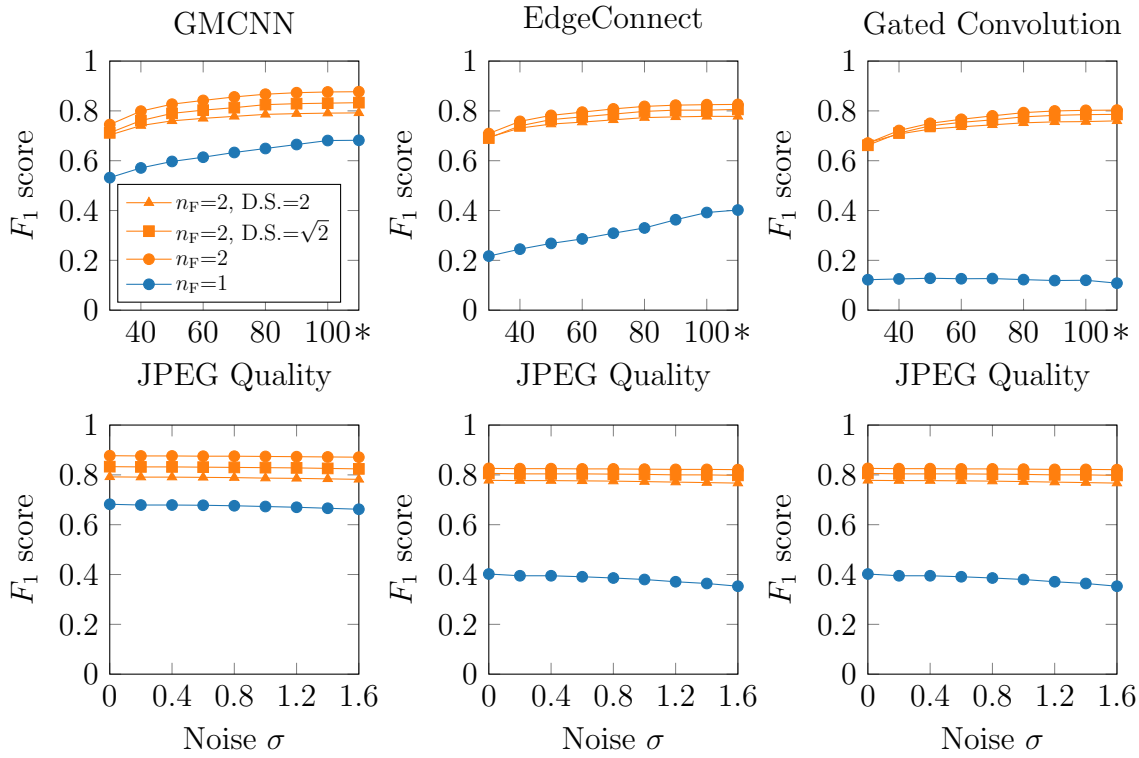

Figure 7.7. Localization F_1 scores for focal stack data with networks trained on GMCNN dataset with JPEG augmentation and tested on GMCNN (1st column), EdgeConnect (2nd column) and Gated Convolution (3rd column) datasets, showing the total pixel dependence. Symbol '*' on x-axis indicates the result without JPEG compression.

Table 7.1. F_1 scores of the model trained on GMCNN inpainted focal stacks with focusing disparity range [-1, 0.3], and evaluated on focal stacks inpainted by GMCNN, EdgeConnect and Gated Convolution. Three values in each field correspond to the results on focal stacks with focusing disparity range [-1, 0.3], [-0.8, 0.5] and [-1.2, 0.5], respectively.

n_F	GMCNN	EdgeConnect	Gated Convolution
1	0.68 / 0.66 / 0.66	0.40 / 0.37 / 0.37	0.11 / 0.10 / 0.10
2	0.88 / 0.87 / 0.87	0.83 / 0.82 / 0.81	0.80 / 0.79 / 0.79
3	0.91 / 0.91 / 0.85	0.88 / 0.87 / 0.82	0.87 / 0.86 / 0.80
5	0.91 / 0.92 / 0.89	0.89 / 0.89 / 0.86	0.88 / 0.89 / 0.85
7	0.92 / 0.92 / 0.90	0.90 / 0.89 / 0.87	0.89 / 0.89 / 0.87

also evaluated our model using inpainted focal stacks having a different focus setting compared to the training time. Specifically, the model is trained using GMCNN inpainted Lytro flower focal stacks, with focusing disparity evenly distributed in range [-1, 0.3], and tested on Lytro flower focal stacks with focusing disparity evenly distributed in range [-1, 0.3] (same setting as training), and in the ranges [-0.8, 0.5], and [-1.2, 0.5]. The case [-0.8, 0.5] corresponds to the scenario where every image in the testing focal stack is focusing closer to the camera and the case [-1.2, 0.5] corresponds to the scenario where the focus depth range is larger for the testing data compared to the training data. The table shows that there is only a slight drop in inpainting localization performance when testing the trained focal stack based model on focal stacks with different focus setting. This excellent generalization ability across camera focus settings is due to the fact that the focal stack based model relies on the inter-focal stack consistency for detection, which is insensitive to the focus of each image.

Finally, as performance references, we also evaluated the forgery localization F_1 scores by predicting all pixels to be forged (all-forged) and by predicting a pixel to be forged or not with 50 % probability (flip-coin). The all-forged method has a F_1 score of 0.38 and the flip-coin method has a F_1 score of 0.32. Note that these values are higher than some F_1 scores of $n_{\mathrm{F}} = 1$ models evaluating on the unseen datasets. For example, the $n_{\mathrm{F}} = 1$ model trained on the EdgeConnect dataset and tested on the Gated Convolution dataset (top-right panel of Fig. 7.5) at $\sigma = 0$ has a F_1 score of 0.07. This is because the F_1 score is a harmonic mean of precision and recall: $F_1 \triangleq \frac{\mathrm{TP}}{\mathrm{TP}+\frac{1}{2}(\mathrm{FP}+\mathrm{FN})} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$. The all-forged method has a precision of 0.24, a recall of 1, the flip-coin method has a precision of 0.24, a recall of 0.5, and the $n_{\mathrm{F}} = 1$ model at $\sigma = 0$ has a precision of 0.71, a recall of 0.04. As can be seen, although the $n_{\mathrm{F}} = 1$ model has a much higher precision, it's very conservative at predicting forged samples, hence a very low recall value (note that in the case where all pixels are predicted to be real, the F_1 score is 0). The $n_{\mathrm{F}} = 1$ model is essentially predicting almost all pixels to be real since the forgery trace in the unseen dataset is different from the training time and the model struggles to find such traces. This fact could also explain the reason why the $n_{\mathrm{F}} = 1$ trace in the top-right panel of Fig. 7.5 increases slightly with σ: an inclusion of the noise leads to some spurious features in the image, leading to an increased number of pixels to be predicted as forged, which in turn leads to an increased recall. For example, the $n_{\mathrm{F}} = 1, \sigma = 1.6$ point in the same figure has a recall of 0.07, a precision of 0.47, which corresponds to an increased F_1 of 0.12.

7.5 Summary

This chapter proposed a novel system and method of using a focal stack for localizing image inpainting regions in manipulated images. We trained CNN models for inpainting localization and showed that using an image focal stack, instead of a single image, leads to significantly better localization performance and significant robustness to common post-processing image perturbations. The proposed method also shows excellent generalization ability across different inpainting methods and different camera focus settings.

Although we focused on the inpainting type of forgery, we expect the findings are applicable to many other types of forgery detection as well. We hope this work can lead to a new direction for image forgery detection and make images in the future more secure.

CHAPTER VIII

Conclusions and Future Work

We have presented two imaging systems enabled by novel nanophotonic devices, i.e., a HMM-based nanoscale structure fingerprinting system and a transparent graphene based focal stack camera. Following sections summarize the result of each project and discuss the future directions.

8.1 Nanoscale fingerprinting with hyperbolic metaterials

Chapter II presented a new approach of discriminating nanoscale objects using HMM, with deep subwavelength resolution. Instead of imaging directly the nanoscale objects, we proposed to measure a far-field scattering spectrum of the objects placed on the HMM device. Thanks to the highly localized beam profile in the HMM, the measured spectrum is extremely sensitive to the spatial/material configuration of the objects under examination, and can be used as the fingerprinting to discriminate different nanoscale structures. We demonstrated results of localizing a single nanoscale object, determining the gap between two closely spaced objects, and also showed the dependence of the spectrum on the material composition. Importantly, our proposed method only relies on the far-field intensity-only measurement to achieve a deep-subwavelength resolution. Unlike fluorescence based method imaging method, it doesn't require fluorescent labeling process. And compared to hyperlens imaging approach, where the high loss of the metamaterial demands a high intensity illumination, which in turn could damage the sample, our proposed device works using the localized light beam that is already attenuated by the HMM. This makes it possible to increase the illumination level without damaging the sample. Future works of this project include studying the resolution dependence on the HMM top scatterer configuration, demonstrating nanoscale fingerprinting applications experimentally in 3D configuration.

8.2 Learning based light field reconstruction

Chapter III presented methods for reconstructing the light field from focal stack. We first presented the Momentum-Net, an iterative network network based method obtained by unrolling the Block Proximal Extrapolated Gradient Method (BPEG-M), to solve inverse problems. Compared with reconstruction using hand-crafted 4D EP regularization, Momentum-Net improves significantly the reconstruction quality. And compared with existing iterative network based method (BCD-Net with 3 inner iteration), it is 2.5 times faster. Possible future works of this project include designing a sharper majorizer to further improve the reconstruction speed and accuracy and learning Momentum-Net regularization parameters from datasets during the training stage.

However, as an iterative reconstruction method, reconstruction from Momentum-Net is still not applicable to real-time applications. As a result, a non-iterative light field reconstruction method is highly favored. To this end, in the latter part of Chapter III, we proposed a learning based non-iterative light field reconstruction method. The method reconstructs the light field using physics-based rendering using CNN estimated depth and all-in-focus image. The PSNR of reconstructed light fields on the Lytro flower light field dataset is 2.85 dB higher than the MBIR using 4D EP regularization and runs at a frame rate of 16 fps, making real-time light field photography from focal stack possible. The proposed model estimates an intermediate 4D ray depth from the focal stack, but is trained using light field as supervision. Hence it can also be used as a method for depth estimation from focal stack where the depth ground truth is not available. Future works of this project include designing a better all-in-focus image synthesize network, and using an occlusion-aware light field reconstruction loss to improve the reconstruction performance.

One important aspect of these proposed learning based reconstruction methods, is the stability of the reconstruction, e.g., the robustness against small perturbations in the focal stack. Such issue has been realized and studied in deep learning based image classification tasks [63, 99, 47], and also in the medical image reconstruction tasks [5]. It is possible that a small perturbation, even visually imperceptible, in the input to the network leads to a very different prediction. Hence it would be interesting to investigate the robustness of our proposed method against such perturbations.

8.3 Unsupervised depth estimation from focal stack

Since a typical learning based depth estimation method requires depth supervision, which is not easily available for many applications, chapter IV proposed an unsupervised depth estimation method from focal stack. A CNN is trained to estimate a depth map using focal stack reconstruction loss, hence avoiding the need of the depth ground truth. We compared the proposed method performance with single image based unsupervised depth estimation method and also with focal stack based supervised depth estimation method. Although it performs worse compared to supervised depth from focal stack method, it has significantly better depth quality over single image based unsupervised depth estimation, indicating the benefits of using focal stack as a depth sensing component.

In our proposed method, using the input focal stack, we reconstruct the focal stack using the estimated depth map and the estimated all-in-focus image. We found that the quality of the estimated all-in-focus image plays an important role in the final depth estimation performance. Although the all-in-focus image is estimated using a classical Laplacian focus measure in the current proposed pipeline, it is possible to train an unsupervised CNN to estimate the all-in-focus image from the focal stack, which is one of the future work of this project.

8.4 Focal stack based 3D tracking

Chapter V introduced a method of 3D tracking using the novel focal stack camera. The project is motivated by the fact that in many applications, a dense depth map or a time-consuming light field reconstruction is not necessary and only sparse object location information is needed. To this end, this chapter designed neural networks to accomplish the 3D tracking, and using the proof-of-concept low resolution focal stack of 4×4 with $n_{\mathrm{F}} = 2$, we demonstrates experimentally accurate 3D position tracking of point object. We further showed that it is possible to track an extended object using higher resolution focal stacks and also at the same time estimate its orientation.

In the current method, the algorithm only localize a fixed number of objects. One future work could be to allow the algorithm to detect and localize a variable number of objects in the camera field of view, which is more close to the practical application 3D sensing scenarios.

8.5 Focal stack camera design exploration

Chapter VI explored the camera parameter's dependence, including aperture size, sensor resolution and number of sensor planes on its depth estimation performance. It shows that increasing the aperture size and sensor resolution leads to lower depth estimation error and using more sensor planes is helpful when the sensor resolution is low. We also conducted a performance comparison between the focal stack camera and the light field camera on several datasets. The results indicates that the focal stack camera performs better than the light field camera on the scenes with small disparity. This work provides guidelines for future focal stack camera design and indicates suitable application scenarios for focal stack camera.

8.6 Secure imaging using focal stack camera

Chapter VII proposed to use the focal stack camera for secure imaging purpose. We showed that using a focal stack, instead of a single image, leads to much more robust inpainting-type forgery detection and localization. In contrast to single image based method, the focal stack based forgery detection maintains good performance when the manipulated images are JPEG-ed, noise-corrupted or resized. In addition, the single image based forgery detection method fails quickly when the model is evaluated on a different dataset, while the focal stack based forgery detection generalizes well on unseen datasets. Importantly, the performance gain of using a focal stack of $n_{\mathrm{F}} = 2$ over a single image ($n_{\mathrm{F}} = 1$) is already very significant. There are multiple possible future works on this project. Firstly, since the image security is always a rivalry between faking and detecting techniques, one interesting direction of future work is to explore possible ways to fool the proposed focal stack based forgery detection method. Potentially viable ways to achieve this include learning to synthesize the entire focal stack jointly while encouraging and observing the inter-focal stack consistency, instead of synthesizing each image independently; generating a very photo-realistic 3D model of the scene and then rendering realistic focal stack from it. In addition, it would be interesting to study forgery detections in focal stack videos, instead of focal stack images, and evaluating the focal stack based method on more types of image forgery, including image splicing, deepfakes, etc.

APPENDIX A

Nanoscale Fingerprinting with Hyperbolic Metamaterials

A.1 Volume plasmon polariton modes in HMM

The proposed 2D uniaxial HMM structure can support both ordinary modes (transverse electric polarized or TE polarized for short) and extraordinary modes (transverse magnetic polarized or TM polarized for short). The new property of the HMM originates from the extraordinary modes, which is also known as volume plasmon polariton (VPP) modes in the context of HMM [43]. The dispersion relation of the extraordinary modes is given by:

$$\frac{k_\perp^2}{\varepsilon_\parallel} + \frac{k_\parallel^2}{\varepsilon_\perp} = \frac{\omega^2}{c^2}, \tag{A.1}$$

where $k_\parallel$ and $k_\perp$ are respectively the components of wave vector parallel and perpendicular to the optical axis; $\varepsilon_\parallel$ and $\varepsilon_\perp$ are the structure's effective permittivity tensor components along the optical axis and perpendicular to the optical axis (a calculation of $\varepsilon_\parallel$ and $\varepsilon_\perp$ is given in Supplementary material section A.2); ω is the angular frequency of wave and c is the speed of light.

Note that from eqn. A.1, we have $k_\perp^{\min} = \sqrt{\varepsilon_\parallel}\frac{\omega}{c}$ (when $k_\parallel = 0$). Since $\varepsilon_\parallel > 6$ (Fig. A.1), $k_\perp^{\min} > \frac{\omega}{c}$. As a result, any propagating modes in air and evanescent components with $k_\perp < k_\perp^{\min}$ are totally reflected at the HMM/air interface.

When the top scatterer of the HMM device is illuminated by a TM polarized plane wave, the scattered field from the top scatterer excites many VPP modes with different

wavevectors and the localized beam inside the HMM is a coherent superposition of these modes. Hence its propagation angle is determined by the group velocity direction, i.e., the normal of the iso-frequency curve. Since the asymptote of the hyperbolic dispersion curve has slope $\sqrt{\frac{|Re\varepsilon_{\perp}(\lambda)|}{\varepsilon_{\parallel}(\lambda)}}$ according to eqn. A.1, evaluating the group velocity at $k_{\perp} = \infty$ leads to beam angle $\theta(\lambda) = \tan^{-1}\left(\sqrt{\frac{\varepsilon_{\parallel}(\lambda)}{|Re\varepsilon_{\perp}(\lambda)|}}\right)$.

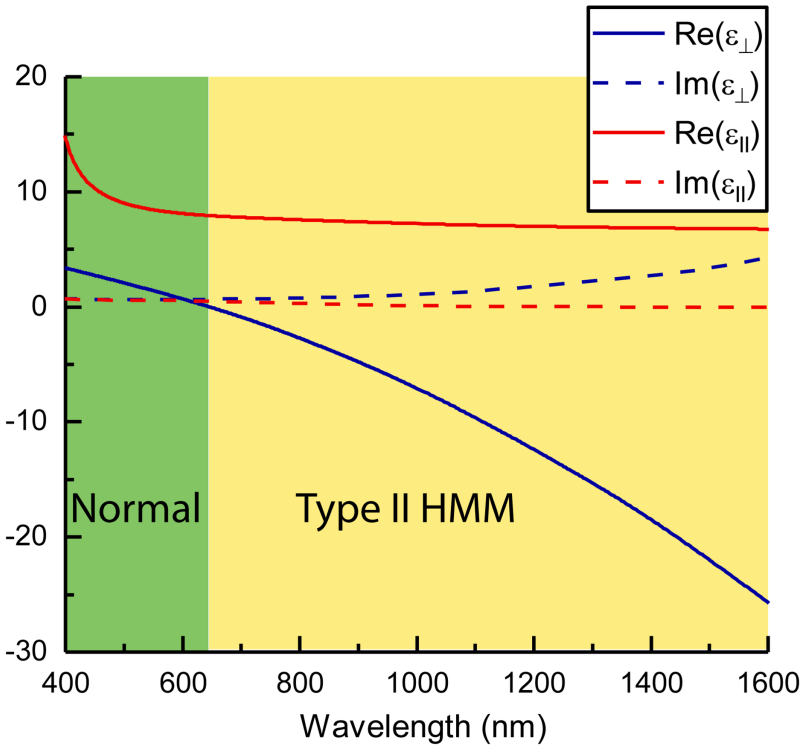

Figure A.1. Effective medium theory calculation of the permittivity tensor components, using eqn. A.2. Device behaves as type II HMM ($Re(\varepsilon_{\perp}) < 0$, $Re(\varepsilon_{\parallel}) > 0$)) in the yellow shaded region and as normal anisotropic medium ($Re(\varepsilon_{\perp})$, $Re(\varepsilon_{\parallel}) > 0$)) in the green shaded region.

The assumption of $k_{\perp} \gg \frac{\omega}{c}$ in calculating beam angle θ is valid for small scatterers. This also holds for the structure we are considering. This can be verified by examining Fig. 2.2(b) in the main text: by only decreasing the unit cell size, the beam angle obtained from the exact simulation approaches the EMT case, where $k_{\perp} \gg \frac{\omega}{c}$ is assumed.

A.2 Effective medium theory (EMT) description of the permittivity tensor

In the effective medium theory (EMT) limit, the structure's permittivity tensor components along the optical axis ($\varepsilon_{\parallel}$) and perpendicular to the optical axis ($\varepsilon_{\perp}$) are given by:

$$\varepsilon_{\perp} = r\varepsilon_m + (1-r)\varepsilon_d, \;\; \varepsilon_{\parallel}^{-1} = r\varepsilon_m^{-1} + (1-r)\varepsilon_d^{-1}, \tag{A.2}$$

where ε_m and ε_d are respectively the permittivity of metal and dielectric layers of thickness d_m and d_d, and r is the filling ratio of the metal given by $r = d_m/\left(d_m + d_d\right)$. Figure A.1 shows the results of the EMT calculation of the HMM permittivity tensor components using equation A.2. As discussed in the main text, it can be seen that $\mathrm{Re}\left(\varepsilon_{\perp}\right) < 0$ and $\mathrm{Re}\left(\varepsilon_{\parallel}\right) > 0$ for wavelengths larger than 647 nm and the structure behaves as type II HMM.

A.3 Calculation of the scattering strength

Here we present the results of our calculation of the scattering strength, which is defined as the ratio of the scattered power in the system with the target to the corresponding scattered power in the system without the target. The motivation for introducing this quantity is to compensate for the wavelength dependent scattering efficiency of the top scatterer. Given this definition, any fluctuation in scattering strength versus wavelength will then be mainly due to the varying degree of interaction between the localized beam and the bottom target only. The peak in the scattering strength then directly corresponds to the case of maximum interaction, i.e., when the localized beam is towards the bottom target. The scattered power versus wavelength without bottom target is shown in Fig. A.2 , which is used for all calculations of scattering strength.

A.4 Target material dependence of the scattering strength

In the main text, we claimed that the scattering strength increases as the refractive index of the target to the air is increased. Fig. A.3 shows an example calculation illustrating this point. Among the materials that are considered here, Ag has the highest index contrast to the air; hence it has largest scattering strength as expected. This material dependence of scattering strength serves as the basis for discriminating targets with different material composition demonstrated in the main text Fig. 2.2(c).

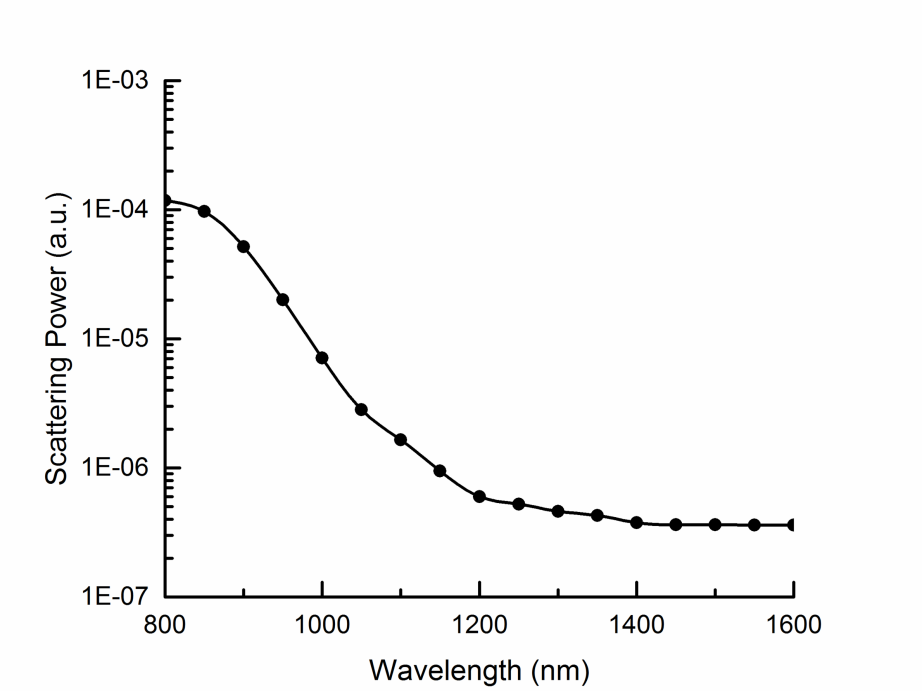

Figure A.2. The scattered power versus wavelength in the system without target.

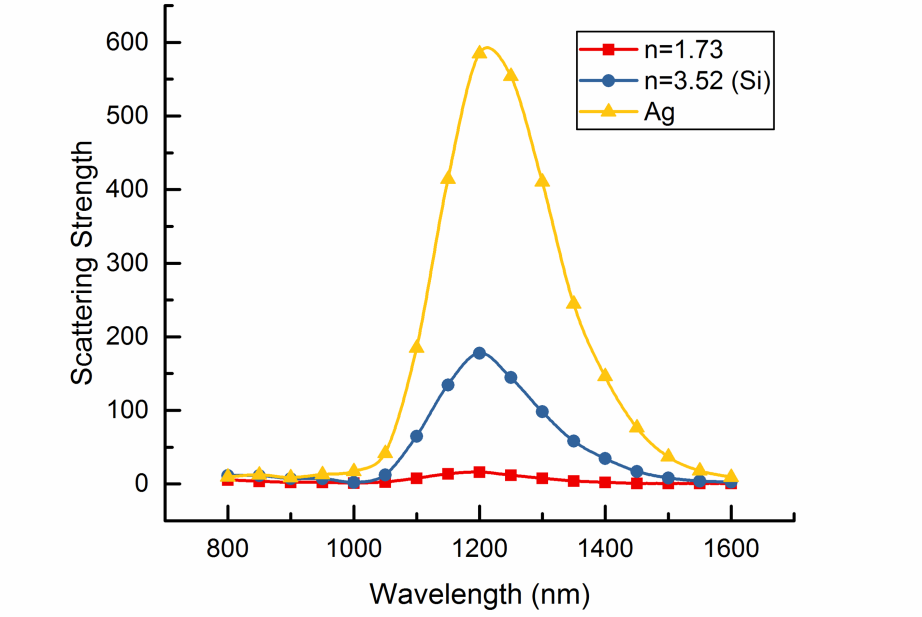

Figure A.3. Scattering strength versus wavelength for different bottom target material at fixed spacing of 400 nm.

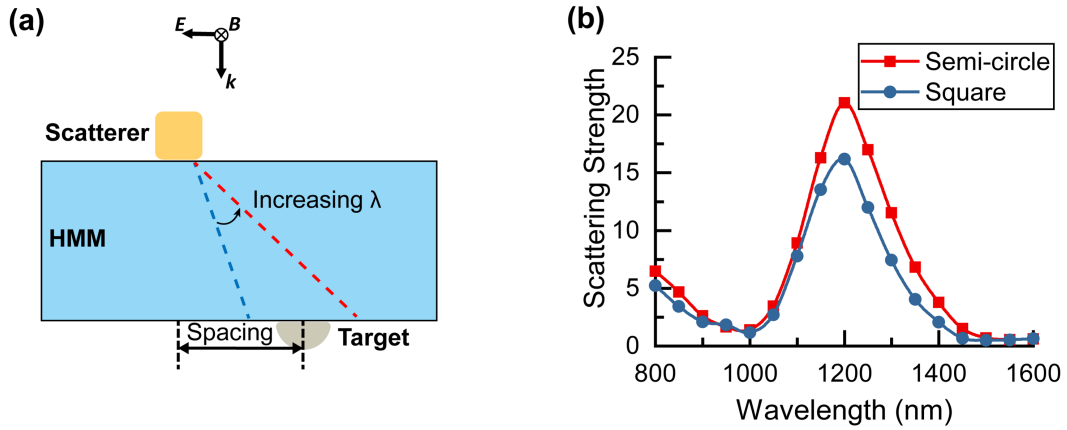

Figure A.4. (a) The device configuration after changing the shape of the bottom target from square to semi-circle. (b) Comparison of scattering strength versus wavelength for semi-circle target (red) and square target (blue). Both targets are at spacing of 400 nm.

A.5 Target shape dependence of the scattering strength

Here we consider the influence of the target shape on the scattering strength. We change the bottom target shape from a rounded square to a rounded semi-circle having the same area, as illustrated in Fig. A.4(a). Fig. A.4(b) shows the scattering strength versus wavelength for the rounded semi-circle (red line) and for the rounded square (blue line). It can be seen that two curves are quite similar. They have the same peak position, indicating the targets are at the same spacing of 400 nm. The scattering strength for the semi-circle target is slightly higher, because it is closer to the HMM surface on average and hence stronger scattering from the localized beam.

APPENDIX B

Focal Stack Based 3D Tracking

B.1 Single-point object focal stack from CMOS camera

We recorded the 1,331 single-point object focal stacks using the transparent graphene transistor array and separately using a CMOS sensor (Thorlab DCC1645C); see the right part of the Fig. 5.1 in the main text. By moving the CMOS sensor along z to focus at either the transparent detector array closer to or farther away from the lens, we captured focal stacks from CMOS camera. This data allows us to test how the image resolution and image quality of the graphene sensors affect the 3D ranging performance of a machine-learning algorithm.

We applied the following procedure to each high-resolution (1280×1024) color image captured by the CMOS camera: we convert the captured color image to gray image and optionally smooth it by spatial averaging and generate low resolution single-point object focal stacks of spatial size 4×4, 9×9 or 32×32. We used the processed images in either single-point tracking (to investigate the effects of imaging resolution to the tracking performance) or synthesizing multi-point object focal stacks.

B.2 Synthesizing multi-point object focal stack

We synthesized the multi-point object focal stack using focal stacks from the scanned single point object (either from transparent graphene transistor array or from CMOS camera). The synthesis is based on the assumption that the detector's response is linear, i.e., suppose I_i is the sensor image of the single point object at location (x_i, y_i, z_i). Then the sensor image I_{multi} consisting of multiple points can

be synthesized as $I_{\text{multi}} = \sum_{i=1}^{N} I_i$, where N is the number of point objects. We constructed the M-point object focal stack dataset, where the dataset consists of multiple subsets, and each subset consists of K possible shapes (relative position between points), by synthesizing each shape independently and then combining them. We translated an object to all possible (i.e., no point of M-point object is off the 3D grid) locations in the 3D $11 \times 11 \times 11$ scanning grid; at each location, we synthesize the corresponding focal stack according to the summation above. The number of synthesized datasets with $(M = 2, K = 2), (M = 2, K = 3), (M = 3, K = 2)$, (M=3, K=3) were 1600,2320,1232, and 1880, respectively.

We constructed the rotating 2-point object focal stack dataset by selecting focal stacks from the M-point focal stack with K possible shapes dataset, with $M = 2, K = 4$. Four shapes of a 2-point object (i.e., $M = 2$, K=4) are chosen to have same inter-point distance but rotated by different angles (26.5°, 63.5°, 116.5°, and 153.5°) about z axis (e.g., $(1, 0, 0)$ means 0° rotation about zaxis and $(0, 1, 0)$ means 90° rotation about zaxis). To form the helical trajectory in the $M = 2, K = 4$ setup, we selected an angle from the set $\{26.5°, 63.5°, 116.5°, \text{ and } 153.5°\}$ at each z position in the following sequence: $63.5°,\ 26.5°, 153.5°, 116.5°, 63.5°, 26.5°, 153.5°, 116.5°, 63.5°, 26.5°, 153.5°$, for $z = -10$ mm, -8 mm, $\ldots, 10$ mm. See graphical illustration in Fig. 5.2(e) of the main text.

B.3 Extended object focal stack

We captured extended object focal stacks using the CMOS sensor. The experimental setup is shown in Supplementary Fig. B.1. We used a ladybug as the extended object and moved it in a 3D spatial grid of size 8.5 mm $\times$ 8.5 mm $\times$ 45 mm. The grid spacing is 0.85 mm along both x and y, 3 mm along z. At each grid point, the object has 8 possible orientations in the x-z plane, with 45° angular separation between neighboring orientations. This leads to in total 1,5488 focal stacks, where each focal stack consists of two images captured by the CMOS sensor positioned at different z positions. Similar to section B.1, all images are converted to gray images before feeding to the neural networks.

B.4 Neural network architectures and training

We implemented all neural networks in Pytorch (ver. 1.0). The network architectures and training details are described below.

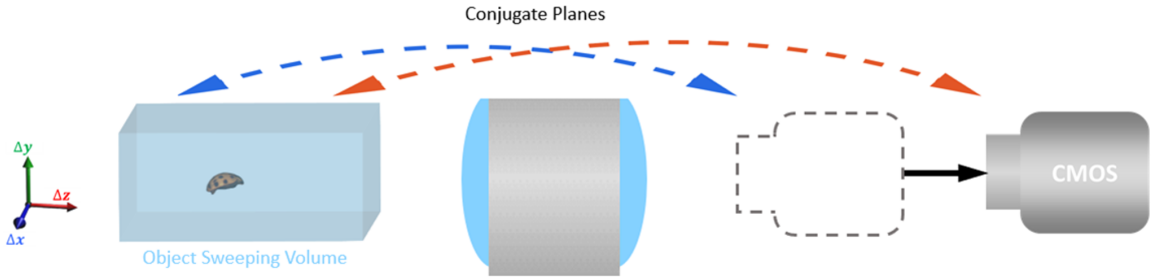

Figure B.1. Experimental set-up for capturing the extended object (ladybug) focal stack, using CMOS sensor.

For single-point object tracking, separate neural networks were trained for estimating the three spatial coordinates x, y and z, respectively. Supplementary Fig. B.1(a) shows the network architecture used for estimating coordinates x and y, and Supplementary Fig. B.1(b) shows the network architecture used for estimating z. For multi-point object tracking, a single neural network (Fig. B.1(c)) is trained to estimate all points' coordinates. In point object tracking cases, the focal stack data is flattened into a one-dimensional vector and subsequently passed through multilayer perceptron (MLP) using Rectified Linear Unit (ReLU) as the activation function.

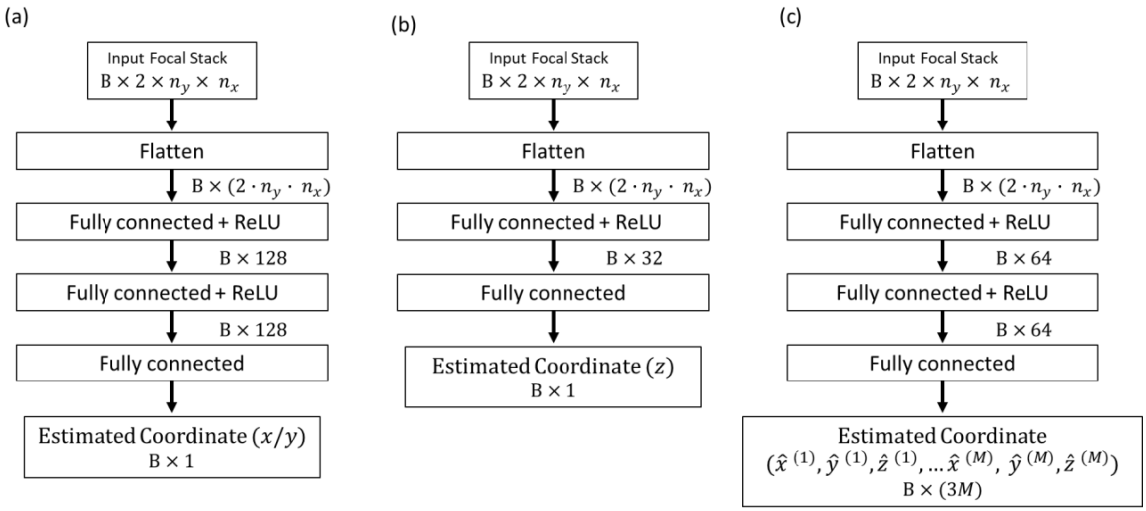

Figure B.2. Neural network architectures for 3D ranging. B is the general batch size of the data (e.g., in training, B is the training batch size; in testing with a single sample, B= 1). (a) Network for estimating single point object's x or y coordinate. (b) Network for estimating single point object's z coordinate. (c) Network for estimating M-point object's (x_i, y_i, z_i) coordinates tuple.

For single-point object tracking, the network outputs a single coordinate value for each focal stack, and the networks are trained by minimizing the following mean-

square error (MSE) loss:

$$\frac{1}{N}\sum_{i=1}^{N}\left(\hat{s_i}-s_i\right)^2,$$

where N is the number of training samples, s_i is the true spatial coordinate (x_i, y_i, or z_i) and $\hat{s_i}$ is the estimated spatial coordinate from a neural network. We trained networks using the Adam optimizer with the learning rate of 10^{-2}, the training batch size of 50, and 2000 epochs.

For training multi-point object tracking neural networks, we defined the following MSE loss that considers the ordering ambiguity of the network outputs in training:

$$\frac{1}{N}\sum_{i=1}^{N}\min_{(p_1,\ldots,p_M)\in P}\sum_{j=1}^{M}\left(\hat{x}_i^{(j)}-x_i^{(p_j)}\right)^2+\left(\hat{y}_i^{(j)}-y_i^{(p_j)}\right)^2+\left(\hat{z}_i^{(j)}-z_i^{(p_j)}\right)^2, \tag{B.1}$$

where M is the number of points of the object, P is the set containing all possible permutations of the tuple $(1, 2, \ldots M)$, $x_i^{(j)}$ and $\hat{x}_i^{(j)}$ are the true and estimated coordinate of the i^{th} data sample, j^{th} point. The network outputs a coordinates tuple for all the points of the object as $\left\{\left(\hat{x}^{(1)}, \hat{y}^{(1)}, \overset{(1)}{z}\right), \ldots, \left(\hat{x}^{(M)}, \hat{y}^{(M)}, \hat{z}^{(M)}\right)\right\}$. To consider the ordering ambiguity of the network outputs in training, e.g., for $\left(x^{(1)}, y^{(1)}, z^{(1)}\right)$, the network cannot determine which estimate gives lower MSE, between $\left(x^{(1)}, \hat{y}^{(1)}, \hat{z}^{(1)}\right)$ and $\left(\hat{x}^{(2)}, \hat{y}^{(2)}, \hat{z}^{(2)}\right)$, we found proper orders by minimizing MSE over the permutation set P in B.1. With the help of minimization over P, the loss will be low as long as a trained network predicts the overall shape of the object, regardless of the order of the network estimates. In the training, we scaled down the true z coordinate values by 33.3 so that it is in the same range as coordinates x and y. This avoids the loss B.1 from being dominated by z component of MSE loss, i.e., avoids training from being biased to z-coordinate estimation. We trained the network using Adam optimizer with the learning rate of 10^{-3}, the training batch size of 100, and 2000 epochs. For tracking the two-point rotating object, we also trained the network by B.1 and scaled the z coordinate values by 33.3. For training the network, we used Adam optimizer with the learning rate of 10^{-3}, the training batch size of 100, and 2000 epochs.

For extended object tracking and orientation estimation, we use two convolutional neural networks (CNNs) (Fig. B.3) similar to VGG-16 [94]. The CNN shown in Fig. B.3(a) is used for the tracking. For each focal image, we first extract high-level feature maps with multiple convolution-batch normalization (BN)-ReLU-pooling layers. Then we apply the following procedure to extracted feature maps from all focal images: 1) concatenation of all feature maps along channel dimension, 2) average

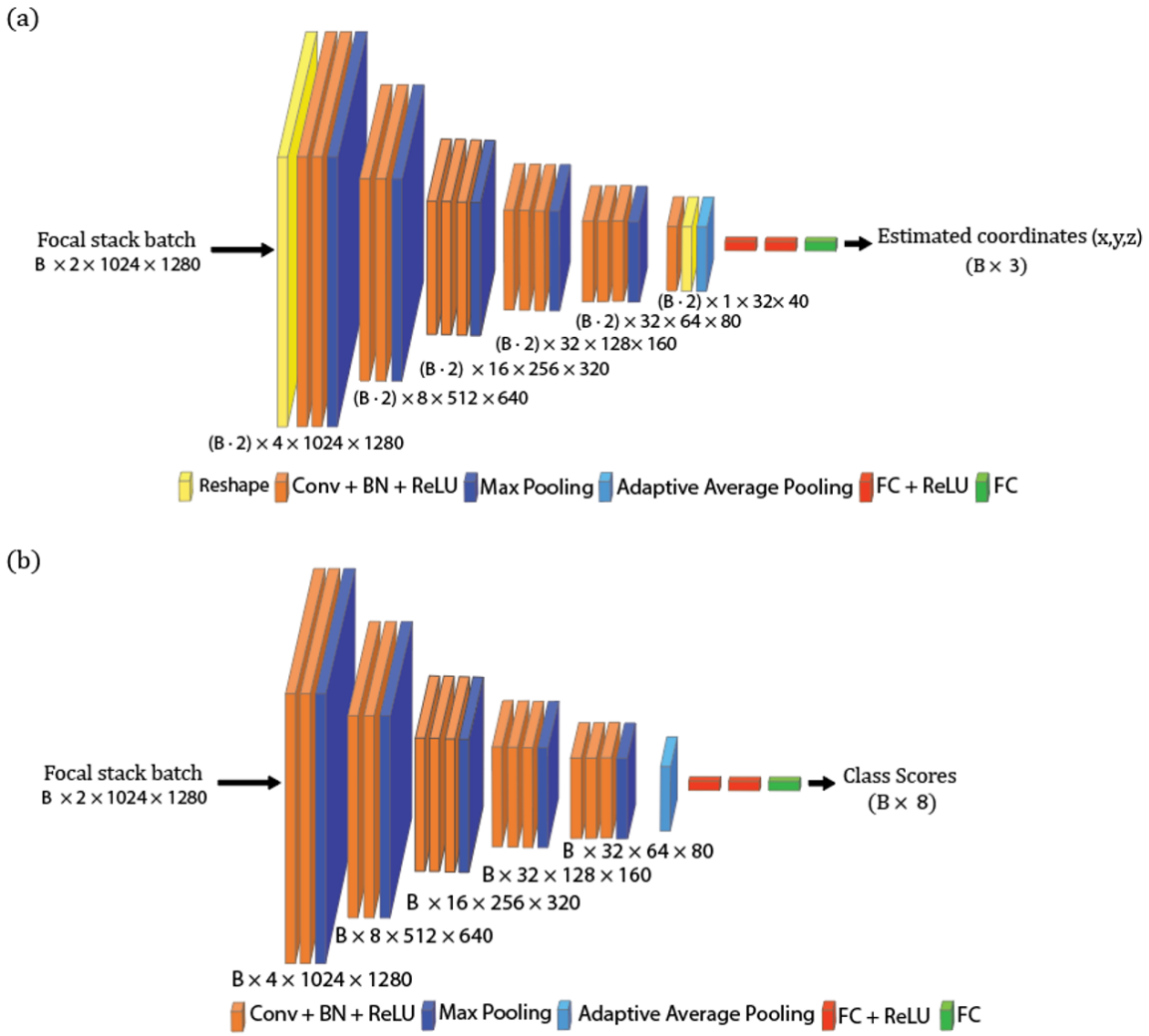

Figure B.3. Convolutional neural network architectures for extended object tracking and orientation estimation. B is the general batch size of the data (e.g., in training, B is the training batch size; in testing with a single sample, B= 1). (a) Network for estimating extended object's spatial coordinates (x, y, z). (b) Network for estimating extended object's orientation.

pooling, 3) flattening, and 4) feeding the output into fully connected layers (FC) that lead to final coordinates. The network is trained by minimizing the following MSE loss:

$$\frac{1}{3N}\sum_{i=1}^{N}\left(\hat{x}_i - x_i\right)^2 + \left(\hat{y}_i - y_i\right)^2 + \left(\hat{z}_i - z_i\right)^2,$$

where N is the number of training samples. In the training, we scaled the true z coordinate values to have the same range as x and y coordinates, for the same reason as in the multi-point object tracking. The CNN shown in Fig. B.3(b) is

used for object orientation estimation. We consider the problem as a multi-class classification problem: the CNN takes focal stack as input and output scores that are used to classify the object orientations with eight different orientations. The network is trained by minimizing the cross-entropy loss.

B.5 Ranging performance comparison

We studied the effect of the detector resolution and spatial smoothing on the single-point object 3D ranging performance. Table B.1 summarizes the results. The resolution of the CMOS focal stack is varied to see its effect on the ranging performance: it can be seen by comparing horizontally the root mean square error (RMSE) in the 2nd, 3rd and 4th columns or in the 5th and 6th columns that higher resolution focal stack gives lower loss. Besides, note that spatially averaged results have lower loss, compared to those without averaging. This is because the noise from interference fringes is suppressed after applying spatial averaging.

	4×4 Graphene	4×4 CMOS	9×9 CMOS	32×32 CMOS	4×4 (Avg. 20) CMOS	9×9 (Avg. 20) CMOS
x	0.012	0.031	0.020	0.021	0.014	0.009
y	0.014	0.028	0.017	0.012	0.012	0.010
z	1.196	1.304	1.192	0.480	0.616	0.458

Table B.1. Single-point object 3D ranging RMSE (unit: mm) table on testing set. Avg. 20 means spatial averaging with window size 20 is performed on the raw high-resolution focal stack.

	4×4 Graphene	4×4 CMOS	9×9 CMOS	32×32 CMOS	4×4 (Avg. 20) CMOS	9×9 (Avg. 20) CMOS
2p2s	0.017	0.036	0.025	0.013	0.020	0.013
2p3s	0.019	0.033	0.022	0.013	0.019	0.012
3p2s	0.019	0.042	0.027	0.025	0.021	0.016
3p3s	0.021	0.041	0.029	0.028	0.022	0.017

Table B.2. Multi-point object 3D ranging RMSE (unit: mm) table of x on testing set. Avg. 20 means spatial averaging with window size 20 is performed on the raw high-resolution focal stack. First column encodes different object configurations, e.g., 2p3s means 2-point object with 3 possible shapes.

Table B.2, B.3, B.4 summarize the study of the effect of the detector resolution and spatial smoothing on the multi-point object 3D ranging performance. Similar to

	4×4 Graphene	4×4 CMOS	9×9 CMOS	32×32 CMOS	4×4 (Avg. 20) CMOS	9×9 (Avg. 20) CMOS
2p2s	0.022	0.045	0.033	0.019	0.026	0.017
2p3s	0.025	0.039	0.028	0.018	0.025	0.015
3p2s	0.010	0.019	0.013	0.016	0.011	0.007
3p3s	0.019	0.035	0.026	0.027	0.021	0.016

Table B.3. Multi-point object 3D ranging RMSE (unit: mm) table of y on testing set. Avg. 20 means spatial averaging with window size 20 is performed on the raw high-resolution focal stack. First column encodes different object configurations, e.g., 2p3s means 2-point object with 3 possible shapes

	4×4 Graphene	4×4 CMOS	9×9 CMOS	32×32 CMOS	4×4 (Avg. 20) CMOS	9×9 (Avg. 20) CMOS
2p2s	0.685	1.073	0.759	0.349	0.557	0.371
2p3s	1.164	1.573	1.142	0.788	0.983	0.641
3p2s	0.793	1.328	0.876	0.715	0.750	0.470
3p3s	0.894	1.444	1.004	0.895	0.850	0.594

Table B.4. Multi-point object 3D ranging RMSE (unit: mm) table of z on a testing set. Avg. 20 means spatial averaging with window size 20 is performed on the raw high-resolution focal stack. First column encodes different object configurations, e.g., 2p3s means 2-point object with 3 possible shapes.

the single-point object case, more pixels are useful in reducing the ranging error, as can be seen by comparing horizontally the RMSE in 2nd, 3rd and 4th columns or in the 5th and 6th columns. The spatial averaging is again helpful, as in the single object case, in reducing the estimation error. The numerical results above are also illustrated graphically in Fig. B.4, B.6, B.7, B.8 below.

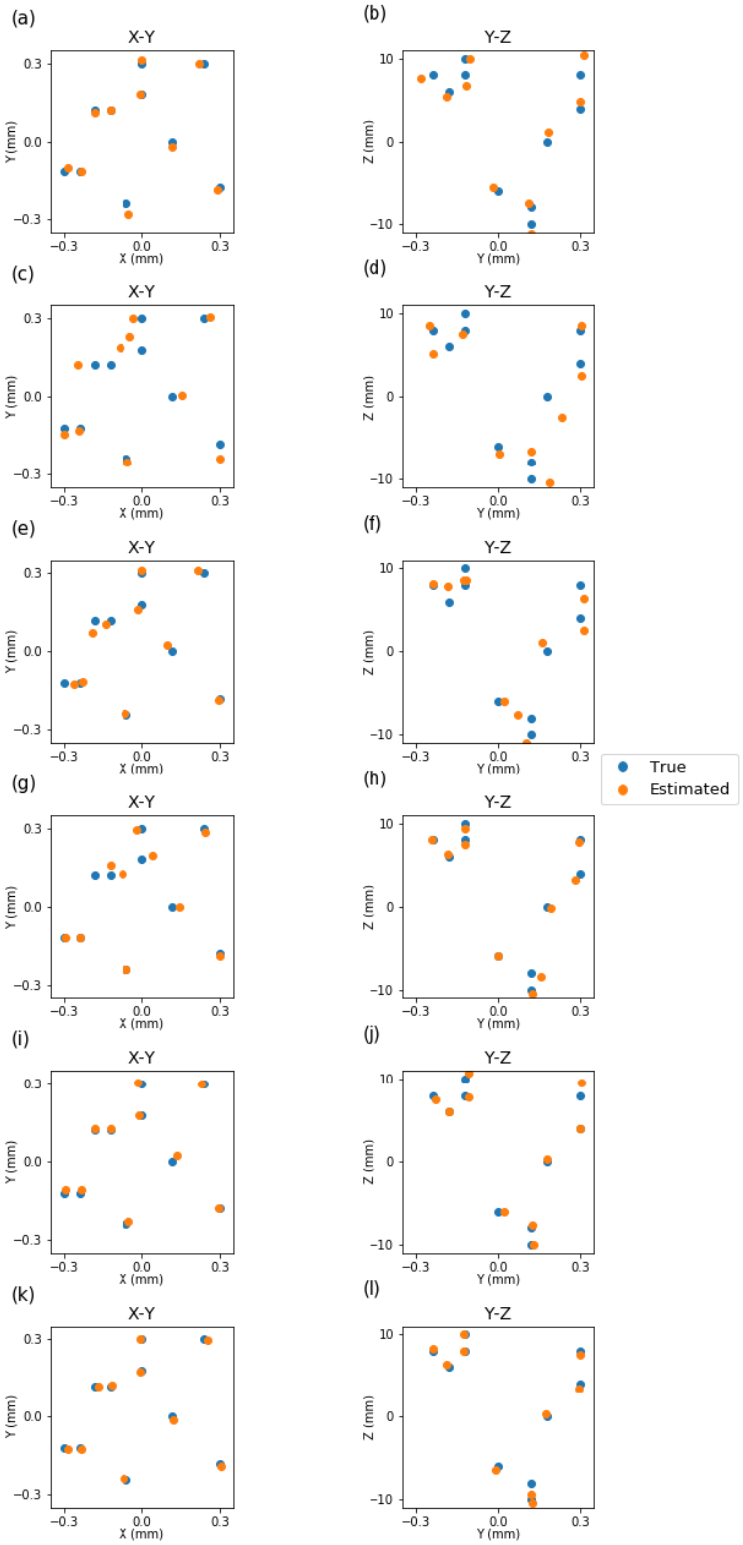

Figure B.4. Single-point object tracking performance (only 10 test samples are shown). Focal stack data from: (a-b) 4×4 transparent graphene detector. (c-d) 4×4 CMOS sensor. (e-f) 9×9 CMOS sensor. (g-h) 32×32 CMOS sensor. (i-j) 4×4 Avg. 20 CMOS sensor. (k-l) 9×9 Avg. 20 CMOS sensor.

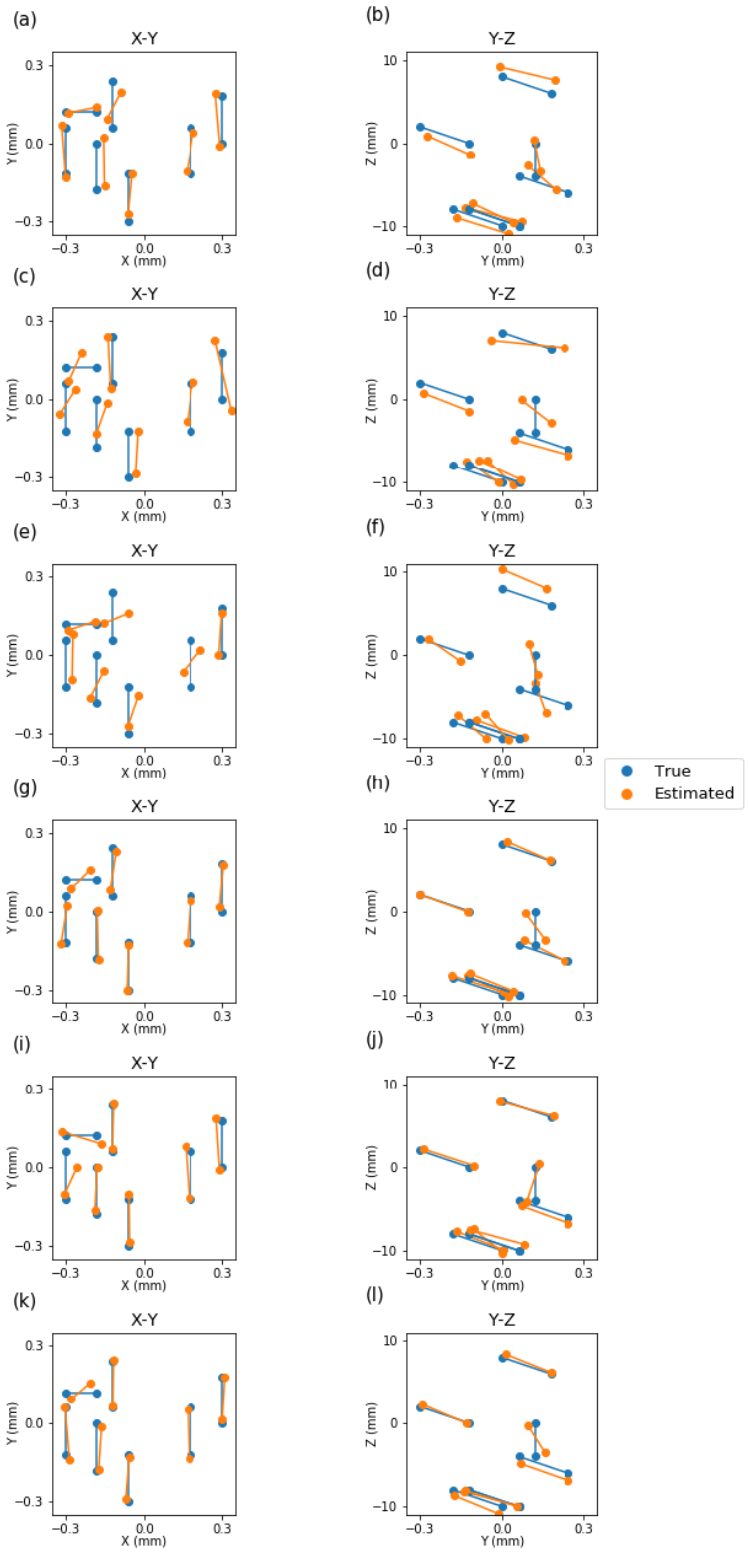

Figure B.5. 2-point object with 2 possible shapes tracking performance (only 7 test samples are shown). Focal stack data from: (a-b) 4×4 transparent graphene detector. (c-d) 4×4 CMOS sensor. (e-f) 9×9 CMOS sensor. (g-h) 32×32 CMOS sensor. (i-j) 4×4 Avg. 20 CMOS sensor. (k-l) 9×9 Avg. 20 CMOS sensor.

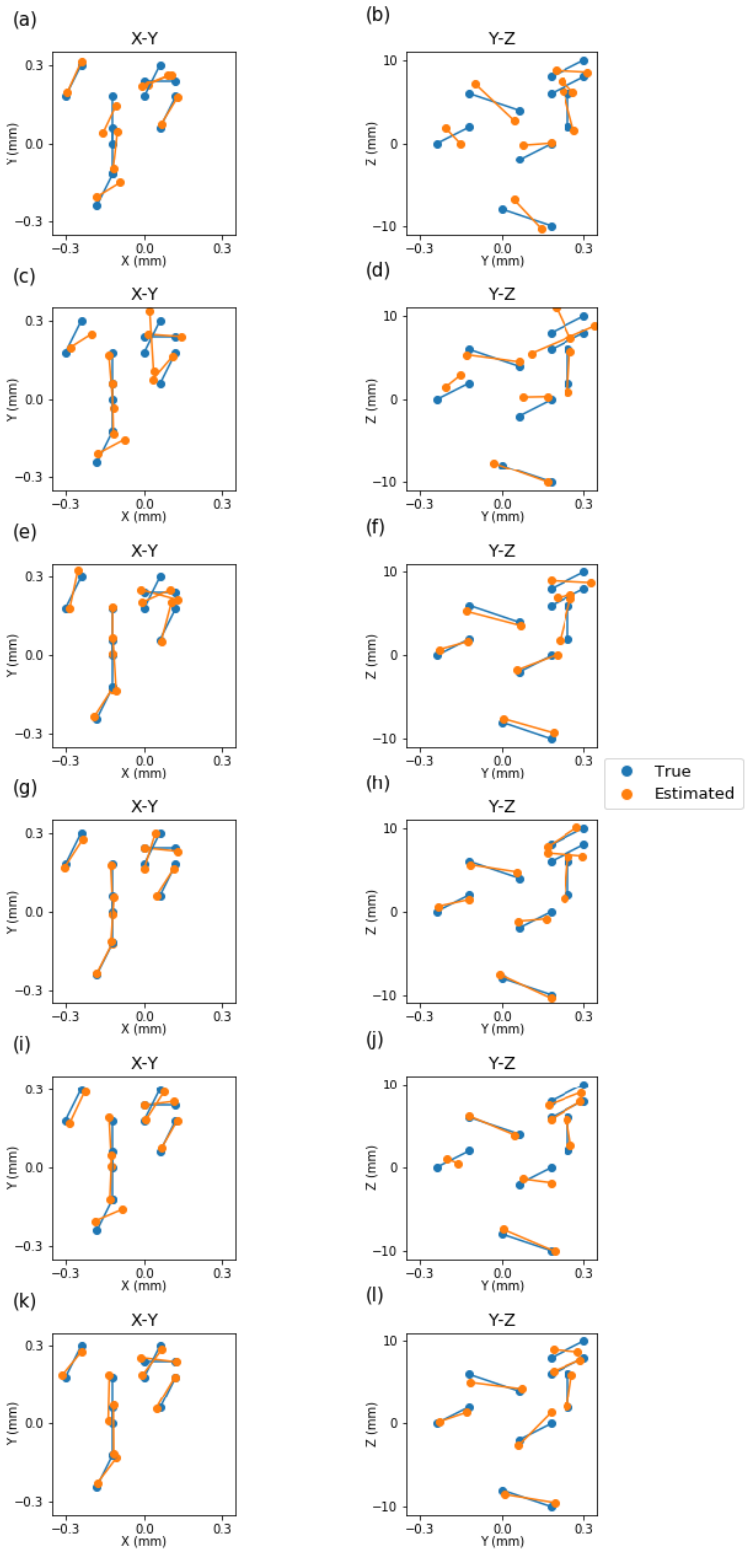

Figure B.6. 2-point object with 3 possible shapes tracking performance (only 7 test samples are shown). Focal stack data from: (a-b) 4×4 transparent graphene detector. (c-d) 4×4 CMOS sensor. (e-f) 9×9 CMOS sensor. (g-h) 32×32 CMOS sensor. (i-j) 4×4 Avg. 20 CMOS sensor. (k-l) 9×9 Avg. 20 CMOS sensor.

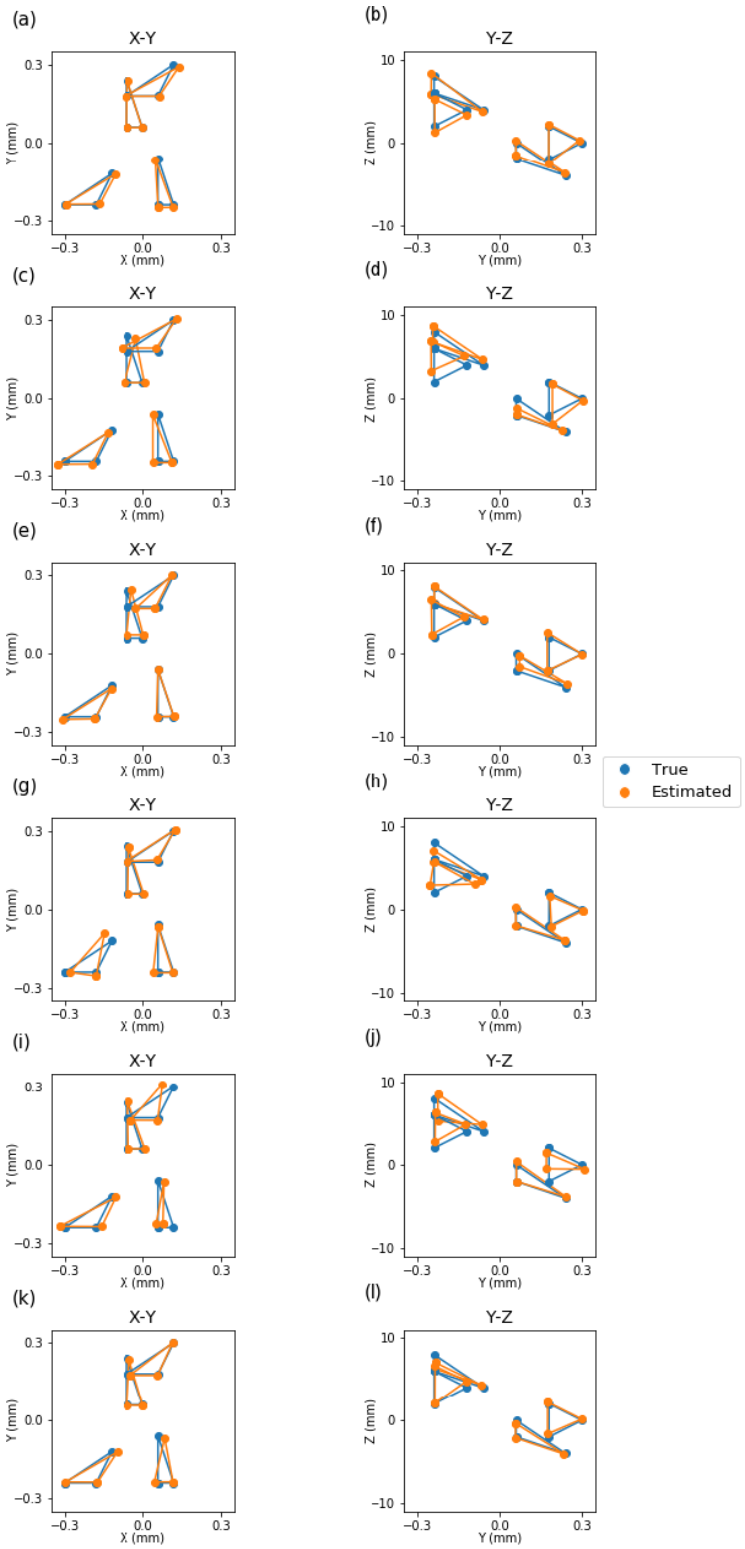

Figure B.7. 3-point object with 2 possible shapes tracking performance (only 4 test samples are shown). Focal stack data from: (a-b) 4×4 transparent graphene detector. (c-d) 4×4 CMOS sensor. (e-f) 9×9 CMOS sensor. (g-h) 32×32 CMOS sensor. (i-j) 4×4 Avg. 20 CMOS sensor. (k-l) 9×9 Avg. 20 CMOS sensor.

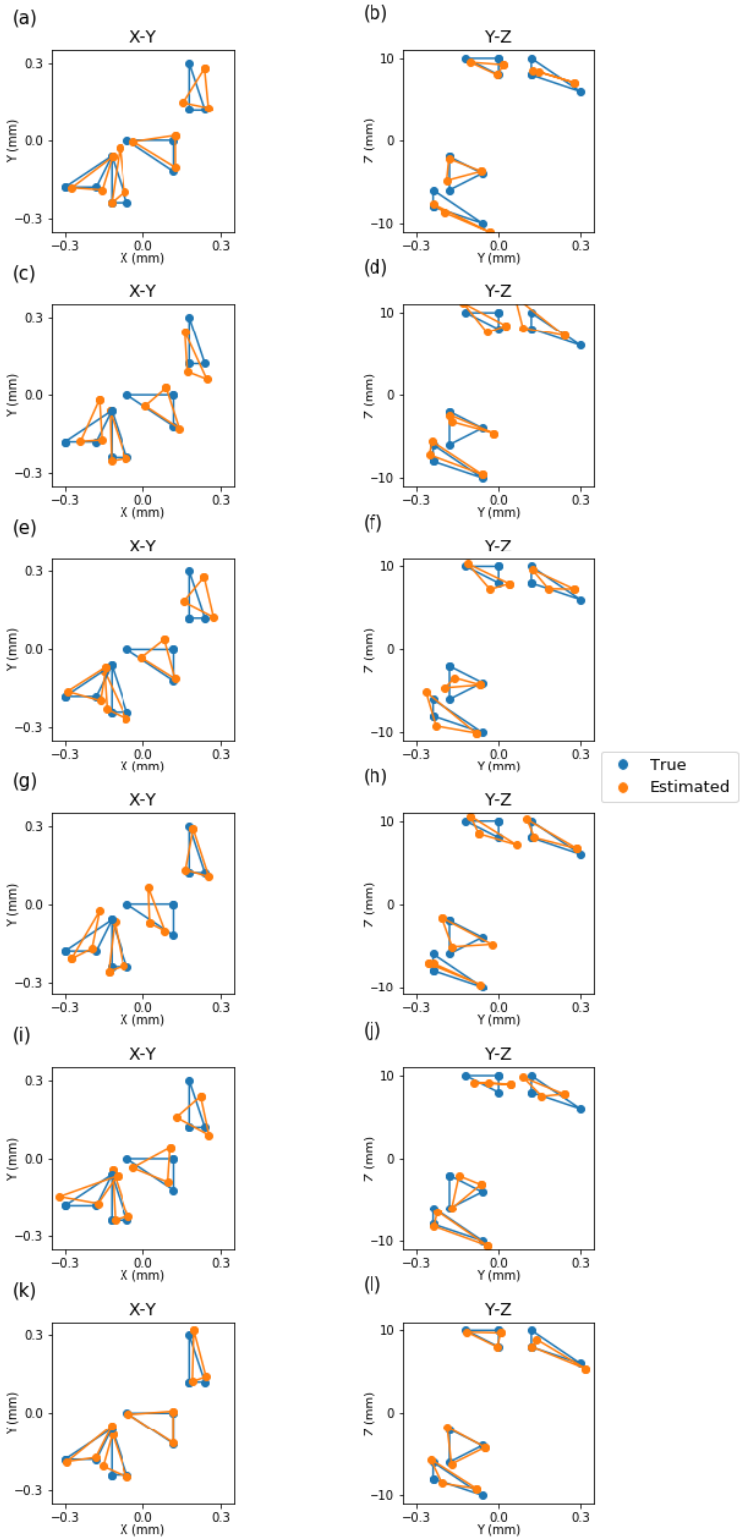

Figure B.8. 3-point object with 3 possible shapes tracking performance (only 4 test samples are shown). Focal stack data from: (a-b) 4×4 transparent graphene detector. (c-d) 4×4 CMOS sensor. (e-f) 9×9 CMOS sensor. (g-h) 32×32 CMOS sensor. (i-j) 4×4 Avg. 20 CMOS sensor. (k-l) 9×9 Avg. 20 CMOS sensor.

APPENDIX C

Secure Imaging using Focal Stack Camera

C.1 Effect of JPEG augmentation for training

Here we include additional results of models trained without JPEG augmentation (section 7.3.2). Comparing Fig. 7.3 and Fig. C.1 shows that include JPEG augmentation during training leads to a model more robust against post-processing perturbations and better performance. The benefit is more significant for Gaussian noise perturbation (1st row of Fig. C.1) and JPEG compression (3rd row of Fig. C.1). The F_1 score of the model trained without JPEG augmentation will degrade quickly when the images are JPEG compressed or noise is added. Regardless, the advantage of using focal stack over single image based method is still significant for this training scheme as well.

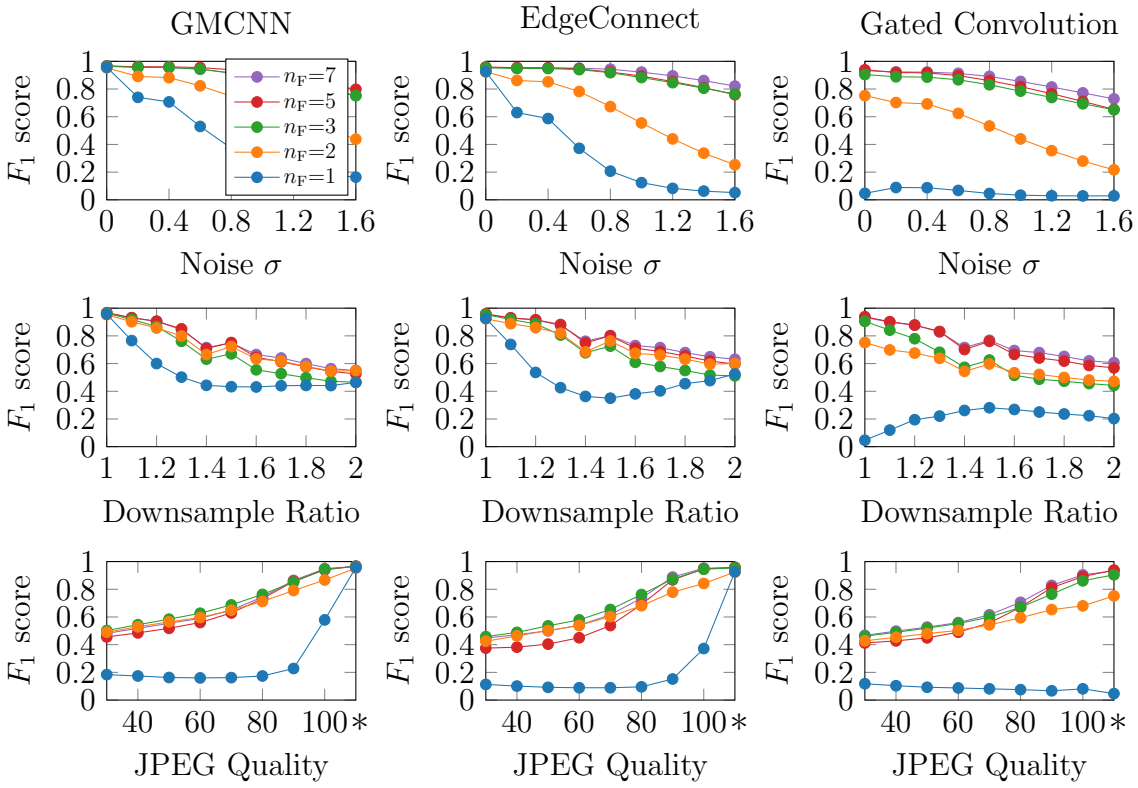

Figure C.1. Localization F_1 scores for focal stack data with networks trained on GMCNN dataset without JPEG augmentation and tested on GMCNN (1st column), EdgeConnect (2nd column) and Gated Convolution (3rd column) datasets. Symbol '*' on x-axis indicates the result without JPEG compression.

APPENDIX D

Focal Stack Camera Depth Estimation Performance Comparison and Design Exploration

This supplement describes additional details in the Nikon focal stack dataset collection process.

D.1 Focal stack collection

Figure D.1 shows the setup of the RGB camera and the depth camera. When changing the aperture size, we also changed the exposure time and the ISO setting such that the exposure level is approximately the same. Specifically, the shutter settings are 1/60, 1/25, 1/6, 1/3 for aperture f/3.2, f/5, f/10, f/22, respectively, and the ISO is set to 1000 for f/3.2, f/5, f/10 and set to 2500 for f/22.

Since there could be mechanical hysteresis in the focus motor, when capturing each focal stack sequence, we adjusted the focus such that the image is sharply focused at the reference target on the bottom-left corner of the field of view. This step ensures that the first image in each focal stack starts at the same focus position.

D.2 Focal stack alignment

The magnification change when changing the focus of the camera can be modeled by a homography [4]. To correct this magnification change, we place a chessboard in front of the camera and captured a focal stack, using aperture f/16. We extract all the corners locations using corner detector, and then estimate the homography between the first image in the focal stack and any other images in the focal stack. We

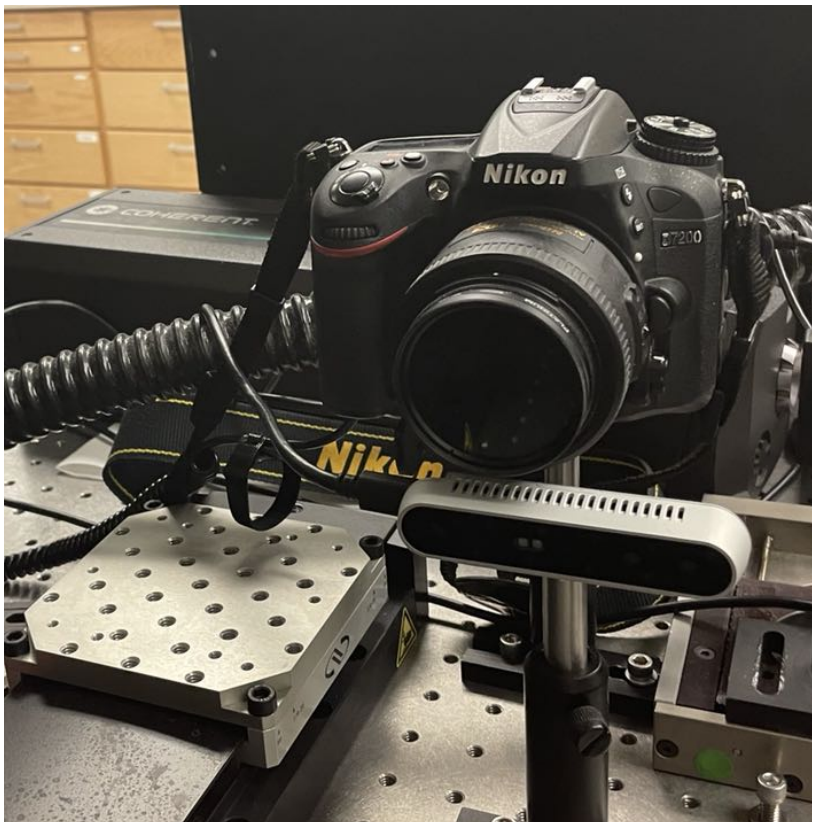

Figure D.1. Setup of the RGB camera and the Intel RealSense D415 Depth Camera.

then use the estimated homographies to align other collected focal stack data. Figure D.2 shows the first and the last image in the focal stack used for the homography estimation.

D.3 Depth registration

The depth map captured by the depth camera has a different resolution and view point than the RGB camera. Thus, a depth registration process was needed to register the raw depth map onto the RGB image. Using a set of chessboard images with different poses, we calibrated the RGB camera and the depth camera separately to extract intrinsic camera parameters and then use stereo calibration to estimate the relative rotation R and translation T between them. The estimated parameters were then used to project the depth pixel onto the RGB image plane using the 'registerDepth' function in OpenCV library. Figure D.3 shows example images for camera calibration.

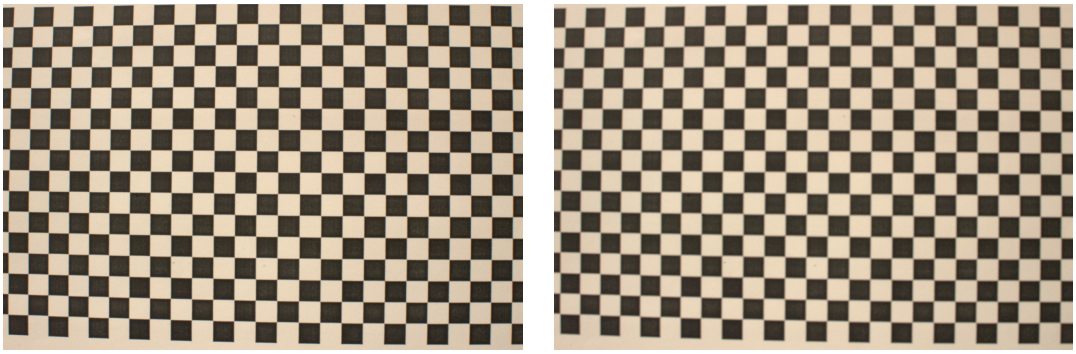

Figure D.2. The first and the last image of the focal stack used for focal stack alignment.

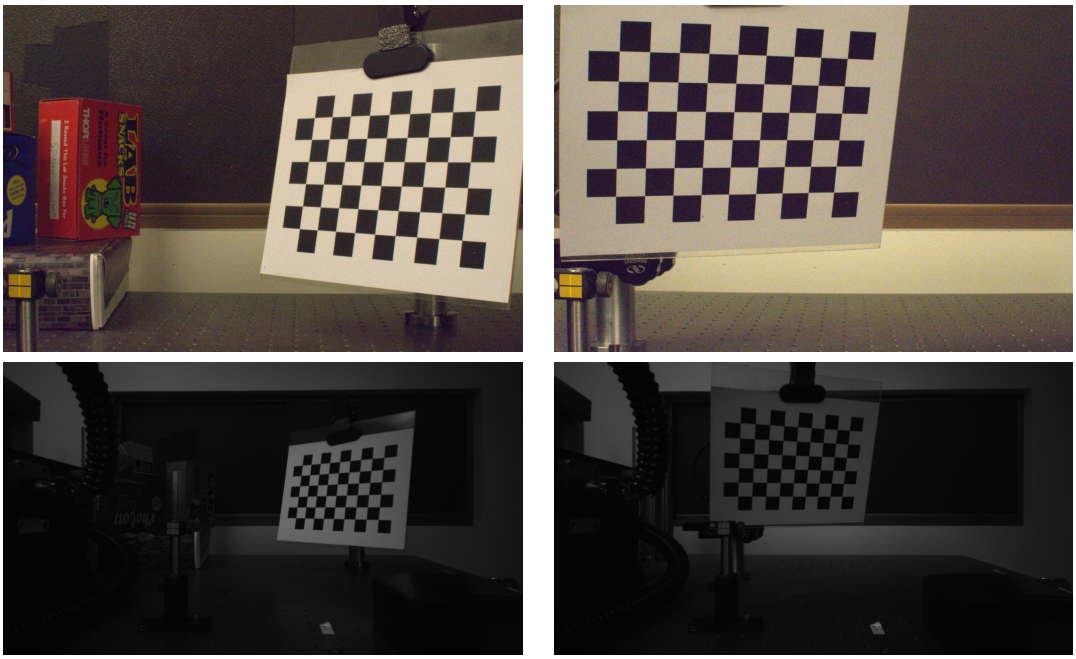

Figure D.3. Example images used for RGB camera and depth camera calibration.

BIBLIOGRAPHY

[1] Deepfakes faceswap. `https://github.com/deepfakes/faceswap`.

[2] Edward H. Adelson and James R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, pages 3–20. MIT Press, 1991.

[3] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE Transactions on Medical Imaging*, 37(6):1322–1332, 2018.

[4] Alex M Andrew. Multiple view geometry in computer vision. *Kybernetes*, 2001.

[5] Vegard Antun, Francesco Renna, Clarice Poon, Ben Adcock, and Anders C Hansen. On instabilities of deep learning in image reconstruction and the potential costs of ai. *Proceedings of the National Academy of Sciences*, 117(48):30088–30095, 2020.

[6] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

[7] Claire Berger, Zhimin Song, Tianbo Li, Xuebin Li, Asmerom Y Ogbazghi, Rui Feng, Zhenting Dai, Alexei N Marchenkov, Edward H Conrad, Phillip N First, et al. Ultrathin epitaxial graphite: 2d electron gas properties and a route toward graphene-based nanoelectronics. *The Journal of Physical Chemistry B*, 108(52):19912–19916, 2004.

[8] Claire Berger, Zhimin Song, Xuebin Li, Xiaosong Wu, Nate Brown, Cécile Naud, Didier Mayou, Tianbo Li, Joanna Hass, Alexei N Marchenkov, et al. Electronic confinement and coherence in patterned epitaxial graphene. *Science*, 312(5777):1191–1196, 2006.

[9] Eric Betzig, George H Patterson, Rachid Sougrat, O Wolf Lindwasser, Scott Olenych, Juan S Bonifacino, Michael W Davidson, Jennifer Lippincott-Schwartz, and Harald F Hess. Imaging intracellular fluorescent proteins at nanometer resolution. *Science*, 313(5793):1642–1645, 2006.

[10] Eric Betzig and Jay K Trautman. Near-field optics: microscopy, spectroscopy, and surface modification beyond the diffraction limit. *Science*, 257(5067):189–195, 1992.

[11] Cameron J Blocker, Il Yong Chun, and Jeffrey A. Fessler. Low-rank plus sparse tensor models for light-field reconstruction from focal stack data. In *Proc. IEEE Image, Video, and Multidim. Signal Process. Workshop (IVMSP)*, pages 1–5, June 2018.

[12] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[13] Gregery T Buzzard, Stanley H Chan, Suhas Sreehari, and Charles A Bouman. Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium. *SIAM Journal on Imaging Sciences*, 11(3):2001–2020, 2018.

[14] Can Chen, Haiting Lin, Zhan Yu, Sing Bing Kang, and Jingyi Yu. Light field stereo matching using bilateral statistics of surface cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1518–1525, 2014.

[15] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[16] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272, 2016.

[17] Il Yong Chun and Jeffrey A Fessler. Convolutional analysis operator learning: Acceleration and convergence. *IEEE Transactions on Image Processing*, 29:2108–2122, 2019.

[18] Il Yong Chun, Zhengyu Huang, Hongki Lim, and Jeff Fessler. Momentum-net: Fast and convergent iterative neural network for inverse problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[19] Il Yong Chun, Hongki Lim, Zhengyu Huang, and Jeffrey A Fessler. Fast and convergent iterative image recovery using trained convolutional neural networks. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 155–159. IEEE, 2018.

[20] Il Yong Chun, Xuehang Zheng, Yong Long, and Jeffrey A. Fessler. BCD-Net for low- dose CT reconstruction: Acceleration, convergence, and generalization. In *Proc. Med. Image Computing and Computer Assist. Interven. (MICCAI)* (to appear), Shenzhen, China, Oct. 2019.

[21] Davide Cozzolino, Diego Gragnaniello, and Luisa Verdoliva. Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5302–5306. IEEE, 2014.

[22] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. ForensicTransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv preprint arXiv:1812.02510*, 2018.

[23] Sajjad Dadkhah, Azizah Abd Manaf, Yoshiaki Hori, Aboul Ella Hassanien, and Somayeh Sadeghi. An effective svd-based image tampering detection and self-recovery using active watermarking. *Signal Processing: Image Communication*, 29(10):1197–1210, 2014.

[24] Walt A De Heer, Claire Berger, Ming Ruan, Mike Sprinkle, Xuebin Li, Yike Hu, Baiqian Zhang, John Hankinson, and Edward Conrad. Large area and structured epitaxial graphene produced by confinement controlled sublimation of silicon carbide. *Proceedings of the National Academy of Sciences*, 108(41):16900–16905, 2011.

[25] Hany Farid. Image forgery detection. *IEEE Signal Processing Magazine*, 26(2):16–25, 2009.

[26] Jeffrey A Fessler. Penalized weighted least-squares image reconstruction for positron emission tomography. *IEEE Transactions on Medical Imaging*, 13(2):290–300, 1994.

[27] S Foteinopoulou, M Kafesaki, EN Economou, and CM Soukoulis. Two-dimensional polaritonic photonic crystals as terahertz uniaxial metamaterials. *Physical Review B*, 84(3):035128, 2011.

[28] Marco Furchi, Alexander Urich, Andreas Pospischil, Govinda Lilley, Karl Unterrainer, Hermann Detz, Pavel Klang, Aaron Maxwell Andrews, Werner Schrenk, Gottfried Strasser, et al. Microcavity-integrated graphene photodetector. *Nano letters*, 12(6):2773–2777, 2012.

[29] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–756. Springer, 2016.

[30] Davis Gilton, Greg Ongie, and Rebecca Willett. Neumann networks for linear inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 6:328–343, 2019.

[31] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 270–279, 2017.

[32] Shir Gur and Lior Wolf. Single image depth estimation trained via depth from defocus cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[33] Caner Hazirbas, Sebastian Georg Soyer, Maximilian Christian Staab, Laura Leal-Taixé, and Daniel Cremers. Deep depth from focus. In *Asian Conference on Computer Vision*, pages 525–541. Springer, 2018.

[34] Caner Hazirbas, Sebastian Georg Soyer, Maximilian Christian Staab, Laura Leal-Taixé, and Daniel Cremers. Deep depth from focus. In *Asian Conference on Computer Vision*, pages 525–541. Springer, 2018.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV))*, pages 1026–1034, 2015.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[37] Stefan W Hell and Jan Wichmann. Breaking the diffraction resolution limit by stimulated emission: stimulated-emission-depletion fluorescence microscopy. *Optics Letters*, 19(11):780–782, 1994.

[38] Katrin Honauer, Ole Johannsen, Daniel Kondermann, and Bastian Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. In *Asian Conference on Computer Vision*, pages 19–34. Springer, 2016.

[39] Zhengyu Huang, Jeffrey A Fessler, Theodore B Norris, and Il Yong Chun. Light-field reconstruction and depth estimation from focal stack images using convolutional neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8648–8652. IEEE, 2020.

[40] Zhengyu Huang, Theodore B Norris, and Evgenii Narimanov. Nanoscale fingerprinting with hyperbolic metamaterials. *APL Photonics*, 4(2):026103, 2019.

[41] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A Efros. Fighting fake news: Image splice detection via learned self-consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 101–117, 2018.

[42] Aaron Isaksen, Leonard McMillan, and Steven J Gortler. Dynamically reparameterized light fields. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 297–306, 2000.

[43] Satoshi Ishii, Alexander V Kildishev, Evgenii Narimanov, Vladimir M Shalaev, and Vladimir P Drachev. Sub-wavelength interference pattern from volume plasmon polaritons in a hyperbolic medium. *Laser & Photonics Reviews*, 7(2):265–271, 2013.

[44] Zubin Jacob, Leonid V Alekseyev, and Evgenii Narimanov. Optical hyperlens: far-field imaging beyond the diffraction limit. *Optics Express*, 14(18):8247–8256, 2006.

[45] Micah K Johnson and Hany Farid. Exposing digital forgeries through chromatic aberration. In *Proceedings of the 8th Workshop on Multimedia and Security*, pages 48–55, 2006.

[46] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Trans. on Graphics (TOG)*, 35(6):193, Nov. 2016.

[47] Can Kanbak, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Geometric robustness of deep networks: analysis and improvement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4441–4449, 2018.

[48] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, pages 1–15, May 2015.

[49] Pavel Korshunov and Sébastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018.

[50] Stein Kuiper and BHW Hendriks. Variable-focus liquid lens for miniature cameras. *Applied physics letters*, 85(7):1128–1130, 2004.

[51] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 31–42, 1996.

[52] Haodong Li and Jiwu Huang. Localization of deep inpainting using high-pass fully convolutional network. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 8301–8310, 2019.

[53] Xuesong Li, Weiwei Cai, Jinho An, Seyoung Kim, Junghyo Nah, Dongxing Yang, Richard Piner, Aruna Velamakanni, Inhwa Jung, Emanuel Tutuc, et al. Large-area synthesis of high-quality and uniform graphene films on copper foils. *science*, 324(5932):1312–1314, 2009.

[54] Miao-Bin Lien, Che-Hung Liu, Il Yong Chun, Saiprasad Ravishankar, Hung Nien, Minmin Zhou, Jeffrey A Fessler, Zhaohui Zhong, and Theodore B Norris. Ranging and light field imaging with transparent photodetectors. *Nature Photonics*, 14(3):143–148, 2020.

[55] Chang-Hua Liu, You-Chia Chang, Theodore B Norris, and Zhaohui Zhong. Graphene photodetectors with ultra-broadband and high responsivity at room temperature. *Nature nanotechnology*, 9(4):273–278, 2014.

[56] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018.

[57] Zhaowei Liu, Hyesog Lee, Yi Xiong, Cheng Sun, and Xiang Zhang. Far-field optical hyperlens magnifying sub-diffraction-limited objects. *Science*, 315(5819):1686–1686, 2007.

[58] Jan Lukáš, Jessica Fridrich, and Miroslav Goljan. Detecting digital image forgeries using sensor pattern noise. In *Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, page 60720Y. International Society for Optics and Photonics, 2006.

[59] R Madhavi and K Ashok Babu. An all approach for multi-focus image fusion using neural network. *International Journal of Computer Science and Telecommunications*, 2(8):23–17, 2011.

[60] Nupam P Mahajan, Katrina Linder, Gail Berry, Gerald W Gordon, Roger Heim, and Brian Herman. Bcl-2 and bax interactions in mitochondria probed with green fluorescent protein and fluorescence resonance energy transfer. *Nature Biotechnology*, 16(6):547–552, 1998.

[61] Wojciech Matusik and Hanspeter Pfister. 3D TV: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. *ACM Transactions on Graphics (TOG)*, 23(3):814–824, 2004.

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

[63] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[64] Philip M Morse and Herman Feshbach. Methods of theoretical physics. *American Journal of Physics*, 22(6):410–413, 1954.

[65] Thomas Mueller, Fengnian Xia, and Phaedon Avouris. Graphene photodetectors for high-speed optical communications. *Nature photonics*, 4(5):297–301, 2010.

[66] Rahul Raveendran Nair, Peter Blake, Alexander N Grigorenko, Konstantin S Novoselov, Tim J Booth, Tobias Stauber, Nuno MR Peres, and Andre K Geim. Fine structure constant defines visual transparency of graphene. *Science*, 320(5881):1308–1308, 2008.

[67] Evgenii Narimanov. Hyperstructured illumination. *ACS Photonics*, 3(6):1090–1094, 2016.

[68] Evgenii E Narimanov. Photonic hypercrystals. *Physical Review X*, 4(4):041014, 2014.

[69] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

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

[71] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. EdgeConnect: Structure guided image inpainting using edge prediction. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[72] AH Castro Neto, Francisco Guinea, Nuno MR Peres, Kostya S Novoselov, and Andre K Geim. The electronic properties of graphene. *Reviews of modern physics*, 81(1):109, 2009.

[73] Ren Ng et al. *Digital light field photography*, volume 7. Stanford University Stanford, 2006.

[74] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, Pat Hanrahan, et al. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11):1–11, June 2005.

[75] Kostya S Novoselov, Andre K Geim, Sergei V Morozov, De-eng Jiang, Yanshui Zhang, Sergey V Dubonos, Irina V Grigorieva, and Alexandr A Firsov. Electric field effect in atomically thin carbon films. *science*, 306(5696):666–669, 2004.

[76] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016.

[77] James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

[78] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016.

[79] John Brian Pendry. Negative refraction makes a perfect lens. *Physical Review Letters*, 85(18):3966, 2000.

[80] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchical vq-vae. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10775–10784, 2021.

[81] Said Pertuz, Edith Pulido-Herrera, and Joni-Kristian Kamarainen. Focus model for metric depth estimation in standard plenoptic cameras. *ISPRS Journal of Photogrammetry and Remote Sensing*, 144:38–47, 2018.

[82] David W Piston and Gert-Jan Kremers. Fluorescent protein fret: the good, the bad and the ugly. *Trends in biochemical sciences*, 32(9):407–414, 2007.

[83] Alexander Poddubny, Ivan Iorsh, Pavel Belov, and Yuri Kivshar. Hyperbolic metamaterials. *Nature photonics*, 7(12):948–957, 2013.

[84] Alin C Popescu and Hany Farid. Statistical tools for digital forensics. In *International Workshop on Information Hiding*, pages 128–147. Springer, 2004.

[85] Alin C Popescu and Hany Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10):3948–3959, 2005.

[86] Yael Pritch, Eitam Kav-Venaki, and Shmuel Peleg. Shift-map image editing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 151–158, 2009.

[87] Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.

[88] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proc. Med. Image Computing and Computer Assist. Interven. (MICCAI)*, pages 234–241, Shenzhen, China, Oct. 2019.

[89] Rahul Roy, Sungchul Hohng, and Taekjip Ha. A practical guide to single-molecule fret. *Nature Methods*, 5(6):507–516, 2008.

[90] S Rytov. Electromagnetic properties of a finely stratified medium. *Soviet Physics JEPT*, 2:466–475, 1956.

[91] Parikshit Sakurikar and P. J. Narayanan. Composite focus measure for high quality depth maps. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[92] Ronald Salloum, Yuzhuo Ren, and C-C Jay Kuo. Image splicing localization using a multi-task fully convolutional network (mfcn). *Journal of Visual Communication and Image Representation*, 51:201–209, 2018.

[93] Changha Shin, Hae-Gon Jeon, Youngjin Yoon, In So Kweon, and Seon Joo Kim. Epinet: A fully-convolutional neural network using epipolar geometry for depth from light field images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4748–4757, 2018.

[94] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, May 2015.

[95] Durgesh Singh and Sanjay K Singh. Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability. *Journal of Visual Communication and Image Representation*, 38:775–789, 2016.

[96] Pratul P Srinivasan, Rahul Garg, Neal Wadhwa, Ren Ng, and Jonathan T Barron. Aperture supervision for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6393–6401, 2018.

[97] Pratul P Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng. Learning to synthesize a 4D RGBD light field from a single image. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2243–2251, Oct. 2017.

[98] Jian Sun, Huibin Li, Zongben Xu, et al. Deep ADMM-Net for compressive sensing mri. *Advances in Neural Information Processing Systems*, 29, 2016.

[99] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[100] Yu-Ju Tsai, Yu-Lun Liu, Ming Ouhyoung, and Yung-Yu Chuang. Attention-based view selection networks for light-field disparity estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12095–12103, 2020.

[101] Anil Kumar Vadathya, Sharath Girish, and Kaushik Mitra. A unified learning based framework for light field reconstruction from coded projections. *arXiv preprint arXiv:1812.10532*, 2018.

[102] Sheng-Yu Wang, Oliver Wang, Andrew Owens, Richard Zhang, and Alexei A Efros. Detecting photoshopped faces by scripting photoshop. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 10072–10081, 2019.

[103] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8695–8704, 2020.

[104] Ting-Chun Wang, Jun-Yan Zhu, Ebi Hiroaki, Manmohan Chandraker, Alexei A Efros, and Ravi Ramamoorthi. A 4d light-field dataset and cnn architectures for material recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 121–138. Springer, 2016.

[105] Weiyue Wang and Ulrich Neumann. Depth-aware cnn for rgb-d segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018.

[106] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 329–338, 2018.

[107] Armand Wirgin. The inverse crime. *arXiv preprint math-ph/0401050*, 2004.

[108] Yue Wu, Wael Abd-Almageed, and Prem Natarajan. BusterNet: Detecting copy-move image forgery with source/target localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 168–184, 2018.

[109] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7508–7517, 2020.

[110] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *Proc. ICLR*, May 2016.

[111] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4471–4480, 2019.

[112] Cheng Zhang, Nathaniel Kinsey, Long Chen, Chengang Ji, Mingjie Xu, Marcello Ferrera, Xiaoqing Pan, Vladimir M Shalaev, Alexandra Boltasseva, and L Jay Guo. High-performance doped silver films: overcoming fundamental material limits for nanophotonic applications. *Advanced Materials*, 29(19):1605177, 2017.

[113] Cheng Zhang, Dewei Zhao, Deen Gu, Hyunsoo Kim, Tao Ling, Yi-Kuei Ryan Wu, and L Jay Guo. An ultrathin, smooth, and low-loss al-doped ag film and its application as a transparent electrode in organic photovoltaics. *Advanced Materials*, 26(32):5696–5701, 2014.

[114] Dehui Zhang, Zhen Xu, Zhengyu Huang, Audrey Rose Gutierrez, Cameron J Blocker, Che-Hung Liu, Miao-Bin Lien, Gong Cheng, Zhe Liu, Il Yong Chun, et al. Neural network based 3d tracking with a graphene transparent focal stack imaging system. *Nature Communications*, 12(1):1–7, 2021.

[115] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[116] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017.

[117] Shuo Zhang, Hao Sheng, Chao Li, Jun Zhang, and Zhang Xiong. Robust depth estimation for light field via spinning parallelogram operator. *Computer Vision and Image Understanding*, 145:148–159, 2016.

[118] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in GAN fake images. In *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2019.

[119] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2017.

[120] Changyin Zhou, Daniel Miau, and Shree K Nayar. Focal sweep camera for space-time refocusing. 2012.

[121] Z. Zhou, X. Chen, and O. C. Jenkins. LIT: Light-field inference of transparency for refractive object localization. *IEEE Robotics and Automation Letters*, 5(3):4548–4555, 2020.

[122] Bo Zhu, Jeremiah Z Liu, Stephen F Cauley, Bruce R Rosen, and Matthew S Rosen. Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487–492, 2018.

[123] Mingqiang Zhu and Tony Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report*, 34:8–34, 2008.